Architecting Virtual Machine Labs

Written by: Tony Robinson

Dedication: This is dedicated to IT and computer security people, newbie and veteran alike. I wrote this guide for two reasons: 1) “Published Author” sounds like a great title to have (plus I can cross it off my bucket list) 2) Knowledge sharing. One day, you’re going to inherit the mess of infrastructure and terrible security practices we leave behind. This guide may not have all the answers, but hopefully it will give you a nice head start. I just hope that if this guide helped you in some way, that maybe you’ll repay the kindness to the generation that follows you, offering to guide them as well.

I would like to thank my wife for her patience and unending love, my family for inspiring me to do better, my employer for saying ‘GO WRITE YOU A BOOK’, and finally, several friends who offered to step in and provide feedback and editing when I asked for help. I threw walls of text into a Google doc, but you made it awesome.

Table of Contents

Purpose of this Book

This guide is designed to teach you about virtualization and how to build out the virtual machine lab environment that is easy to maintain, portable, relatively well secured, and flexible enough to accommodate IT and security students that need an environment to practice their trade. The goal is to teach you how to build the baseline network and get familiar with using a hypervisor of your choice. The initial network and VM (Virtual Machine) design we will be working on together can easily be expanded upon, or swapped out to support various roles, such as:

- Testing and/or developing new systems administration tools
- Learning the ropes for offensive security tools for red team
- Practicing with detection and response tools for blue team
- Providing a safe, secure environment to perform reverse engineering, malware analysis and/or exploit development with reasonably good security protections in place

This guide is not meant to be read from front to back. If you do this, you are going to get really bored, and notice a lot of repetition. Think of this book as a “Choose your own adventure” novel; this book covers how to produce a robust virtual machine lab environment across five different hypervisors. Unless you’re crazy, a VM enthusiast, or a researcher (or some combination), the likelihood that you will want or need to read all five hypervisor setup guides is going to be pretty low. Keeping that in mind, here are my recommendations:

Read all the chapters up to, and including the “Hypervisor Guides” chapter in order to develop a better understanding of the skills you’ll want and need to create your own lab, better understand how virtualization works in general, hardware recommendations, and finally, understand what you are building, before you pick a hypervisor and actually start building your lab environment. There are then five chapters detailing how to perform initial setup and configuration of five unique hypervisors:

- Oracle VirtualBox
- Microsoft Client Hyper-V
- VMware Workstation Pro
- VMware Fusion Pro
- VMware ESXi

These chapters instruct you on how to acquire and install the hypervisor of your choosing, configure the hypervisor and VMs to support the virtual machine lab environment I will teach you how to build, and perform initial installation and setup of those virtual machines. Choose a hypervisor that suits your budget and your goals, and follow the setup instructions.

After performing the setup and configuration tasks for the hypervisor of your choice, you are then meant to finish configuring the virtual machines in the lab environment, as well as the hypervisor host or management workstation you will be using to access your virtual machines. Each of the hypervisor setup guides has a section entitled “Next Steps” that will guide you on recommendations on what tasks need to be done to finish making the lab functional (e.g. the IDS and Splunk installation chapters), as well as what supplemental chapters to consider reading (e.g. enabling remote access for the lab VMs, hardening hosted hypervisors on Windows, network design factors for bare-metal hypervisors, etc.) for a much better experience when utilizing your newly built lab environment.

A Note About Software Versions

Writing books for security and/or most IT disciplines is a daunting task. The moment you put ink to paper, the information contained in the book deprecates. You see this a lot with textbooks where there are multiple revisions that need to be written to discuss updates in the material.

I'll make mention of what software version for both hypervisors and operating installation ISOs I used throughout the guides, but don't obsess over using the exact same version I used when I made these guides. These are merely the software versions I had available to me while writing this book. As a security practitioner, I always recommend updating your software when updates are available, and using the most current software version available. This includes hypervisors and OS distributions.

If you're from the future and using future versions of hypervisors, and future versions of operating systems, you may notice that some configuration settings may or may not be in the exact place a screen capture I made said it was going to be in. Button colors and styles may have changed, radio buttons may now be checkboxes, etc. This is because UI (User Interface) developers may or may not have changed exactly where a given configuration setting is. This is just a fact of life when it comes to new software releases. Sometimes they do it because they can't leave well enough alone, or sometimes they just want to make the user experience (UX) for their product better.

So if you're panicking because a given configuration setting has moved, or a checkbox isn't in the location indicated by my screen captures or instructions indicated they should be, **don't panic**. This is the first rule of any IT related discipline. The second rule is that **software changes**. Sometimes arbitrarily, sometimes for the better. The third rule is to **consult the documentation.** Maybe the configuration option has migrated to a new menu location, or maybe it was integrated as part of another, related setting. Consult the product patch notes, documentation included with the software, and/or online knowledgebase/forums for the product to find out where the configuration setting lives now. The goal of this book isn't to mindlessly instruct you to click here, open this menu, and check these boxes, it is also for you to understand WHAT the configuration settings that you are modifying do, and WHY I am telling you to modify them so that if and when you want to experiment, make changes, and add or remove features to your lab, you have the knowledge and proper understanding what the settings do as opposed to what buttons you need to press.

Prerequisite Knowledge

Here are some things you should have a basic to intermediate grasp of in order to get the most out of this guide:

- **TCP/IP Networking**
    - Understand what an IP address is as well as how to configure an IP address, default gateway and/or DNS servers in Windows as well as Linux/Unix-based operating systems. (e.g. network adapter properties in Windows, `ifconfig`, `ip link`, `ip route`, `route`, `/etc/resolv.conf`, etc.)
    - Basic understanding of RFC1918 addressing
    - Basic understanding of subnetting
    - Understand the Open Systems Interconnection (OSI) model and/or TCP/IP networking models
    - A basic understanding on how stateful firewalls operate, and on what criteria they typically block (e.g. IPv4/IPv6 addresses, transport protocol (tcp/udp/icmp) and port/icmp code (e.g. port 80/tcp, port 123/udp, icmp type 0))
    - Familiarity on ports and protocols (e.g. is it TCP or UDP) for common network services (e.g. FTP, HTTP/HTTPS, NTP, SSH, DNS, etc.)
- **Familiarity with Operating Systems and their Installation Procedures**
    - Familiarity with installing Linux/Unix/Windows operating systems is absolutely necessary. If you know what an ISO image is, and how to boot from one, this will help you progress a bit faster
    - Familiarity with the Unix/Linux/Windows command line (e.g. “shells”) is key; most of the lab is based on Linux and BSD systems that will NOT have a GUI installed. You’ll need to know how to navigate the shell to navigate the file system, edit configuration files, utilize CLI tools, and run shell scripts.
        - You should know how to perform various network troubleshooting commands from the command line on Linux/OS X/BSD systems, as well as Windows systems (e.g. `ping`, `netstat`, `wget`, `curl`, `ipconfig`, `ifconfig,` etc.)
        - You should be familiar with at least one command line text editor that Linux/OS X/BSD systems use (e.g. `vim`, `nano`, `emacs`, etc.), or if you insist on writing/editing your config files on Windows, then copying them to lab VMs, you should be familiar with the `dos2unix` command to prevent Linux systems from failing to parse config files properly
        - Familiarity with SSH and SCP for remotely managing and copying files to Linux/OS X/BSD systems will be extremely helpful
- **Virtualization**
    - If you are familiar with the basics of virtualization, it will make a lot of this guide

much easier to understand.

There are numerous resources out there for learning the basics I've listed. If you understand most of the points listed above, you shouldn't have too hard of a time following along. If you don't… things might be difficult at first, but the only way they will get easier is if you struggle along the way. If you are not against reading a good book or two, the publishing company No Starch Press produces a variety of high quality reference books to help you gain a better understanding of various subjects; of particular note are the books "The Linux Command Line", "TCP/IP Guide", "Network Know-How", and "Practical Packet Analysis".

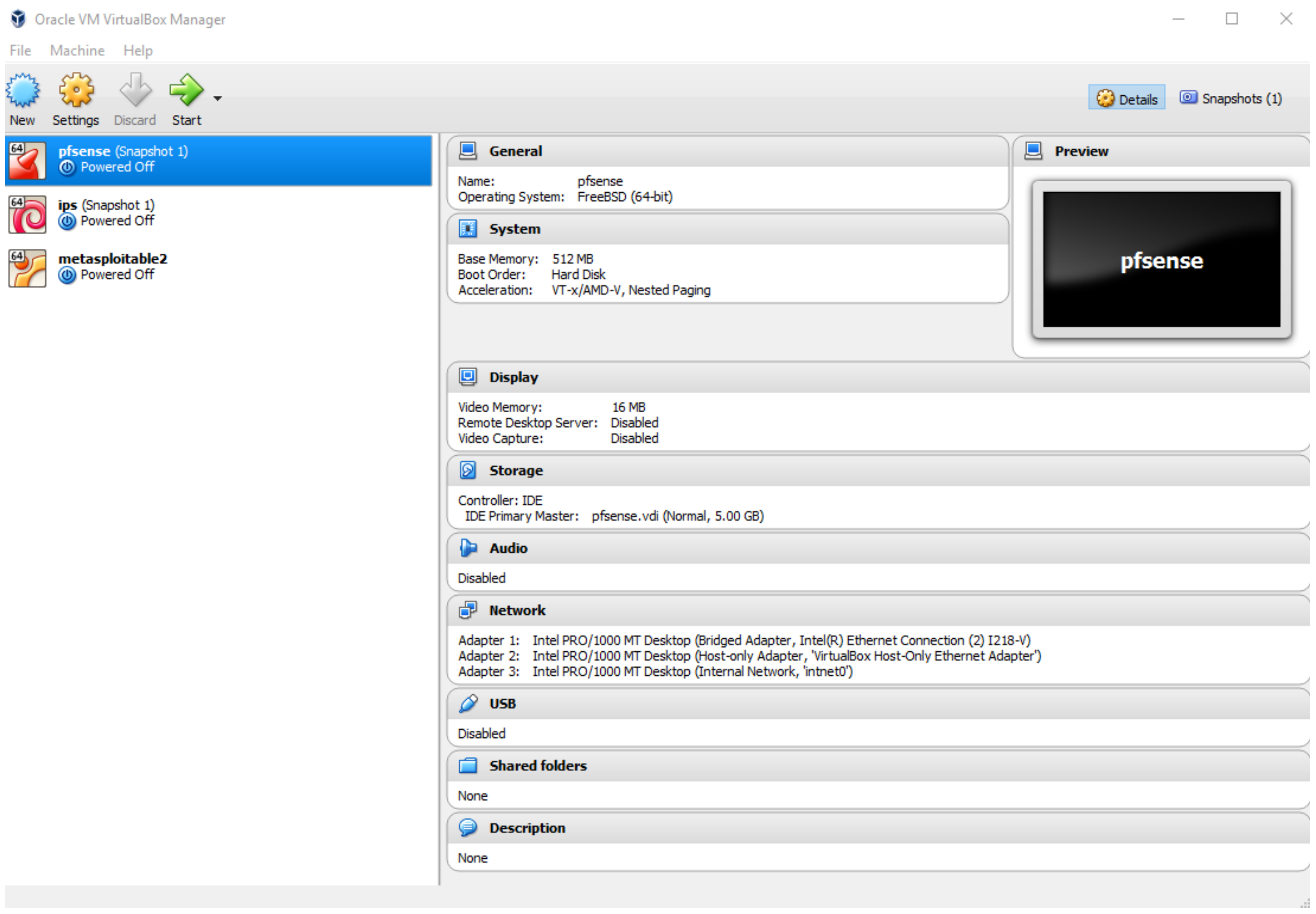
If you're looking for free resources, the website cybrary.it has a great collection of free videos. Take a look at some of the videos produced for COMPTIA certifications, specifically the Linux+, A+ and Network+ coursework, as they have direct application to what we're doing here, and will teach enough of the basics to allow you to progress a bit easier. Additionally, Zed Shaw has created a series of guides collectively called "Learn Programming The Hard Way". He has a nice crash course introduction to the command line for various operating systems here: https://learnpythonthehardway.org/book/appendixa.html.

Hypervisor and Hardware Considerations

Before you get started with the hypervisor setup guides, it helps to know some of the basics of virtualization. This is going to involve you learning about hardware resources hypervisors need, how VMs actually work, virtual networking, and finally, understand how your hypervisor choices will influence your lab's network design. Before we dive into the hypervisor guides, I'm going to give you the crash course on what virtualization is, different types of hypervisors, how virtual networking works on different hypervisors, illustrate how the lab network we will be created, then instructing you on how to recreate the lab network and VMs on a variety of hypervisor platforms.

Introduction to Virtualization

Virtualization is the concept of taking one physical computer and splitting its resources into separate chunks/containers in order to host smaller, independent "virtual" systems using those reserved resources. For example, if you have a physical machine with 2 CPU cores, 4GB of RAM, and 40GB of disk space, you could, using special software called a hypervisor, allocate a portion of these resources to create a container to host a virtualized computer. You could install any OS you have licensing for and have it using that chunk of disk, CPU and memory you set aside. These virtual machines, for all intents and purposes, are independent computers. You can host many virtual machines on a single physical computer, with the only limitations being the amount of resources the host computer has, and how many resources the virtualized computers (known as virtual machines, or VMs) require to run.



pfsense (Snapshot 1)
Powered Off
ips (Snapshot 1)
Powered Off
metasploitable2
Powered Off

General
Name: pfsense
Operating System: FreeBSD (64-bit)

System
Base Memory: 512 MB
Boot Order: Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging

Preview
pfsense

Display
Video Memory: 16 MB
Remote Desktop Server: Disabled
Video Capture: Disabled

Storage
Controller: IDE
IDE Primary Master: pfsense.vdi (Normal, 5.00 GB)

Audio
Disabled

Network
Adapter 1: Intel PRO/1000 MT Desktop (Bridged Adapter, Intel(R) Ethernet Connection (2) I218-V)
Adapter 2: Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')
Adapter 3: Intel PRO/1000 MT Desktop (Internal Network, 'intnet0')

USB
Disabled

Shared folders
None

Description
None

In the screen capture above for example, there are three VMs listed. The VM that is highlighted is named “pfsense”. In this case, I set aside 512MB of ram, 5GB of hard drive space, and 1 virtual CPU to install and run the pfSense OS (operating system) in a VM.

Virtualization became a big deal because it allows companies to run more services, and do more things with less physical hardware; not to mention it's easier for developers, researchers, and IT professionals to have a testing environment full of virtualized systems for testing compatibility, patch deployments, and a ton of other tasks that would have required a lot of physical hardware. Not only that, virtualization software supports “snapshots” or the ability to capture the status of a virtualized system at a certain point in time, and restore the virtual machine to that state at will. This means that test cases, malware analysis and several other scenarios that make a significant number of chances to the system's operating status can be undone in mere moments, restoring the VM to a known good configuration state with the click of a button.

Introduction to Hypervisors

Now that you know what a VM is, let's talk about hypervisors. There are a wide variety of them to choose from. Your first hurdle is choosing which option is right for you, and that depends on a number of different factors.

What is a Hypervisor?

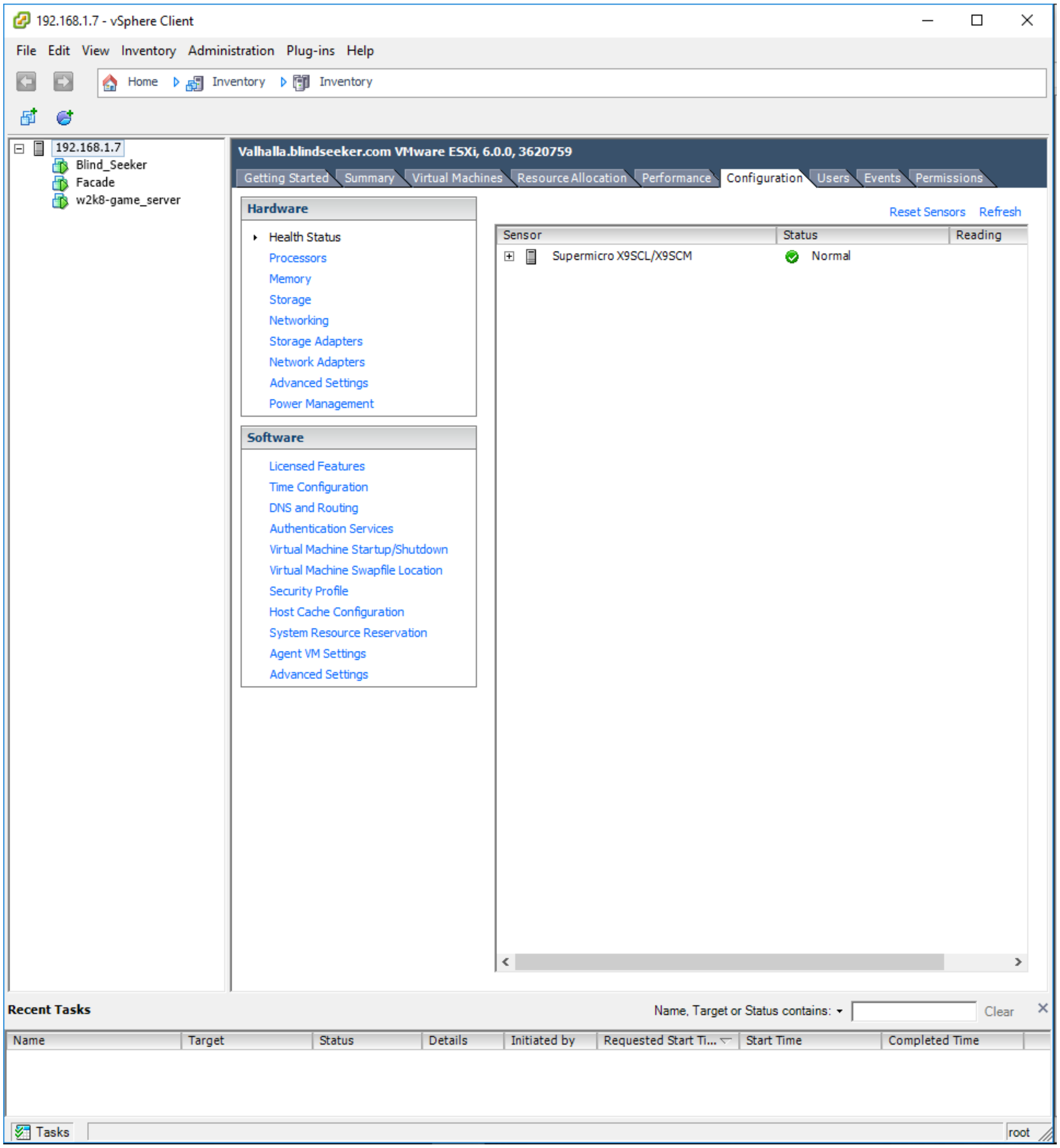
A hypervisor is software that is used to create, manage, and run, virtual machines. Hypervisors are responsible for resource allocation (CPU, disk, RAM), network configuration, virtual hardware allocation, snapshot management, and controlling the current operating status of VMs that they are responsible for managing. There are two classes of hypervisors recognized today: bare-metal hypervisors, and hosted hypervisors.

Bare-metal Hypervisors

A bare-metal hypervisor is installed directly onto server hardware (in IT/sysadmin circles, known as “big iron” or the “bare metal” - hence the name for this class of hypervisor). Bare-metal Hypervisors usually feature a very lightweight operating system with minimal functionality in addition to their hypervisor functionality. In fact, in most cases, bare-metal hypervisor installation can be done on an SD card or USB drive, leaving the internal hard drives and/or Solid State Drives (SSDs) 100% dedicated to the virtual machines.

Once the bare-metal hypervisor is installed, booted, and given a network configuration, typically it is managed remotely over the network through a web interface, API, or through some sort of an application installed on a workstation you use to manage the hypervisor remotely. For sake of simplicity, throughout this guide, I will refer to this system as a management workstation. All the major configuration aspects for a bare-metal hypervisors are done from the management console/web interface, using a management workstation.

While bare-metal hypervisors tend to be very lean, they are packed with advanced features with more of a focus towards enterprise environments that require those features for a production environment where downtime means lost revenue. These are features such as fault tolerance, availability/failover solutions, advanced networking, etc. Since bare-metal hypervisors are commonly used in enterprise networks all over the world, it would definitely benefit you to learn how to use them in lab environment, and making yourself familiar with how they operate. However, be aware that some bare-metal hypervisors are EXTREMELY picky about what hardware they work with (VMware vSphere Hypervisor for example, is notorious for being extremely picky with what hardware it will detect and use). Even then, even if you can actually get the hypervisor to recognize your system’s hardware, not all of your hardware’s features may be supported. For example, integrated network ports on a system motherboard may not be supported, or a built-in RAID controller may not be recognized by the hypervisor. This is something you either have to deal with, unfortunately. Either you will need to find supported hardware, or find another bare-metal hypervisor that isn’t so picky about what hardware it supports. Popular bare-metal hypervisors include Microsoft's Hyper-V (server edition), Citrix Xenserver, Proxmox, and VMware vSphere Hypervisor (also known as ESXi).

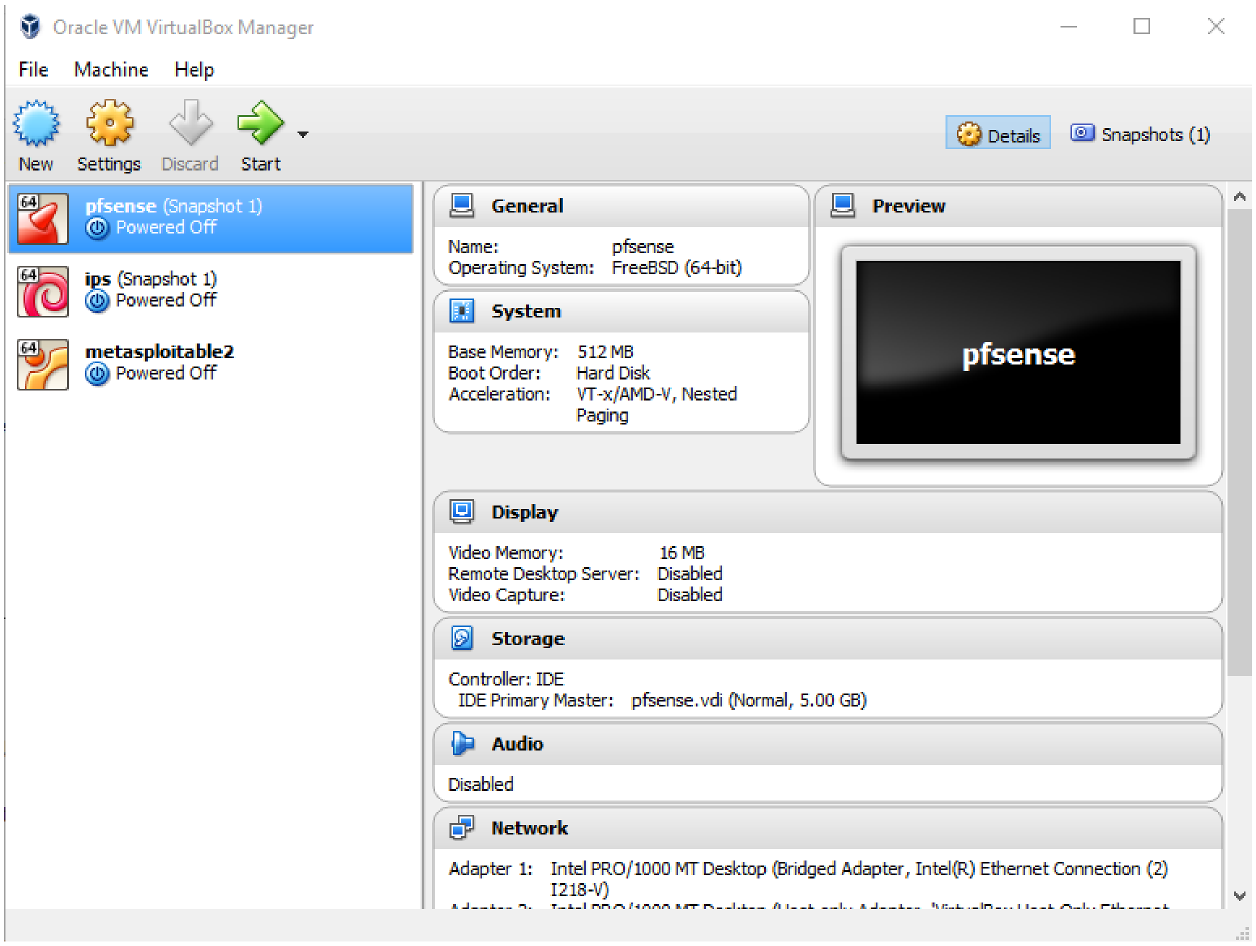


Sensor	Status	Reading
Supermicro X9SCL/X9SCM	Normal	

Note: The illustration is VMware vSphere Client, which is used to manage ESXi bare-metal hypervisors remotely. The newest versions of ESXi also feature a built-in web interface for system management as well.

Hosted Hypervisors

Hosted hypervisors differ from bare-metal hypervisors in that they are an application that is installed on top of an already installed operating system, such as a Linux, OS X, BSD or Windows. Hosted hypervisors are applications that are “hosted” by the installed operating system, thus the name. Typically, some sort of a console is installed as a part of the hosted hypervisor to allow users logged on to the workstation to manage the configuration locally, though some hosted hypervisors have remote management features that can also be used. Popular hosted hypervisors include Oracle VirtualBox, VMware Workstation, VMware Fusion, and Microsoft Client Hyper-V.



pfsense (Snapshot 1) Powered Off
ips (Snapshot 1) Powered Off
metasploitable2 Powered Off

General
Name: pfsense
Operating System: FreeBSD (64-bit)

Preview
pfsense

System
Base Memory: 512 MB
Boot Order: Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging

Display
Video Memory: 16 MB
Remote Desktop Server: Disabled
Video Capture: Disabled

Storage
Controller: IDE
IDE Primary Master: pfsense.vdi (Normal, 5.00 GB)

Audio
Disabled

Network
Adapter 1: Intel PRO/1000 MT Desktop (Bridged Adapter, Intel(R) Ethernet Connection (2) I218-V)

The illustration depicted above is Oracle VirtualBox. VirtualBox is a free, multi-platform hosted hypervisor.

Hardware Considerations

What hardware do you have available to run a VM lab? Do you plan on using a spare desktop you had lying around? Your primary workstation? A laptop? An actual server? The hardware you have available to dedicate to your lab environment will determine how far you can take your lab, and what type of hypervisor you will be using (e.g. bare-metal vs. hosted). In this section, we will be discussing how RAM, Disk I/O, number of CPU cores, CPU featureset, and motherboard BIOS dictate how well your hypervisor and VMs will perform.

RAM as a Performance Factor

All operating systems, applications, utilities and VMs all require some amount of system memory to perform their tasks. Obviously, the more you have, the more applications and VMs with their own applications you can run concurrently. While there are some design factors in hypervisors that allow RAM that is not being used in one VM to be allocated to another VM when it is needed, you should never rely on these technologies, because it could result in overextending your RAM, which leads to swapping/paging to disk, which results in “disk thrashing”, which leads to disk performance problems, further exacerbating your performance problems. The only solutions to a lack of RAM are to reduce the number of running applications and services to decrease the demand, or to increase the amount of RAM installed on your system and/or the amount of RAM allocated to the affected VMs.

Disk I/O as a Performance Factor

If at all possible, regardless of bare-metal or hosted hypervisor, you should consider using SSDs where possible. SSDs are a relatively new storage medium. They have virtually no seek time and read/write performance that is orders of magnitude faster than your traditional magnetic platter based hard drives.

These performance factors are a big deal if you are doing multiple disk intensive things across your VMs at the same time. This means that the VMs will be fighting one another for I/O (input/output) access to read and write data from the same disk, which directly impacts the performance of your VMs, and if you’re running a hosted hypervisor, also affects the performance of the OS you are hosting the hypervisor on. This means that the more VMs you have on the same disk doing disk intensive tasks, the longer they all have to wait and content for disk access to read and write their data to the hard drive.

If you don’t have an SSD or didn’t budget for one, don’t fret too much. Here are a few recommendations that might help you squeeze performance out of your disks and allow your

VMs to coexist peacefully.

- If possible, consider installing more than one disk to running your VMs. Bare-metal hypervisors benefit by being able to spread disk I/O across multiple disks. Hosted hypervisors benefit by being able to dedicate one of the disks for OS operations, and the other for VM storage and operations.
- If possible, consider putting your disks into a RAID1 (minimum of 2 disks) or RAID5 (minimum of 3 disks) array to protect against drive failure and possibly improve disk I/O.
- If you are using a bare-metal hypervisor, consider installing it on a separate disk, USB drive, or SD card (if possible), in order to fully dedicate your disk drives solely to VM storage.
- If you are using a hosted hypervisor, minimize the amount of resource-intensive intensive tasks and applications you have running on the host OS (aside from the hypervisor of course).
- Do everything possible to avoid paging/swapping to disk on your VMs. This occurs when VMs need more RAM than the system has physically available; the VM will start using the disk like RAM. This should be avoided at all costs, even if you are using SSDs since paging/swapping causes intense wear and tear on all drive types platter or SSD (this is also known as disk thrashing). The key to preventing this is to ensure your have plenty of RAM installed, and that you have allocated enough RAM to your VMs. This means you have to monitor system performance on a regular basis.

What is seek time?

Seek time is the time delay associated with traditional spinning platter-based hard drives that is required for them to spin up and move tiny arms over portions of the disk platter (called read/write heads) to retrieve the data a system requires.

CPU Cores and Features as a performance Factor

All services and applications have processing and calculations that need to be done to achieve some sort of input or output, or manage some aspect of system operation. All of these applications and services need some slice of time on the CPU to have their calculations performed so that they perform their functions. Hypothetically, the more CPUs/cores you have available, the more tasks a system can perform at once. Whether that is multiple tasks for a single application, or multiple tasks for multiple processes the idea is supposed to be that more cores are better. Each VM has at least 1 virtual CPU (tied to the physical CPUs on your system) attached to it, with the ability to allocate more within the limits of how many physical CPUs/cores you have installed on your PC, laptop, or server.

Each VM has services and applications that need CPU time to run calculations to manage system services and input/output for running applications, just like any other physical system. The more intense the applications and services are in terms of CPU processing required, the more time and utilization those applications take to finish their calculations. This means that other applications and services need to wait for time on the CPU to perform their calculations. This in turn results in applications and services (even the entire OS in some cases) becoming unresponsive during extremely CPU intensive tasks.

Like with RAM utilization, the only solutions here are to reduce the number of running applications or services, deal with the factors that are causing them to take so much CPU time, or increase the number of CPUs/cores available.

In addition to the number of cores your system has available to handle CPU load, it is also important to confirm that both your CPU and/or system motherboard include support for virtualization features. Intel and AMD processors have virtualization extensions (Intel:VT-x AMD:AMD-V) that are built-in CPU features. Most of the time you can simply use your favorite search engine, and search for your processor name and the results will return links to the CPU manufacturer that usually include a spec sheet page that confirms whether or not virtualization features are included in that particular CPU model. Some of the low-end CPUs do NOT include virtualization extensions in order to cut costs, so be aware of this!

In addition to checking the CPU specs, you also need to confirm that the motherboard BIOS supports virtualization. If you are using a prebuilt system from a large PC manufacturer (e.g. Dell, HP, etc.) then usually it's as easy as searching for your PC or laptop’s model name on the manufacturer’s support website to find and download a system manual, or spec sheet to confirm the features the system supports. However, if you’re like me and you’re into building your own PCs, try visiting the motherboard manufacturer’s website, search for the model of motherboard you wish to use, then download and review the motherboard documentation to confirm that virtualization is a supported BIOS feature.

Performance is a Vicious Cycle

You should be seeing a pattern at this point. Don’t have enough RAM? That leads to resource contention for memory and swapping/paging to disk. Don’t have enough disk I/O? That leads resource contention for apps to read/write to and from disk. Don’t have enough CPU/cores? That leads to resource contention for calculations. All of these can feed on one another and result in poor system stability and performance as a whole.

Keep an eye on CPU, disk and RAM performance metrics on a regular basis for your hosted and/or bare-metal hypervisors, and adjust accordingly. Windows systems have task manager, while most Unix-like operating systems (That is, Linux, OS X, BSD) have several performance measuring utilities that can be ran from the command line such as `top`, `free`, `iostat` and `iotop`. Most bare-metal hypervisors have their own performance graphs that measure the performance of the hypervisor as a whole, as well as the performance of individual VMs in terms of disk, RAM, and CPU utilization.

Understanding Virtual Networks - Hosted vs. Bare-metal Hypervisor Networking

Let's discuss how networking is done on both hosted and bare-metal hypervisors, because this is a key factor in how you access your lab VMs, and how our lab will be built on bare-metal and hosted hypervisors.

Hosted Hypervisor Networking - Host-Only, Bridged, and NAT Network Segments

In most hosted hypervisors, virtual networks are typically divided into special network segments that all serve a different function. There are usually three types of network segments that VMs get connected to. “NAT” network segments, “Bridged” network segments, or “Host-Only” network segments. Some hosted hypervisors allow you create additional custom network segments, but they usually fit into one of these three categories.

Bridged Networking

Bridged network segments are used to share the host system’s network card to directly connect to the same network as the host operating system. In this case, VMs that connect to the bridged network segment act as though they are directly connected to the same physical network as the host system. From a network perspective, they look exactly like any other system connected to your physical network. They can have their own IP address on the physical network to which they are attached, and would respond to network requests like any other system on that physical network.

NAT Networking

If you have an understanding on how Network Address Translation/Port Address Translation (NAT/PAT) works on a regular/home network for connecting multiple machines to the internet through a single publically routable IP address, then NAT network segments are exactly how you’d imagine them: The host operating system makes connections to external resources on behalf of the VMs connected to the NAT network. VMs connected to a NAT network share the host system’s IP address. All outbound traffic appears to be coming from the host system, and any services you set up on the VM for external access (through port forwarding) appear to be hosted on the host system as well.

For the rest of you who have no idea what I’m talking about, VMs that get added to a NAT network have their own internal network addressing scheme (IP address range, subnet mask, default gateway, etc.). The default gateway for this private network that does that NAT functions is usually a special virtual network card or IP address attached to the host operating system. Any traffic not destined for other hosts in the NAT network gets routed to this address. The host operating system then sends out the request on behalf of the system on the NAT network, using its network connection. The responses (or lack thereof) are then relayed back to the VM in the

NAT network. NAT networks can be used to share the host operating system's network connection and make it appear as though the connection requests are coming directly from the host operating system, instead of directly exposing the virtual machines to the network. Some hosted hypervisors also allow you to setup port forwarding so that connections to the host OS on certain ports get forwarded to a specific VM's IP address in the NAT network.

In general, I tend to avoid using NAT network segments that most hypervisors provide. The only time I generally use NAT networks for network connectivity is if for some reason, bridged networking isn't working properly. For instance, a bridged VM isn't getting an IP address, or network connectivity is otherwise broken (maybe there is some sort of MAC address filtering, preventing network access, etc.). I also tend to avoid using NAT network segments because with most hosted hypervisors, the port forwarding functionality that NAT networks are supposed to provide is either buried in the user interface, or requires modifying configuration files buried somewhere in the hosted hypervisor's installation files.

Host-Only Networking

Host-only networks are network segments that, by definition, do not get any sort of network connectivity to the outside world. VMs on these network segments can only communicate with hosts on the same host-only virtual network, and/or the hypervisor itself.

Virtual Network Adapters and You

The NAT and Host-Only networks allow the hypervisor's host OS to communicate with VMs inside of their respective networks through the use of virtual network adapters. The hypervisor creates a virtual network card and attaches it to the hypervisor host. To the host OS running the hypervisor, this virtual network card is just like any physical network card, and can be configured like one. Virtual network adapters allow your hypervisor host to connect to VMs in these networks, and interact with their network services.

Some hypervisors actually allow you to disable creation of the virtual network adapter for these network segments. In the case of a NAT network, this would create a network that uses the host system for external network connectivity, but the host would not be able to directly connect to VMs behind the NAT IP address; you'd have to configure port forwarding to access any of the network services for the VMs in that NAT network. In the case of host-only networks, this creates a "private" internal network in which the VMs could only connect to one another, but the host OS would not be able to connect (except through the hypervisor's console so that you could still install and configure VM on this network). Usually this is done to provide some degree of isolation between the hypervisor host and the VMs, and provide better network segmentation.

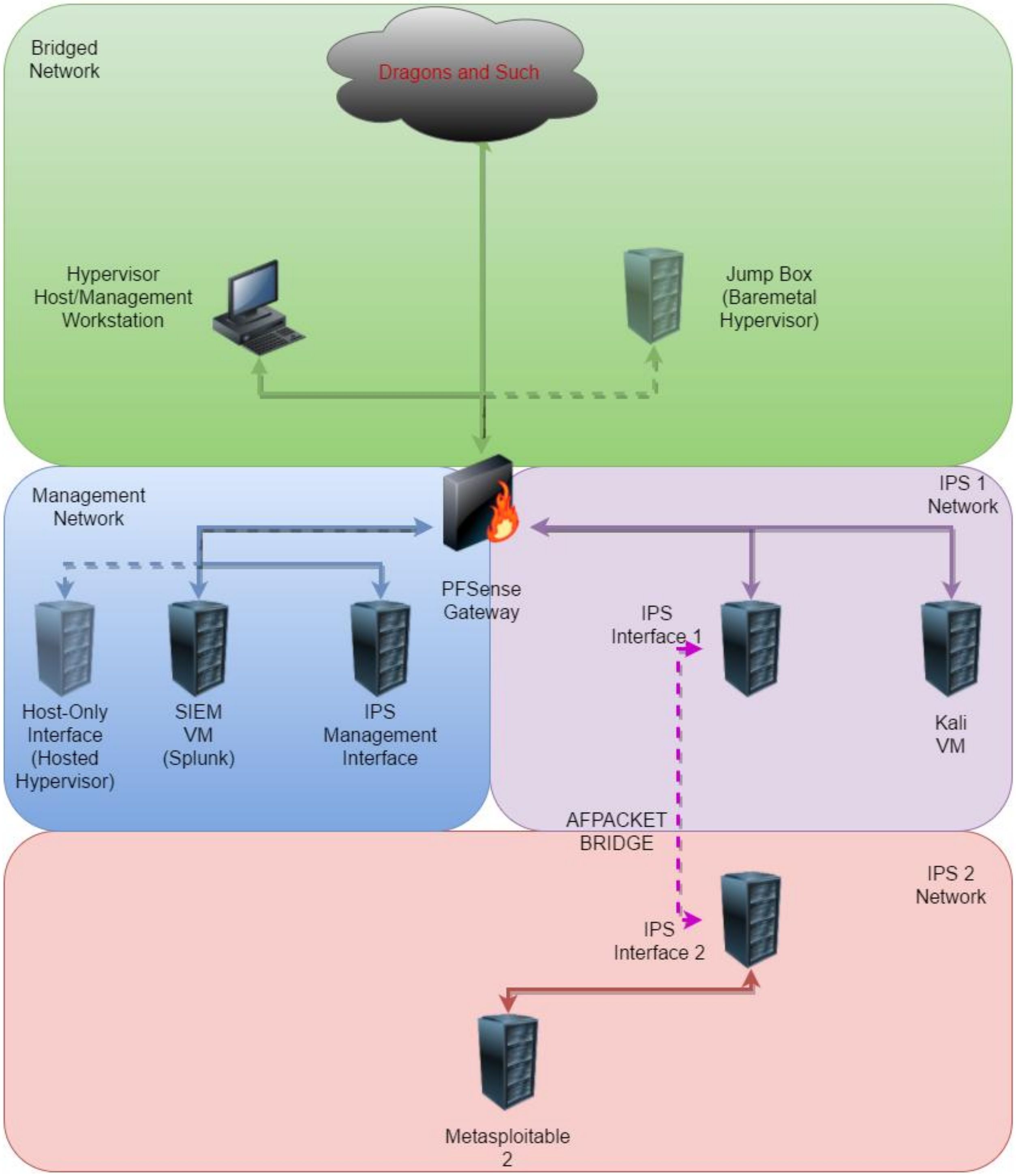
Bare-metal Hypervisor Networking - Virtual Switches

Most bare-metal hypervisors implement virtual networking via virtualized network switches (vswitches). You don't really have the concept of host-only, NAT and/or bridged networks the same way you do in hosted hypervisors. I'm going to use VMware's vSphere Hypervisor (also known as ESXi) as an example here, since most bare-metal hypervisors (but not all) implement their virtual networking in roughly the same manner.

With ESXi, you simply have virtual switches that have a number of ports you can attach virtual machines to, so that they are on the same Layer 2 network. Those vswitches can also be attached to the server's physical network cards, allowing virtual machines attached to that switch to interact with the rest of the network through that network card. In this way, it emulates how most professional grade network switches have designated "uplink" ports to connect the switch to the rest of a larger enterprise network, not unlike how a hosted hypervisor's bridged nework operates. Alternatively, you could choose not to uplink to the server's physical network connection and the hosts attached to that vswitch would be considered isolated and only able to talk to machines attached to that particular vswitch, not unlike how a hosted hypervisor's host-only network operates.

Lab Overview

In this chapter we will review a network diagram of the VM lab we will be building together, discuss resource allocations for our VMs, and discuss how our virtual network lab will be laid out, regardless of what hypervisor you use.


Bridged Network
Dragons and Such
Hypervisor Host/Management Workstation
Jump Box (Baremetal Hypervisor)
Management Network
PFSense Gateway
IPS 1 Network
Host-Only Interface (Hosted Hypervisor)
SIEM VM (Splunk)
IPS Management Interface
IPS Interface 1
Kali VM
AFPACKET BRIDGE
IPS 2 Network
IPS Interface 2
Metasploitable 2

Lab Network Description

We have 5 virtual machines in our network, and 4 distinct network segments. We're using a pfSense VM with three network interfaces to route traffic and provide security for the bridged, management, and both IPS networks. pfSense is a very popular, flexible and easy to use BSD-based firewall distro that, in addition to acting as a network firewall, can provide a variety of other network services. Our pfSense VM will handle routing traffic between network segments as necessary, network segmentation and security through a series of firewall rules, and core network services for our lab environment, including DHCP, DNS, and NTP services. Having pfSense sit between all of these network segments leads to why the network was designed the way it was. Each of the network segments in our lab has a particular function and reason for being there, so let's talk about how they're set up and why.

Bridged Network

The bridged network is like an upstream internet connection that connects our VMs to the local physical network the hypervisor is connected to, and (eventually) the internet. Firewall rules we put on the pfSense VM will be set up to prevent local hosts from accessing the lab VMs, and will prevent the lab VMs from communicating while the local hosts on the physical network, while still allowing internet access as necessary. If you're running a bare-metal hypervisor, you will also be setting up firewall rules to allow your management workstation, or a dedicated "jump box" to access the VM lab systems.

Management Network

The management network acts as a trusted, safe and secure network. This network is where our SIEM (Security Intrusion Events Manager) VM will be, and one of the three network interfaces we will be using for IPS (Intrusion Prevention System) VM (I will explain why the IPS VM has 3 interfaces in just a moment), and one of the three firewall interfaces for our pfSense VM. Our IPS VM will be running either Snort (https://snort.org) or Suricata (https://suricata-ids.org) for network inspection, while our SIEM VM will be running Splunk (https://www.splunk.com) to collect and allow us to query generated alerts from our IPS system.

IPS 1 and IPS 2 Networks

The IPS 1 network hosts a Kali Linux VM, and one of our three network interfaces connected to the IPS VM, while the IPS 2 network hosts the third of 3 network interfaces for our IPS VM, as well as the Metasploitable 2 VM. Kali Linux is a very popular and easy to use penetration testing distro that is loaded with offensive security tools and scripts, while Metasploitable 2 is a very vulnerable VM. Metasploitable has been traditionally used to introduce red teamers to penetration testing and exploitation with a tool called The Metasploit Framework. In our case, we're using Kali and Metasploitable 2 for the express purpose of testing our IPS system and making sure that these two VMs on the two network segments can communicate with one another.

AFPACKET Bridging between IPS 1 and IPS 2

Wait, so, if IPS vswitch 2 is NOT connected to the pfSense VM, how does Kali reach it? Remember how I mentioned our IPS VM has three network interfaces? One is connected to the management network, while the other two are connected to both the IPS 1 and IPS 2 networks. We will be connecting the IPS 1 and IPS 2 networks together through our IPS VM, using a technology available in both Snort and Suricata called AFPACKET bridging.

Why All The Trouble?

AFPACKET bridging allows the two network interfaces of the IPS VM, connected to IPS 1 and IPS 2 networks to be fused together, allowing communication between the two network segments, so long as the IPS VM is powered on, and the Suricata or Snort IPS service is running. For instance, if you wanted your lab to serve as a malware analysis lab and wanted to ensure complete isolation to the internet. If you place your VMs in the IPS 2 network, you could then turn off the IPS VM, or disable the Snort/Suricata service, and VMs in the IPS 2 network have no external access to the internet or any of the other network segments (even the IPS 1 network) whatsoever. Conversely, if you are host pentesting lab with your VMs hosted on the IPS 2 network, and you have a bunch of lab systems you wanted to update, then isolate again, you can turn on the IPS VM, bridge the two networks together, and get your network access from the pfSense interface connected to the IPS 1 network. After you're done downloading updates, turn off the IPS VM, and you have an isolated lab environment again. This network design is called "fail-closed networking".

This is how it works in a nutshell.

Fail-closed networking operates on the idea if that you would rather have the network be secure and unable to communicate with external networks, as opposed to fail-open, and have no protection/isolation at all. The advantages for our lab in implementing a fail-closed network are pretty great. At a moment's notice, our lab VMs can have access to the internet, and at a moment's notice, if there is some sort of a catastrophic issue, the VMs in the IPS 2 network can be isolated quickly. This allows us the ability to switch between the equivalent of a host-only and bridged network in an instant. To put it bluntly, it's the secret sauce behind how our lab can be modified to accommodate a variety of needs for IT and/or information security training needs.

VMs, Resource Allocations, and Minimum Hardware Requirements

We will be using at least 5 virtual machines as a part of the lab network we will be building together.. Here are the recommended resource allocations for the VMs::

- pfSense: 512MB RAM, 5GB Disk, 1 cpu/core
- Kali Linux:4GB RAM, 80GB Disk, 1 cpu/core
- SIEM (Ubuntu 16.04 Server): 4GB RAM, 80GB Disk, 1 cpu/core
- IPS (Ubuntu 16.04 Server): 2GB RAM, 80GB Disk, 1 cpu/core
- Metasploitable 2: 512MB RAM, 10GB Disk, 1 cpu/core

If you add up all the disk and RAM requirements, we're looking at 255 GB of disk space to host the VMs (note that this does NOT account for space required for VM snapshots), and 11GB of RAM. As such, I would recommend a box with a minimum of 16GB of RAM, 4 CPU cores (with Intel's VT-x or AMD's AMD-V virtualization technology supported by the CPU and motherboard), and 500GB of disk space, **at a minimum**. As with most hardware requirements, more is always better.

The resource allocations I've recommended for the VMs ensure that each of the VMs performs well. The hardware recommendations guarantee that you have room to create an additional VM or two, as well as store snapshots of your VMs, operating system files, hypervisor files, and OS installation ISOs.

If you are really in a squeeze and need to fit more VMs on the same disk, or you have less than 16GB of RAM to work with here are some things you can do to try and spread your hardware a bit further. Keep in mind that pushing your system too hard, or spreading your system resources too thin can lead to poor performance.

- Since this is a simple lab environment, you may consider cutting the RAM allocations on the SIEM and Kali Linux VMs from 4GB to no less than 2GB. This has the potential to reduce the max RAM consumption across your VMs from 11GB to 7GB, but graphical tools you use in the Kali Linux VM may be a little sluggish, and take a little longer to load.
- You could decrease the disk space allocation on the SIEM, IPS and Kali Linux VMs from 80GB each, to 60GB each for the IPS and SIEM VMs, and 40GB for the Kali Linux VM. This has the potential to save you 20-80GB of disk space, but means you will have to closely monitor disk usage in your VMs.

Hypervisor Guides

If you’ve made it this far, you know what we’re building, you have the hardware to build it, and you may even have a hypervisor in mind that you want to use. I have created a guide on how to set up 5 different hypervisors to support the lab network that we want to create. Which one you choose is up to you. Please note that any software requirements specified below are in addition to the hardware recommendations we discussed earlier (e.g. RAM, disk, number of cores, Virtualization Extension support, etc.). Here are your hypervisor choices:

Microsoft Client Hyper-V: A hosted hypervisor by the Microsoft Corporation for users of Windows 8.1 or Windows 10. You must be running Professional, Enterprise, Ultimate or Education edition, and the operating system must be 64-bit.

Oracle VirtualBox: A hosted hypervisor by the Oracle Corporation. Available on practically every operating system out there. 64-bit OS is recommended.

VMware Fusion Pro: A hosted hypervisor by the VMware Corporation made specifically for Apple OS X. This guide specifically requires VMware Fusion Pro edition.

VMware Workstation Pro: A hosted hypervisor by the VMware Corporation for Microsoft Windows and Linux systems. 64-bit OS recommended.

VMware vSphere Hypervisor (ESXi): A bare-metal hypervisor by the VMware Corporation. Extremely picky about hardware compatibility, but free and very robust. Support for new and/or exotic systems may not be guaranteed.

I have provided step-by-step instructions on how to create your VM lab on the five hypervisors listed above. Each hypervisor setup guide has the following sections:

- Initial installation of your hypervisor
- Setup and customization of the hypervisor
- Detailed step-by-step guide on creating your first VM, pfSense
    - OS installation
    - Initial network setup
    - Initial setup via the webConfigurator (the pfSense web interface)
    - Taking your first snapshot
    - Redirecting you to the pfSense Firewall Rules and Network Services guide to fully set up the pfSense VM including guidance on:
        - Setting up the firewall rules for the Bridged, Management and IPS networks
        - Setting up core network services
            - DHCP
            - DNS

        - NTP
        - Squid Proxy (optional)
- Guidelines to follow for creating 3 out of the remaining 4 VMs
- Instructions on how to acquire and set up Metasploitable 2 as necessary
- Recommendations on remaining chapters to read to create a fully functional lab environment such as:
    - Defense in Depth for Windows Hosted Hypervisors
    - Remote Lab Management
    - Network Design Factors When Working with Bare-metal Hypervisors
    - IDS/IPS Installation
    - Splunk Installation

So without further adieu, choose a hypervisor, and get started!

Setup - Microsoft Client Hyper-V

Note: Please be aware that this guide was written using Client Hyper-V running on Windows 10 Professional, anniversary edition. If you are running Client Hyper-V on Windows 8.x, you may notice some slight differences, but the instructions should more or less be the same.

I've been using Client Hyper-V as my hosted hypervisor of choice recently because it's convenient (I primarily run Windows for playing video games and other applications), the hypervisor's featureset is robust, and the price is right (read: free) if you're running the right version of Windows. This guide will instruct you on how to create the our lab environment on Client Hyper-V, one step at a time.

Installation

The only downside to running Client Hyper-V, is that you have to be running a premium version of Windows in order to get the option to install it. Specifically, Client Hyper-V is an additional feature you can install for the following Windows operating systems:

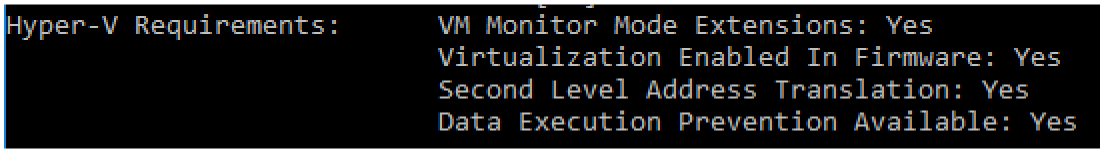
- Windows 8 (or 8.1) Pro 64-bit
- Windows 8 (or 8.1) Enterprise 64-bit
- Windows 10 Enterprise
- Windows 10 Professional
- Windows 10 Education

Note: While not explicitly stated on the "Windows 10 Hyper-V System Requirements" page on the MSDN (https://msdn.microsoft.com/virtualization/hyperv_on_windows/quick_start/walkthrough_compatibility), you'll need a 64-bit OS to be able to address all the memory we will need to use for our lab. Therefore, you'll want to use a 64-bit version of Windows 10. Additionally, **any drives you**

use to store your virtual machines must be formatted as NTFS. I discovered this by mistake when I tried to run my VMs on an ex-FAT formatted SSD drive, and experienced errors initializing/starting my VMs. (https://blogs.msdn.microsoft.com/virtual_pc_guy/2008/08/28/hyper-v-vm-on-usb-disk-fails-to-start/)

So in addition to requiring one of the premium editions of Windows 8 or Windows 10, There are certain specific hardware requirements that Client Hyper-V needs in order to run. To make it easy however, you can open up a command prompt and run:

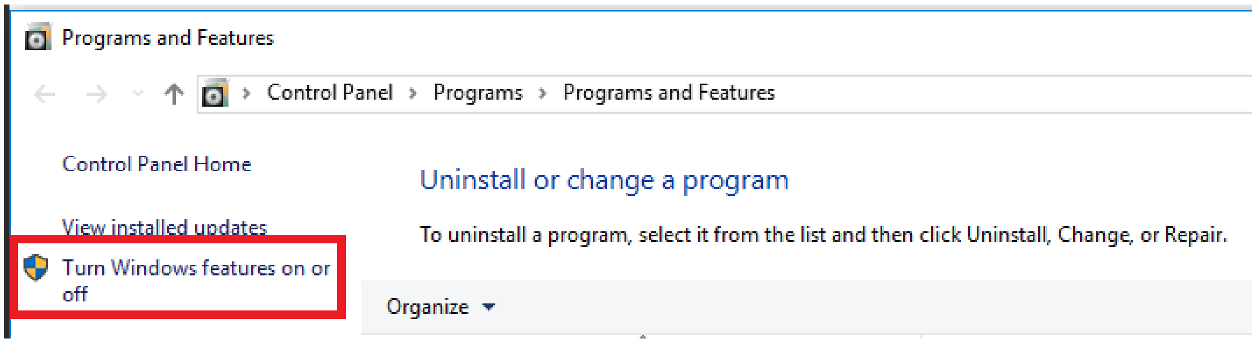
```
systeminfo.exe
```



```
Hyper-V Requirements:      VM Monitor Mode Extensions: Yes
                           Virtualization Enabled In Firmware: Yes
                           Second Level Address Translation: Yes
                           Data Execution Prevention Available: Yes
```

The last thing the command will output is a section labeled *Hyper-V Requirements:* This will tell you whether or not your hardware can run Client Hyper-V. If all the options come back Yes, you should be good to go. Otherwise, double check that your motherboard has virtualization options enabled and that your CPU supports the required virtualization options as well.

To install Client Hyper-V, navigate to the *Programs and Features* menu, and select the *Turn Windows Features on or off* option.



Programs and Features

Control Panel › Programs › Programs and Features

Control Panel Home

View installed updates

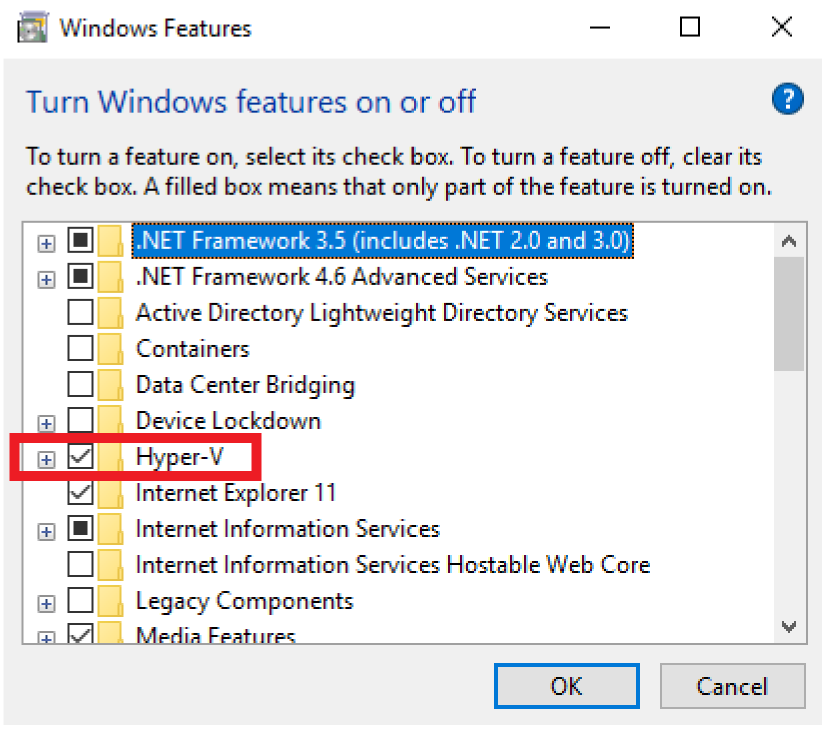
Turn Windows features on or off

Uninstall or change a program

To uninstall a program, select it from the list and then click Uninstall, Change, or Repair.

Organize

In the next window, Locate the *Hyper-V* feature, and click the checkbox so that a checkmark appears in the box, then click *OK*. Windows will enable Client Hyper-V and instal the management tools/console for managing the hypervisor. After the installation is complete, you will need to reboot your system.

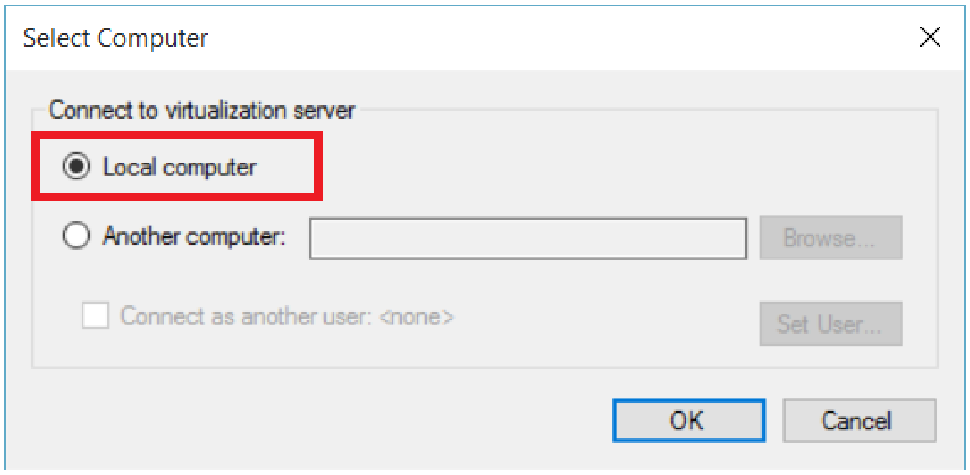

Windows Features

Turn Windows features on or off

To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.

- ■ .NET Framework 3.5 (includes .NET 2.0 and 3.0)
- ■ .NET Framework 4.6 Advanced Services
- ☐ Active Directory Lightweight Directory Services
- ☐ Containers
- ☐ Data Center Bridging
- ☐ Device Lockdown
- ✓ Hyper-V
- ✓ Internet Explorer 11
- ■ Internet Information Services
- ☐ Internet Information Services Hostable Web Core
- ☐ Legacy Components
- ✓ Media Features

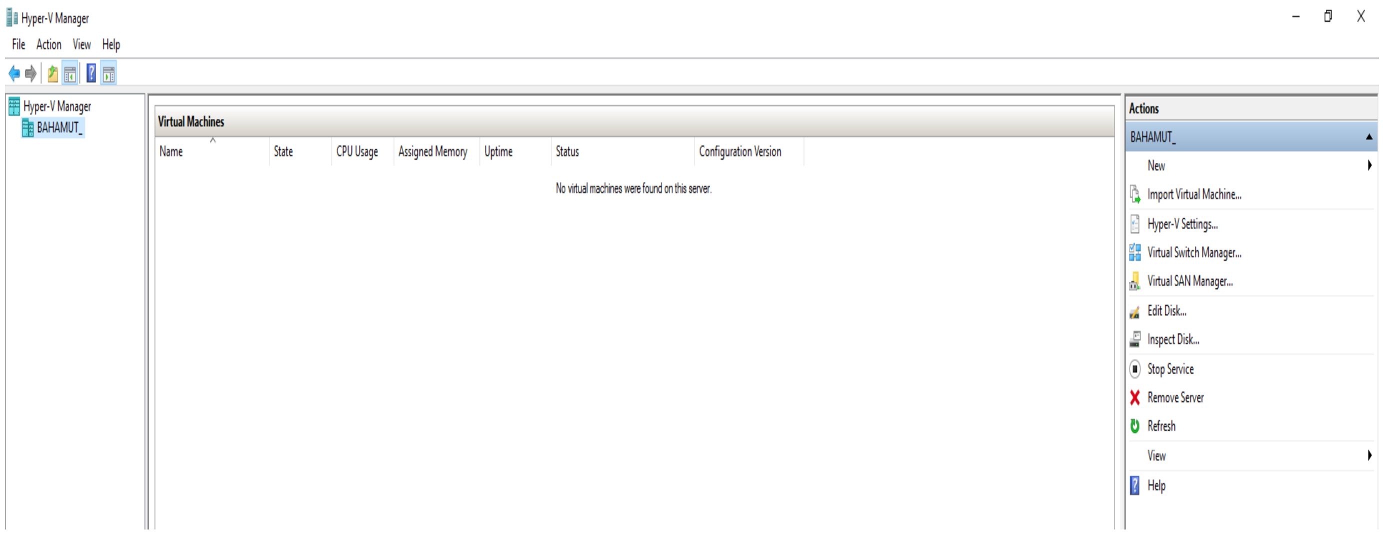
After you have rebooted, run the *Hyper-V Manager* application. The system will ask you if there is a remote server you wish to connect to, or if you would like to connect locally. Select the *Local computer* radio button and click *OK*.


Select Computer

Connect to virtualization server

- ● Local computer
- ○ Another computer:
- ☐ Connect as another user: <none>

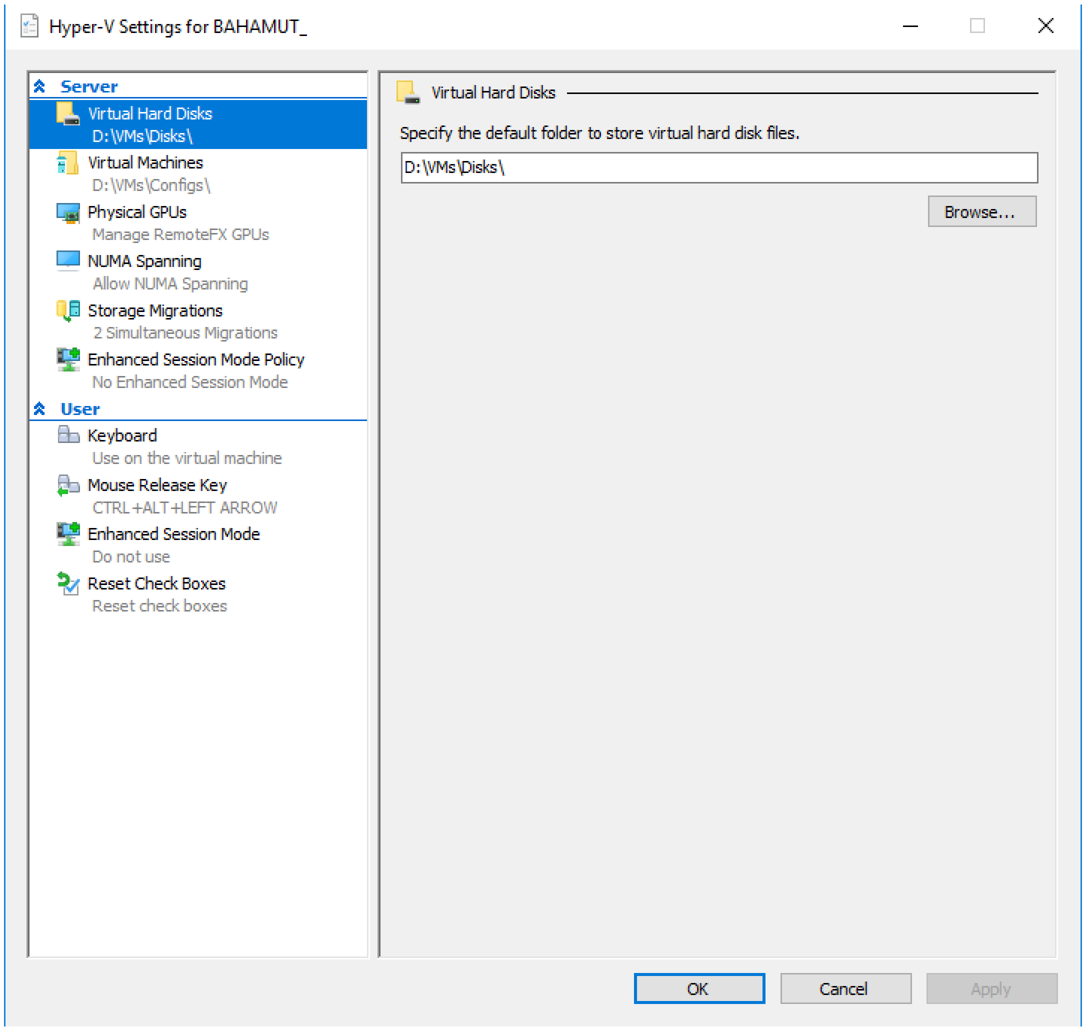
You should be greeted by a screen that looks like this:



No virtual machines were found on this server.

Hypervisor Preferences

Now that Client Hyper-V installed, let's do some customization. On the right pane of the Hyper-V Manager interface labeled *Actions*, click on *Hyper-V Settings…* to bring up the Hyper-V Settings menu for your system.



Hyper-V Settings for BAHAMUT_

Server

- Virtual Hard Disks — D:\VMs\Disks\
- Virtual Machines — D:\VMs\Configs\
- Physical GPUs — Manage RemoteFX GPUs
- NUMA Spanning — Allow NUMA Spanning
- Storage Migrations — 2 Simultaneous Migrations
- Enhanced Session Mode Policy — No Enhanced Session Mode

User

- Keyboard — Use on the virtual machine
- Mouse Release Key — CTRL+ALT+LEFT ARROW
- Enhanced Session Mode — Do not use
- Reset Check Boxes — Reset check boxes

Virtual Hard Disks

Specify the default folder to store virtual hard disk files.

D:\VMs\Disks\

You'll notice that on the left pane, there are two subsections: *Server* and *User*. The Server settings affect how the Hyper-V server operates, while the User settings determine how the client interacts with VMs on the Hyper-V console.

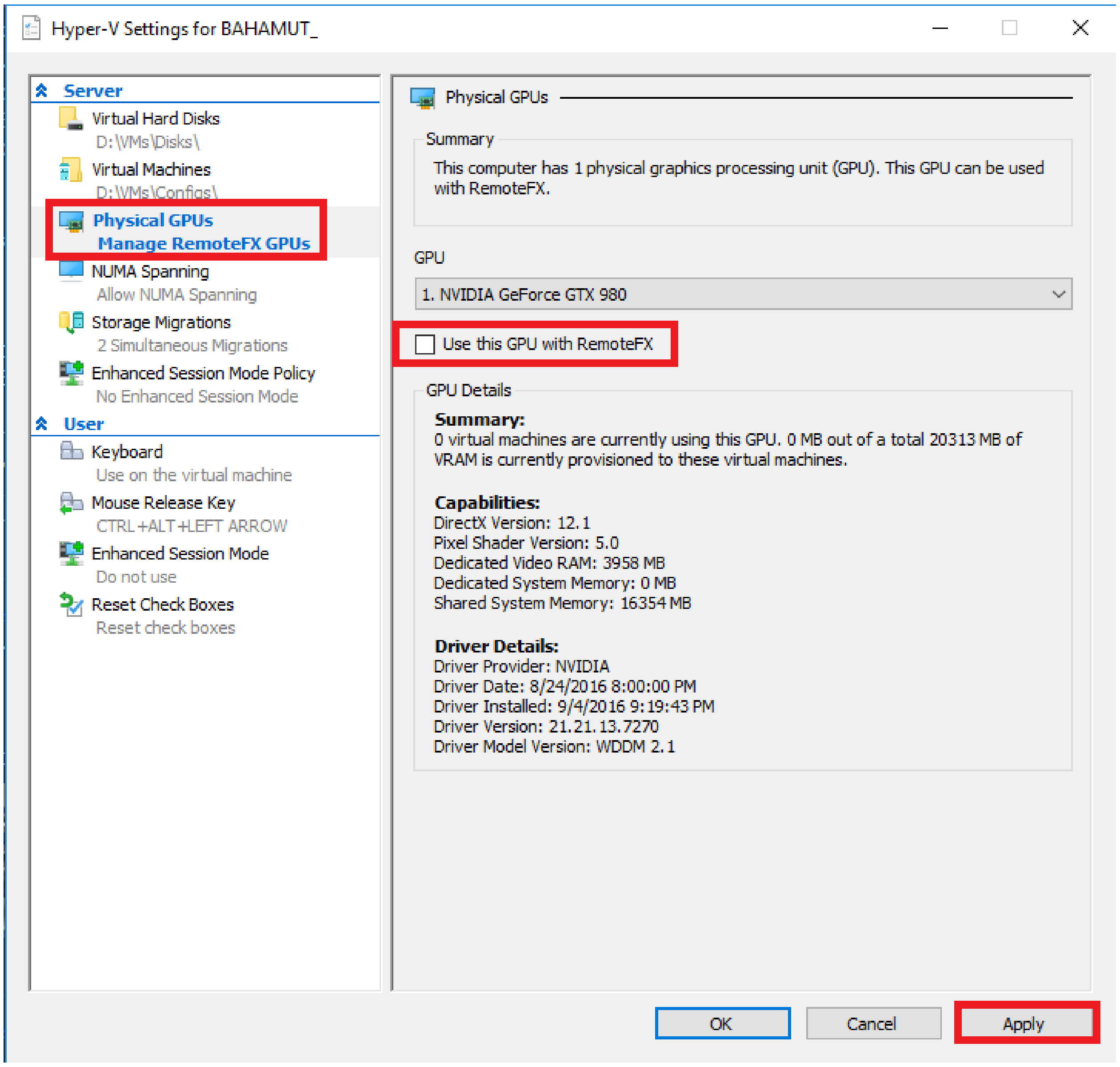
Server Settings

The first server setting, *Virtual Hard Disks* Determines where on the system the virtual hard drives, known as vhd (virtual hard drive) or vhdx (virtual hard drive extended) files in Hyper-V terminology, will be stored. The files are the containers that Hyper-V uses for installing virtual machine operating systems and the files for that operating system.

In the hardware considerations portion of this guide, we discussed the importance of disk I/O as a performance factor, but lets recap here. If you have a second physical hard drive installed on your system, and you are using a hosted hypervisor, it is a good practice to place your virtual machine files on a separate hard drive that the host operating system is NOT installed on. This is in order to maximize the amount of disk I/O available to both your VMs, and the host operating system. For example, If you have two physical hard drives installed in Windows formatted as the `C:` and `D:` drives, and you installed Windows on the `C:` drive (usually the default setting), you would want to store the VM files on the `D:` drive to make sure that your VMs and the host OS are not competing for disk I/O, impacting the performance of both the host OS and the VMs you wish to run. I chose to locate my VHD files under `D:\VMs\Disks`. If you change this setting, be sure to click the *Apply* button in the bottom right corner of the window.

The second server setting, *Virtual Machines*, defines where the configuration files and folders for each of your VMs should be stored. While these files aren't as big as the VHD files, generally speaking, its good practice to store the virtual machine files in the same location as the VHDs for organizational purposes. In my case, I chose to store the virtual machine files in `D:\VMs\Configs`.

The next server setting, *Physical GPUs*, determines whether or not virtual machines created will have direct hardware access to any installed GPUs on the host system. There should be no reason to have this functionality enabled, since none of our VMs have the means to utilize hardware like that, so you'll want to uncheck the *Use this GPU with RemoteFX* checkbox, then click the *Apply* button.



This computer has 1 physical graphics processing unit (GPU). This GPU can be used with RemoteFX.

1. NVIDIA GeForce GTX 980

[ ] Use this GPU with RemoteFX

Summary:
0 virtual machines are currently using this GPU. 0 MB out of a total 20313 MB of VRAM is currently provisioned to these virtual machines.

Capabilities:
DirectX Version: 12.1
Pixel Shader Version: 5.0
Dedicated Video RAM: 3958 MB
Dedicated System Memory: 0 MB
Shared System Memory: 16354 MB

Driver Details:
Driver Provider: NVIDIA
Driver Date: 8/24/2016 8:00:00 PM
Driver Installed: 9/4/2016 9:19:43 PM
Driver Version: 21.21.13.7270
Driver Model Version: WDDM 2.1

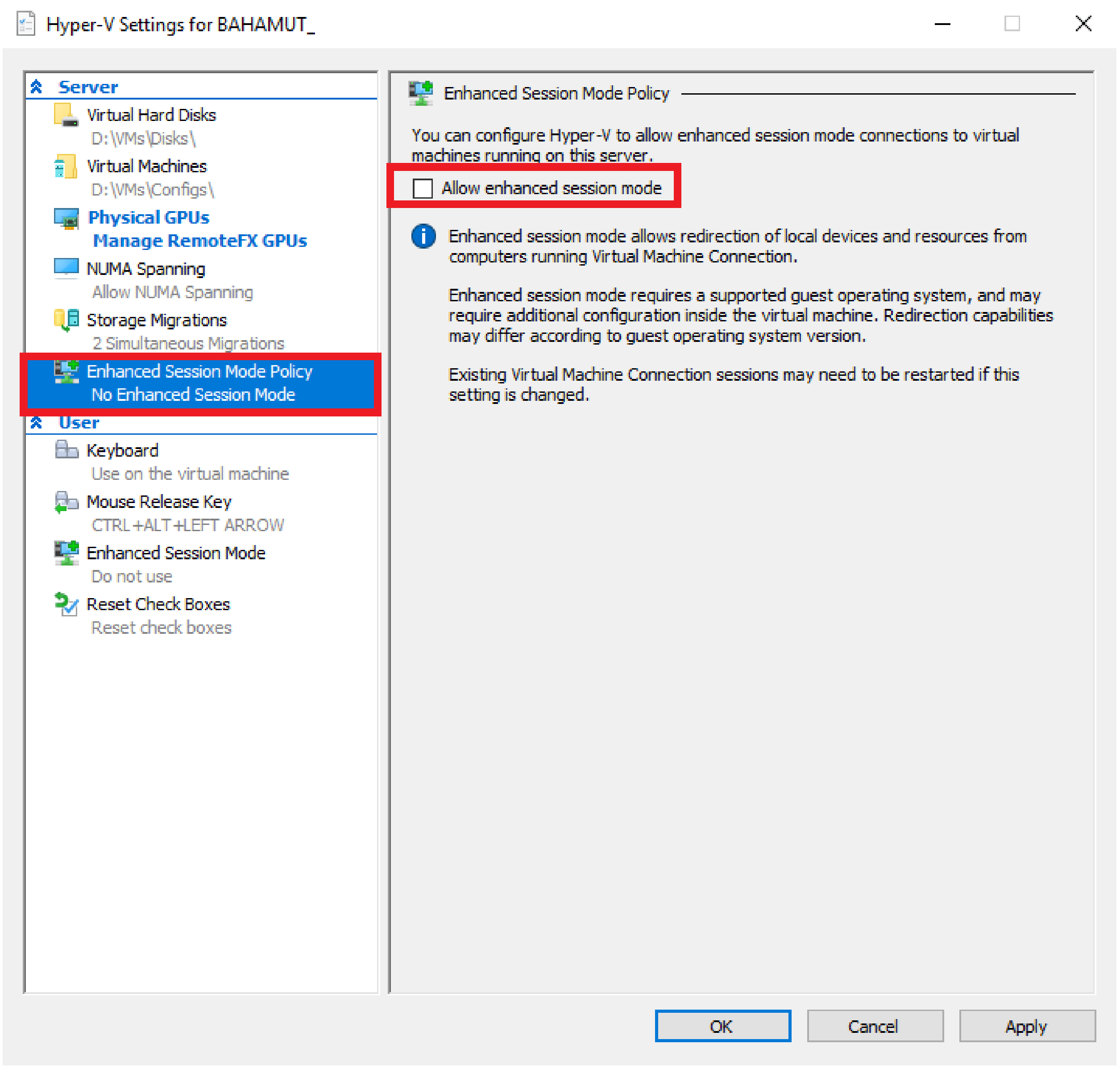
I left the *NUMA Spanning* and *Storage Migrations* settings alone, Since they won't really have an effect on our environment. NUMA spanning is more for massive multi-CPU systems with tons of memory. It allows the system to use memory that isn't necessarily local to the CPU. Storage migration is the ability to move a number of VMs while the VMs you want to migrate are still up and running. It's a useful feature for enterprise environments, but a feature you won't have to worry about too often in a lab environment. The only time you'll be moving VMs is if you get a hardware upgrade, and chances are, you can spare the downtime in your lab if that's the case.

The final server setting, *Enhanced Session Mode Policy* governs whether or not extra features of the vmconnect application will be available or not. The vmconnect application is what you use to interact with your VMs in Hyper-V. If you're familiar with how Microsoft's Remote Desktop Protocol application works, vmconnect operates in a very similar manner.

In the RDP application, used for Remote Desktop connections, there is a tab called "Local Resources" that allows you to select what things, if anything, you want to have available when you connect to a remote computer You can select local drives, printers, sound devices, etc., and have them behave as though they're connected to the remote system. Enhanced Session Mode is essentially the same thing, except for the vmconnect application. If Enhanced Session Mode is allowed, the vmconnect client will be able to share client resources on the VM the user connects to.

As always, there are restrictions. Enhanced Session Mode only applies to Windows Hyper-V VMs that have Remote Desktop Protocol enabled. This makes sense, seeing as how it is practically the same thing. . Since our lab is almost entirely Linux/BSD based, this has absolutely no bearing on us whatsoever. However, since the possibility exists that Windows VMs may be added to your lab network later as you need them, so I would highly recommend unchecking the *Allow enhanced session mode* checkbox, then clicking the *Apply* button to disable it.

You need to be very careful about features that make resources from your hypervisor host available to VMs in your lab, because this has a habit of breaking segmentation, security boundaries, and isolation between the VM host and the VMs themselves. For instance, if you had enhanced session mode enabled, and shared drives from your host system to the VM, there is a chance that your hypervisor host could become infected with malware, or active samples could escape your lab network, limiting your ability to effectively contain it. This is why I recommend disabling enhanced session mode entirely, as a precaution.


Hyper-V Settings for BAHAMUT_

Server
Virtual Hard Disks
D:\VMs\Disks\
Virtual Machines
D:\VMs\Configs\
Physical GPUs
Manage RemoteFX GPUs
NUMA Spanning
Allow NUMA Spanning
Storage Migrations
2 Simultaneous Migrations
Enhanced Session Mode Policy
No Enhanced Session Mode
User
Keyboard
Use on the virtual machine
Mouse Release Key
CTRL+ALT+LEFT ARROW
Enhanced Session Mode
Do not use
Reset Check Boxes
Reset check boxes

Enhanced Session Mode Policy

You can configure Hyper-V to allow enhanced session mode connections to virtual machines running on this server.

☐ Allow enhanced session mode

Enhanced session mode allows redirection of local devices and resources from computers running Virtual Machine Connection.

Enhanced session mode requires a supported guest operating system, and may require additional configuration inside the virtual machine. Redirection capabilities may differ according to guest operating system version.

Existing Virtual Machine Connection sessions may need to be restarted if this setting is changed.

If you want to find out more about NUMA spanning check out this MS technet article:

https://blogs.technet.microsoft.com/pracheta/2014/01/22/numa-understand-it-its-usefulness-with-windows-server-2012/.

If you're interested in learning more about Storage Migration, you can read this MS technet article:
https://technet.microsoft.com/en-us/library/hh831656(v=ws.11).aspx.

If you'd like to learn more about Enhanced Session Mode, go here:
https://technet.microsoft.com/en-us/windows-server-docs/compute/hyper-v/learn-more/use-local-resources-on-hyper-v-virtual-machine-with-vmconnect.

User Settings

The Hyper-V user settings are fairly self-explanatory, but for sake of completeness, let's go over them.

The *Keyboard* setting determines how the system should respond to special key combinations when you are using the keyboard while interacting with a VM through the console. This setting determines whether or not the host OS or the VM should interpret the keyboard shortcut you are sending.

The *Mouse Release Key* setting defines what key combination should be entered in order to release control of the mouse from a virtual machine. Most hypervisors (hosted or bare-metal) have a way to allow the user of the hypervisor to "attach" to the console of a VM they are running. This would be the equivalent of attaching your keyboard, mouse, and monitor directly to the VM. If you're familiar KVM (Keyboard, Video, Mouse) switches operate, its like that. When you interact with the VM console in most hypervisors, (e.g. click anywhere in the console), the VM takes over the keyboard and mouse, and assumes that all button clicks are intended for the VM. This setting controls what key combination releases the mouse (and keyboard in most cases) from the VM console so that you can interact with other applications on the hypervisor host.

The final user setting is for configuring whether or not the client will use Enhanced Session Mode, if it is available. If you followed the previous instructions, you should have already disabled enhanced session mode under the server settings. For the sake of completion, you'll want to make sure that the *Use enhanced session mode* checkbox is unchecked under the user settings section, then click *Apply*.

The final option *Reset Check Boxes* allows you to restore all of the settings (Both *Server* and *User* settings) we configured here back to defaults if needed.

Virtual Switches

So now that we have the hypervisor configured, we need to set up the network infrastructure. Hyper-V uses virtual switches to signify different virtual network segments. These virtual switches come in three varieties: External, Internal, and Private.

Virtual Switch Types

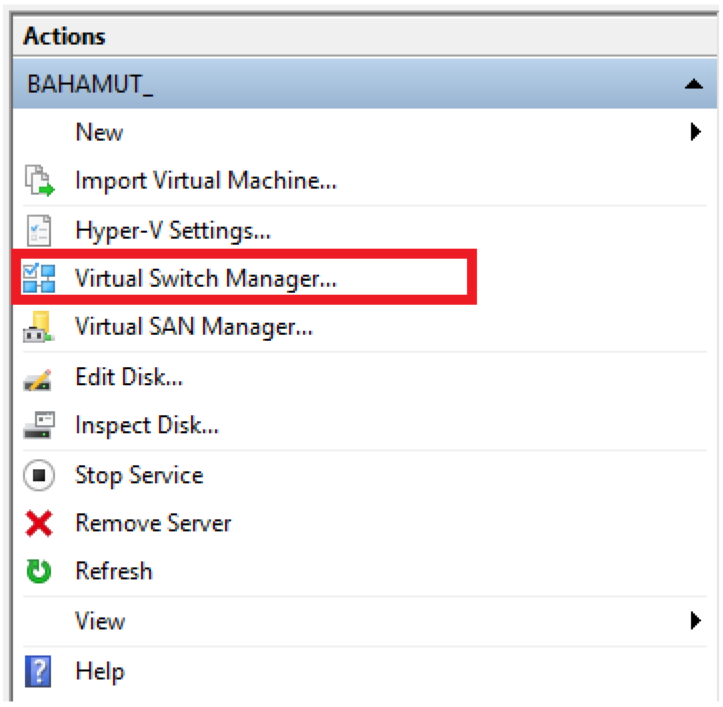
External virtual switches allow you to use your physical host system's network connection as an uplink port to connect any VMs connected to the External virtual switch to the same physical network your host system is connected to. This would allow remote systems on your network to interact with VMs hosted on your system. External virtual switches are similar in functionality to bridged network segments that other hosted hypervisors use.

Internal virtual switches create a standalone virtual network with no connection to the local network, or the internet. Hosts connected to an Internal virtual switch can only communicate with other hosts on that virtual switch, including the hypervisor host. The host system is able to talk to VMs on the Internal virtual switch through a virtual network card the hypervisor creates that is attached to the virtual switch. This virtual network card allows it the hypervisor host to communicate with VMs connected to the internal network switch over the network. As you might have guessed, Internal virtual switches are similar to "host-only" network segments.

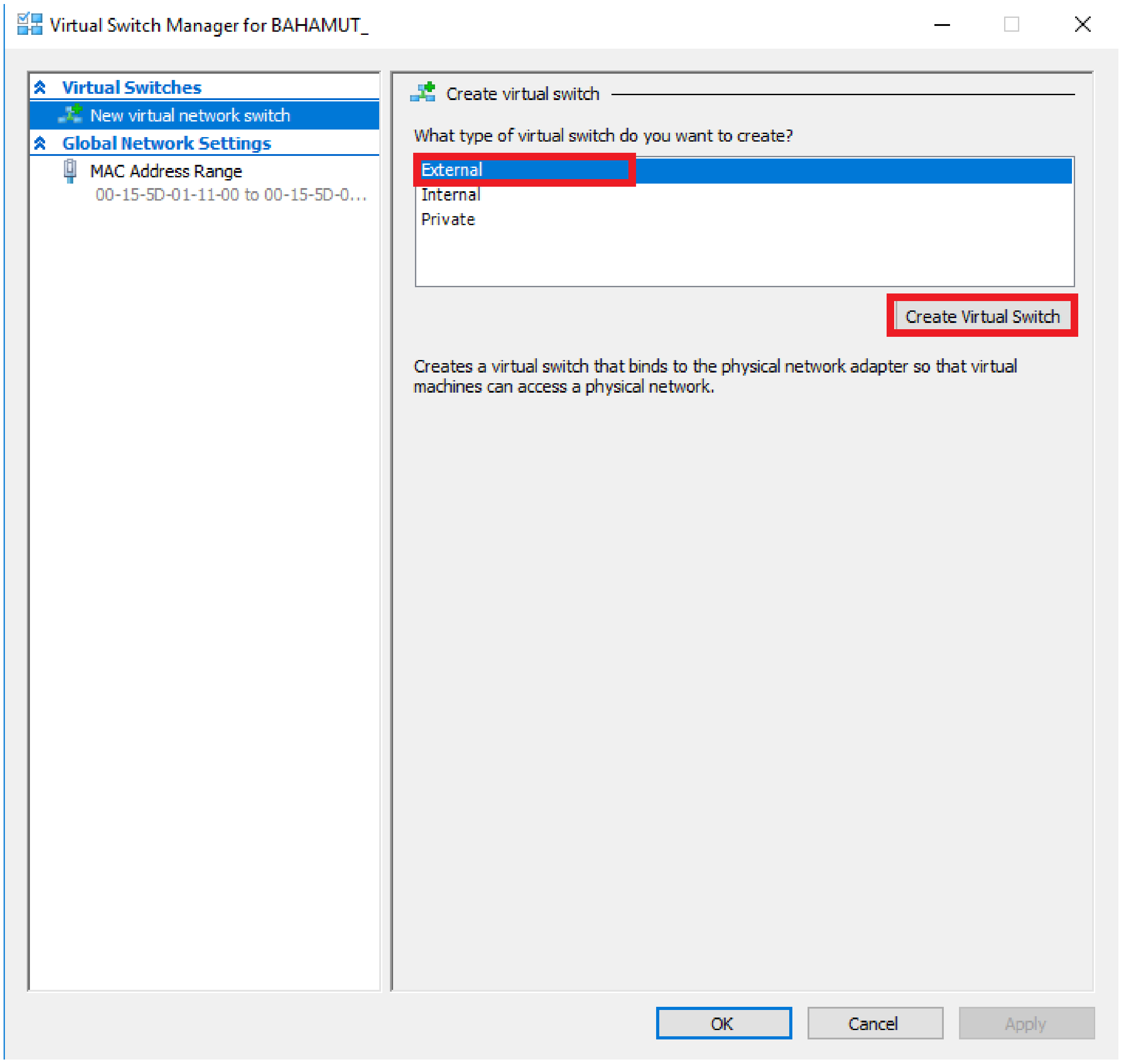
Private virtual switches are very similar to Internal virtual switches, in that they create a standalone network that only the hosted virtual machines can use to talk to one another directly. The main difference between an Internal and Private virtual switch is that the host system does not get a virtual network card, and therefore cannot communicate with VMs on a Private virtual switch directly.

Creating Virtual Switches Using the Virtual Switch Manager

So now that you know the difference, between the three types of virtual switches, we need to create 4 of them: 1 External, 1 Internal, and 2 Private switches. In the Hyper-V Manager main screen, on the right pane labeled *Actions*, click on *Virtual Switch Manager…* to get started.



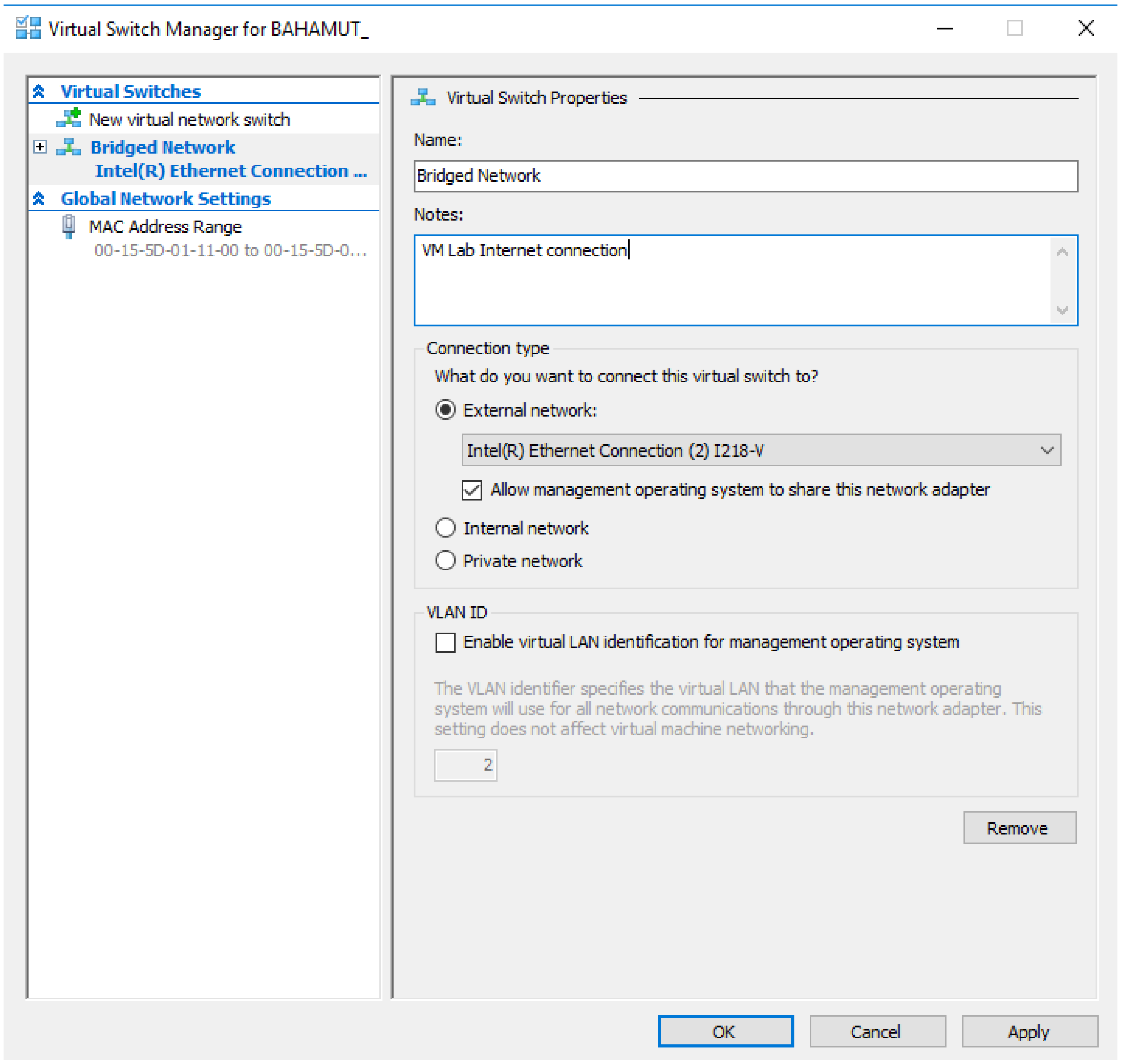
A new window titled *Virtual Switch Manager for [system name]* will appear. On the right pane, you will see the label *Create Virtual switch*, and a window that has the choices External, Internal, and Private. Click on External so that it is highlighted, and click the *Create Virtual Switch* button.


Virtual Switch Manager for BAHAMUT_
Virtual Switches
New virtual network switch
Global Network Settings
MAC Address Range
00-15-5D-01-11-00 to 00-15-5D-0...
Create virtual switch
What type of virtual switch do you want to create?
External
Internal
Private
Create Virtual Switch
OK
Cancel
Apply

Creates a virtual switch that binds to the physical network adapter so that virtual machines can access a physical network.

The screen will update with the new virtual switch you just created. Under virtual switch properties, there are two input boxes titled *Name:* and *Notes:*. This External virtual switch is our Bridged network in the network diagram, so name it something simple like “Bridged Network”. As for notes, be sure to input notes to remind you of what function the switch serves, something like, “VM lab internet connectivity”.

Underneath this, you will see the section titled *Connection type*. If you made the wrong type of switch and need to change its type, you can do so here. Note that under external network, there is a drop-down selection. This drop-down contains a list of all the network cards installed on the physical host. This determines which network card VMs attached to this External switch will use for access to the physical network (and/or internet). Underneath the drop-down list, is a checkbox labeled *Allow management operating system to share this network adapter*. If you want the host system to be able to access the physical network through the adapter you selected, this box must be checked.


Virtual Switch Manager for BAHAMUT_

Virtual Switches
New virtual network switch
Bridged Network
Intel(R) Ethernet Connection ...
Global Network Settings
MAC Address Range
00-15-5D-01-11-00 to 00-15-5D-0...
Virtual Switch Properties

Name: Bridged Network

Notes: VM Lab Internet connection

Connection type

What do you want to connect this virtual switch to?

[✓ External network:] Intel(R) Ethernet Connection (2) I218-V

[✓] Allow management operating system to share this network adapter

Internal network

Private network

VLAN ID

[ ] Enable virtual LAN identification for management operating system

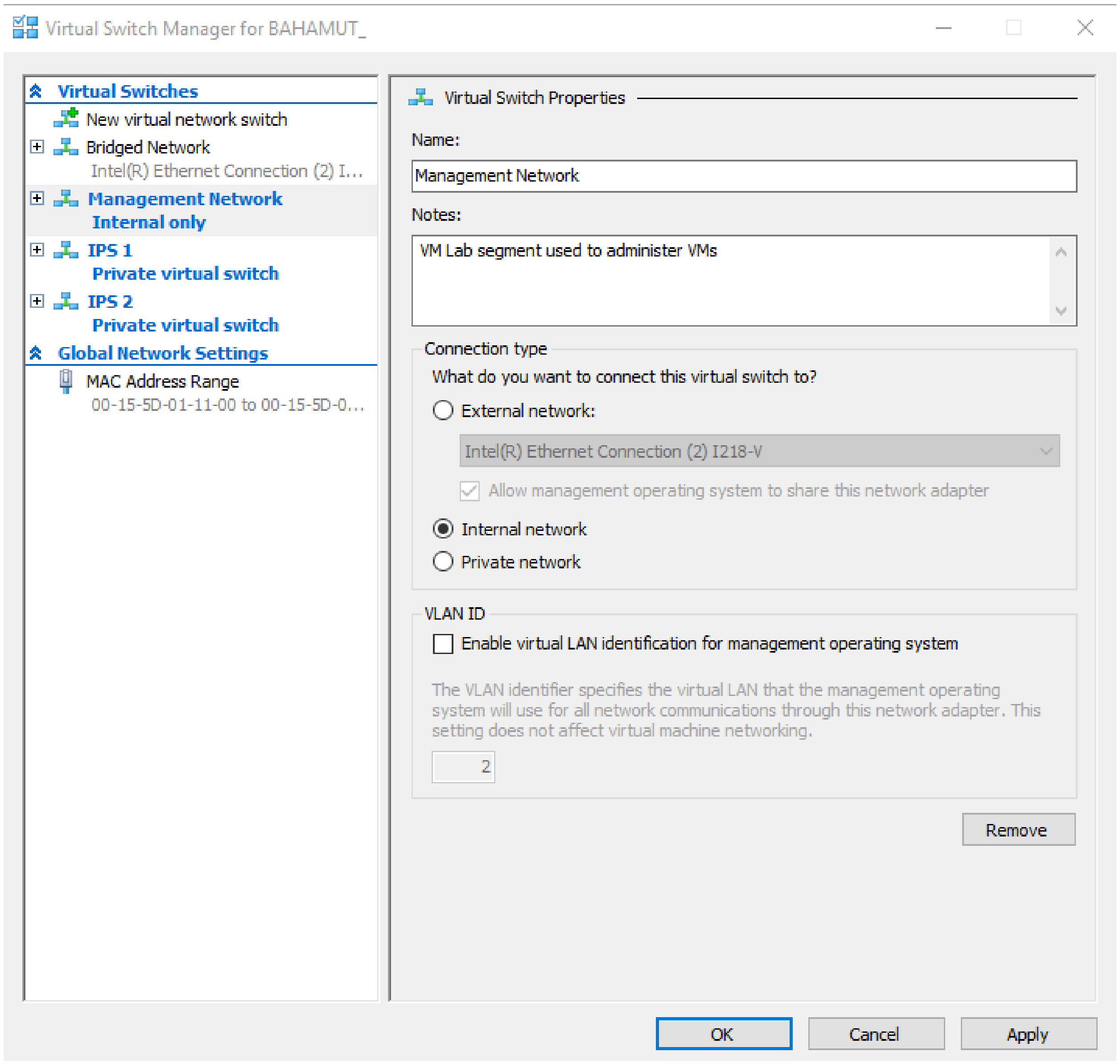
The VLAN identifier specifies the virtual LAN that the management operating system will use for all network communications through this network adapter. This setting does not affect virtual machine networking.

2

After you are done configuring the “Bridged Network” virtual switch, click on *New virtual network switch* on the left pane to return to the previous screen. Create the following:

Create one Internal virtual switch with the name “Management Network” and a description “VM lab segment used to administer VMs.”

Create two Private virtual switches with the names “IPS 1” and “IPS 2”. “IPS 1” should have the description “Connects firewall to IPS”. “IPS 2” should have the description “Connects IPS to vulnerable VMs”.When you are done you should have four virtual switches. The Virtual Switch Manager interface should look something like this:


Virtual Switch Manager for BAHAMUT_

Virtual Switches
- New virtual network switch
- Bridged Network — Intel(R) Ethernet Connection (2) I...
- Management Network — Internal only
- IPS 1 — Private virtual switch
- IPS 2 — Private virtual switch

Global Network Settings
- MAC Address Range — 00-15-5D-01-11-00 to 00-15-5D-0...

Virtual Switch Properties

Name: Management Network

Notes: VM Lab segment used to administer VMs

Connection type

What do you want to connect this virtual switch to?
- External network: Intel(R) Ethernet Connection (2) I218-V
  - Allow management operating system to share this network adapter
- [✓ Internal network]
- Private network

VLAN ID
- Enable virtual LAN identification for management operating system

The VLAN identifier specifies the virtual LAN that the management operating system will use for all network communications through this network adapter. This setting does not affect virtual machine networking.

2

Click the *Apply* button at the bottom right of the window to have Hyper-V create the new virtual switches, then click OK.

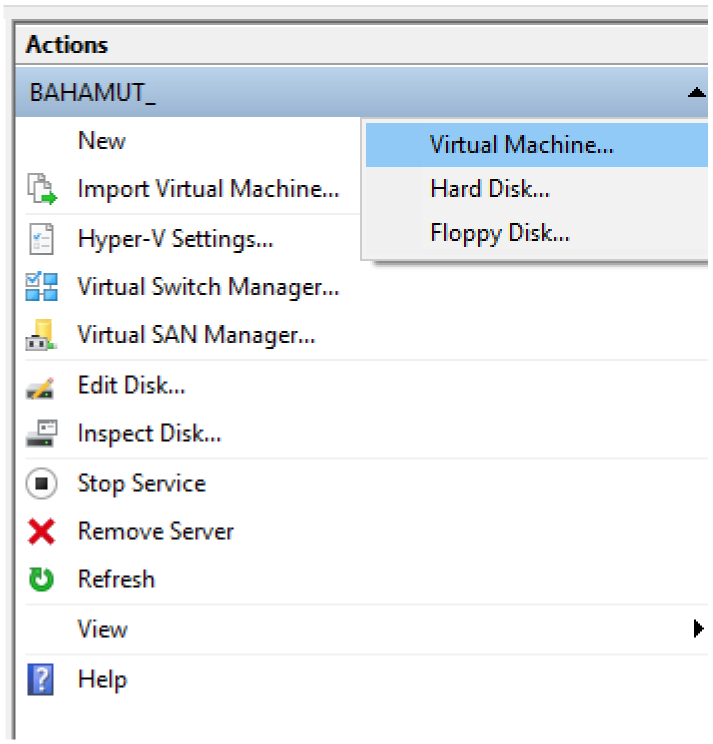
Creating the First VM, pfSense

pfSense is the keystone holding this entire configuration together. Personally, it's my favorite firewall distro due to ease of use, the amount of functionality it includes out of the box, combined with a plugin/add-on system for additional functionality. If you have the CPU, RAM, and disk, pfSense can easily be converted into a so-called “Next-Generation” firewall.

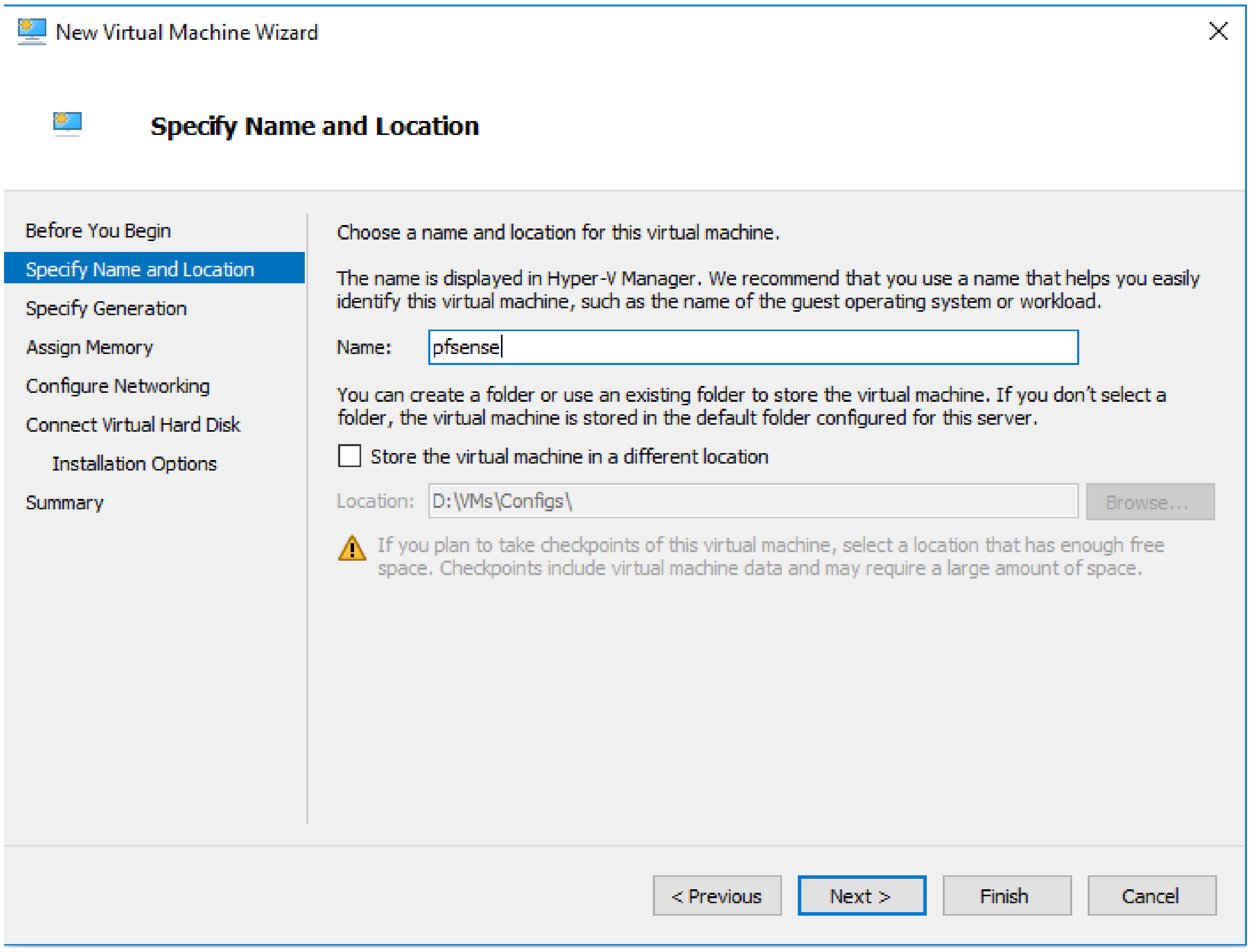
Make your way to https://www.pfsense.org/download/. Download the latest installation ISO for the amd64 architecture. While you’re at it, you may want to download a compression utility. The pfSense maintainers distribute pfSense as an ISO image file compressed with `gzip`. This means that we’ll need to decompress the ISO file at some point. On Windows, I prefer 7-Zip (http://www.7-zip.org/) as my compression utility of choice for decompressing files, since 7-Zip can handle zip, gzip, rar, and 7z files (among others) easily.

Adding a New VM

On the Hyper-V Manager window, on the left pane titled *Actions*, click *New* and a drop-down menu appears. Click on *Virtual Machine…* in the drop-down menu to start the *New Virtual Machine Wizard*.



The first screen has us specify a name for our new VM. We’ll keep it simple and name it “pfSense”. This screen also has an option to allow you to store the VM files in an alternate location if you need to. Click *Next*.


New Virtual Machine Wizard

Specify Name and Location

Choose a name and location for this virtual machine.

The name is displayed in Hyper-V Manager. We recommend that you use a name that helps you easily identify this virtual machine, such as the name of the guest operating system or workload.

Name: pfsense

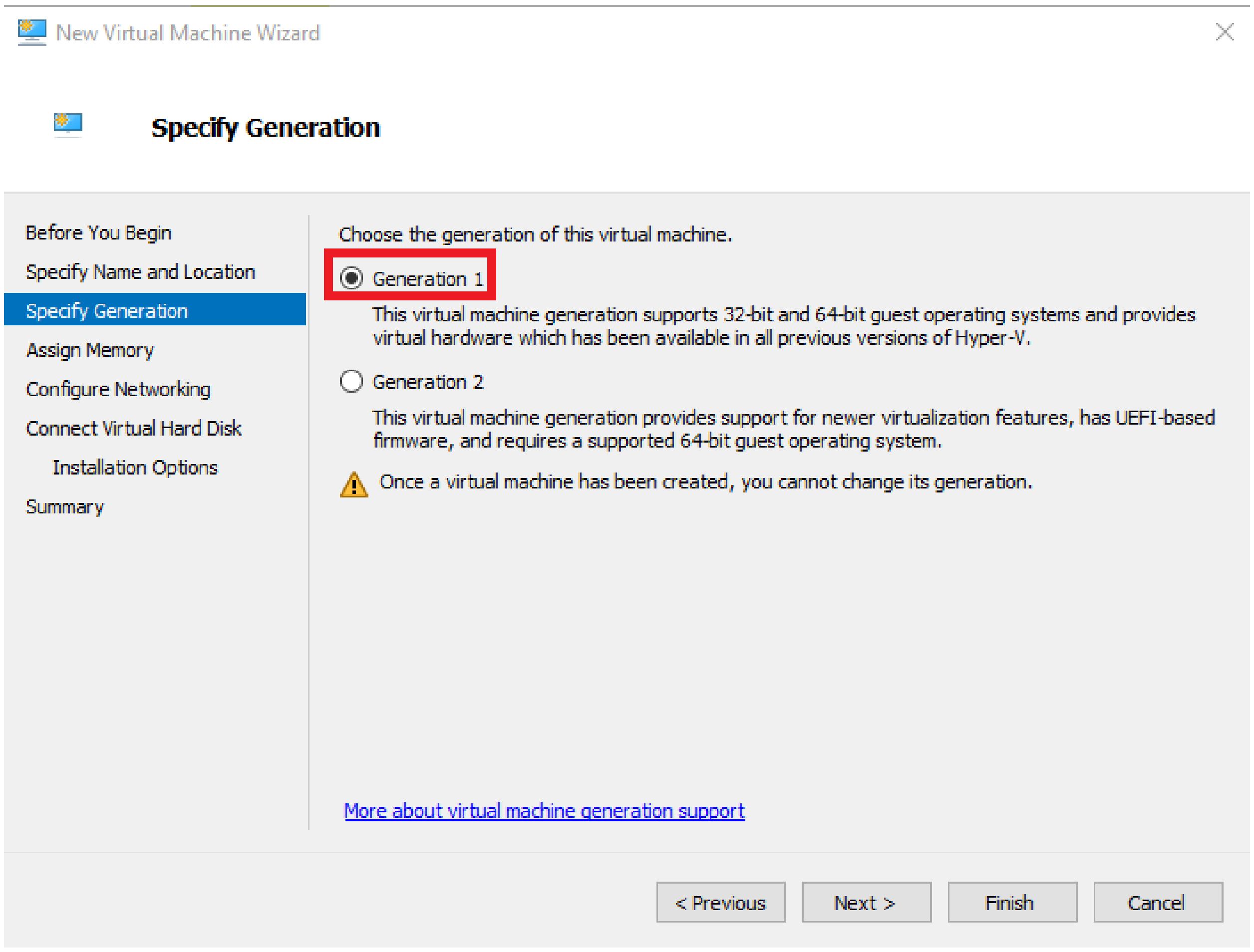
You can create a folder or use an existing folder to store the virtual machine. If you don't select a folder, the virtual machine is stored in the default folder configured for this server.

☐ Store the virtual machine in a different location

Location: D:\VMs\Configs\

If you plan to take checkpoints of this virtual machine, select a location that has enough free space. Checkpoints include virtual machine data and may require a large amount of space.

The next screen has us specify whether or not the virtual machine is a generation 1 or generation 2 VM. Without going into too many details, all of the VMs we will be creating will be generation 1 virtual machines. Click on the *Generation 1* radio button, then click *Next*.



Choose the generation of this virtual machine.

[✓ Generation 1]

This virtual machine generation supports 32-bit and 64-bit guest operating systems and provides virtual hardware which has been available in all previous versions of Hyper-V.

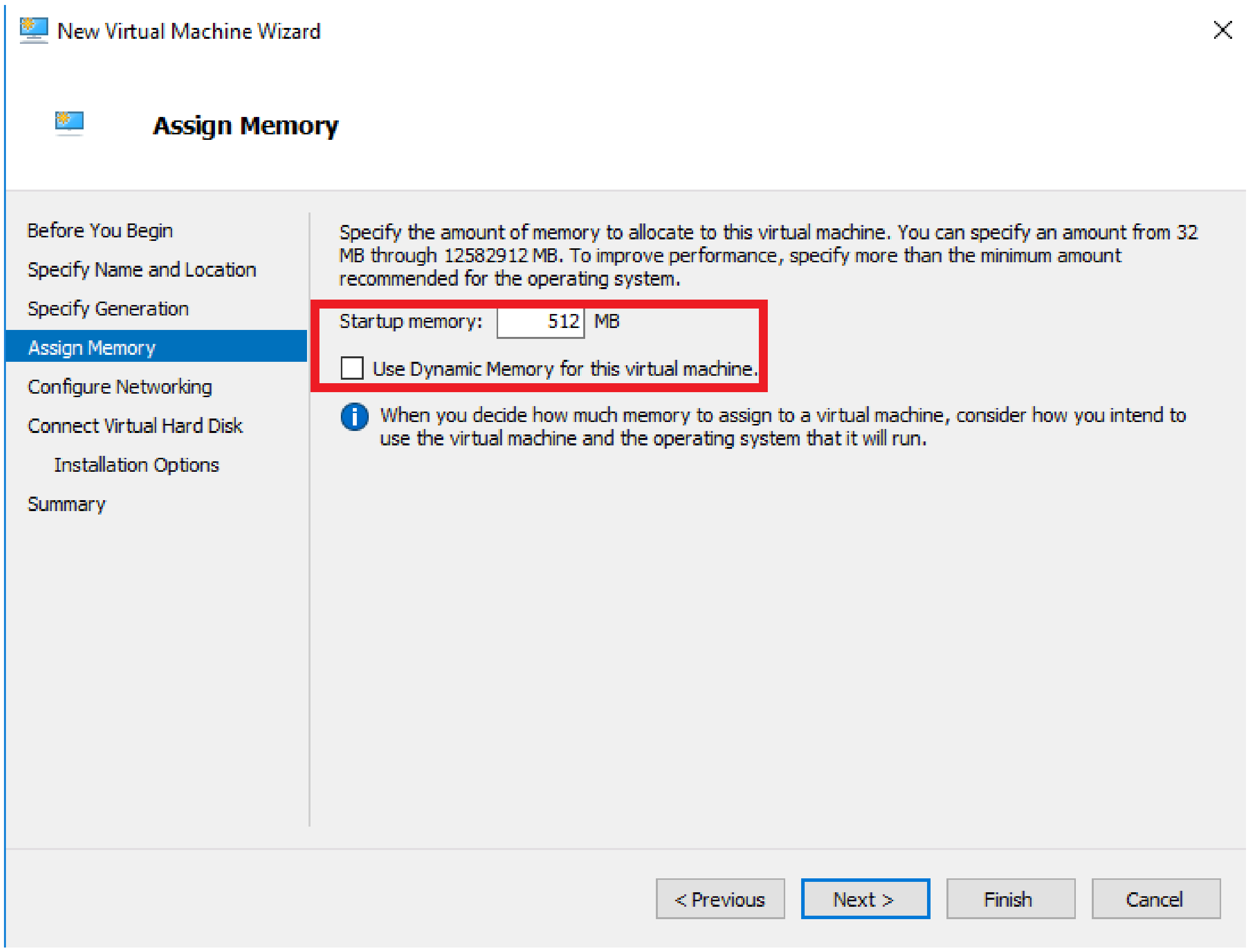
Generation 2

This virtual machine generation provides support for newer virtualization features, has UEFI-based firmware, and requires a supported 64-bit guest operating system.

Once a virtual machine has been created, you cannot change its generation.

Note: If you want to find out more about Generation 1 vs. Generation 2 VMs (including more OS support information) visit this page: https://technet.microsoft.com/windows-server-docs/compute/hyper-v/plan/should-i-create-a-generation-1-or-2-virtual-machine-in-hyper-v.

The next screen has us specify how much memory we want to allocate to our VM. Enter "512" in the input box labeled *Startup memory:* and uncheck the *Use Dynamic Memory for this virtual machine.* checkbox. Dynamic memory allows the virtual machine to allocate more RAM to itself as it sees fit. If you have a bit more RAM to play with, you could experiment with this feature, but for now, we're going to leave it disabled. When you are finished, click *Next*.



New Virtual Machine Wizard

Assign Memory

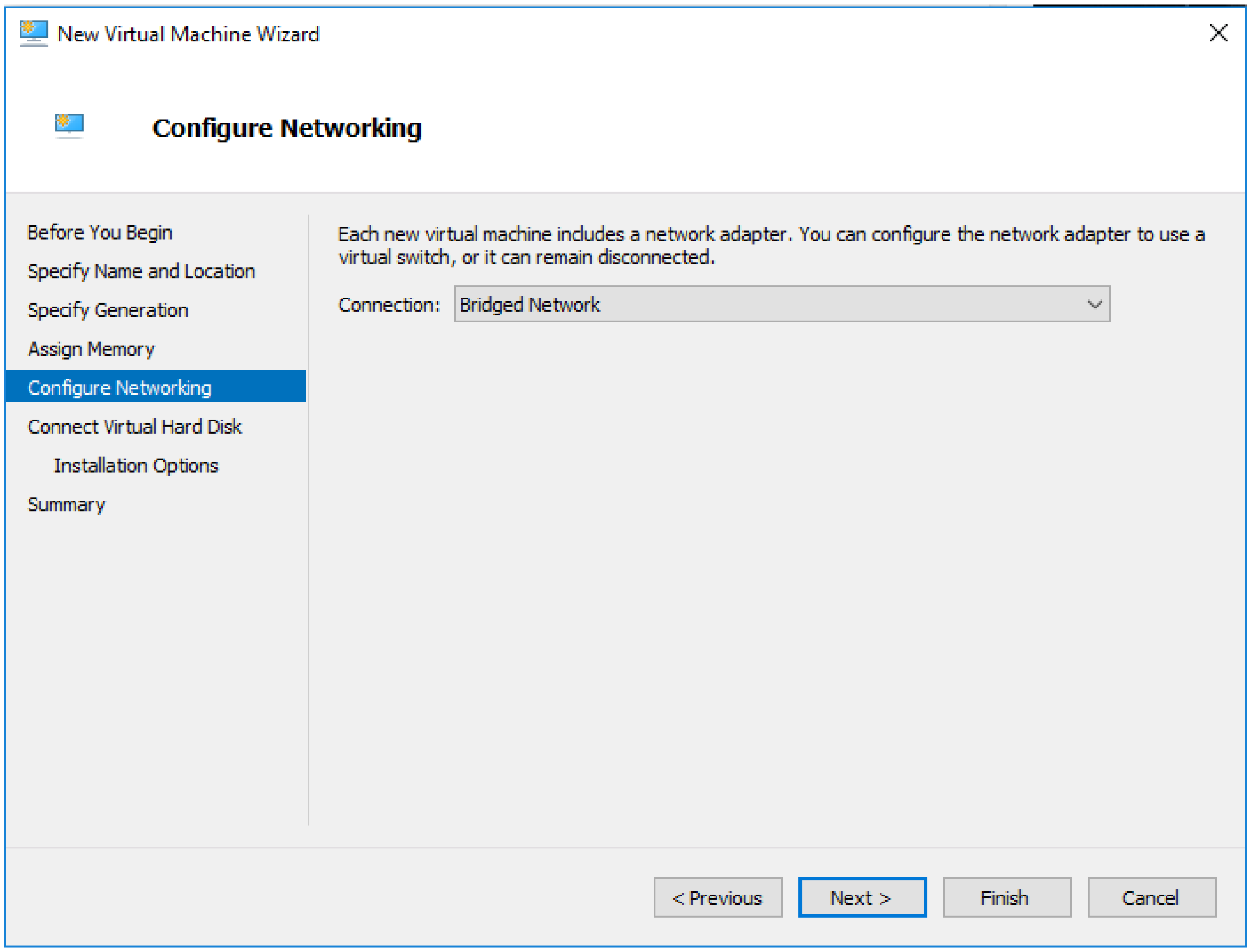
Specify the amount of memory to allocate to this virtual machine. You can specify an amount from 32 MB through 12582912 MB. To improve performance, specify more than the minimum amount recommended for the operating system.

Startup memory: 512 MB

☐ Use Dynamic Memory for this virtual machine.

When you decide how much memory to assign to a virtual machine, consider how you intend to use the virtual machine and the operating system that it will run.

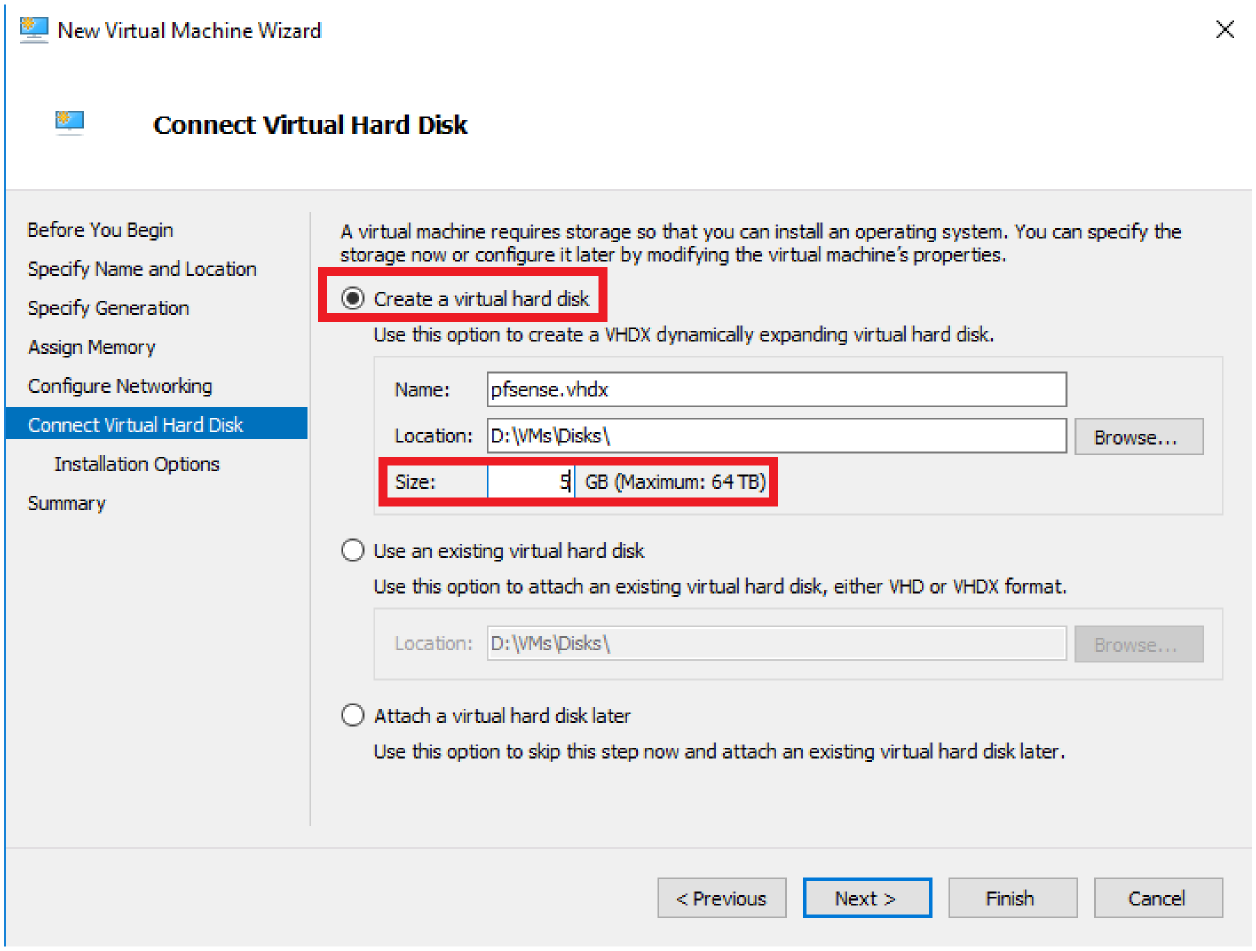
The next screen has us configure networking. Eventually we will have 3 network interfaces in total for this VM, but for now, select *Bridged Network* from the *Connection:* dropdown. Afterwards, click *Next*.



Each new virtual machine includes a network adapter. You can configure the network adapter to use a virtual switch, or it can remain disconnected.

Connection: Bridged Network

The next screen has us allocate disk space to create a virtual hard disk for to install our VM's OS and files. Select the *Create a virtual hard disk* radio button. Unless you have a compelling reason, the vhdx filename and location should not need to be changed. The *Size:* input box should be changed to "5" to allocate 5GB of disk space to this virtual machine. After altering the size of the virtual hard disk, click *Next*.



A virtual machine requires storage so that you can install an operating system. You can specify the storage now or configure it later by modifying the virtual machine's properties.

[✓ Create a virtual hard disk]

Use this option to create a VHDX dynamically expanding virtual hard disk.

Name: pfsense.vhdx

Location: D:\VMs\Disks\

Size: 5 GB (Maximum: 64 TB)

Use an existing virtual hard disk

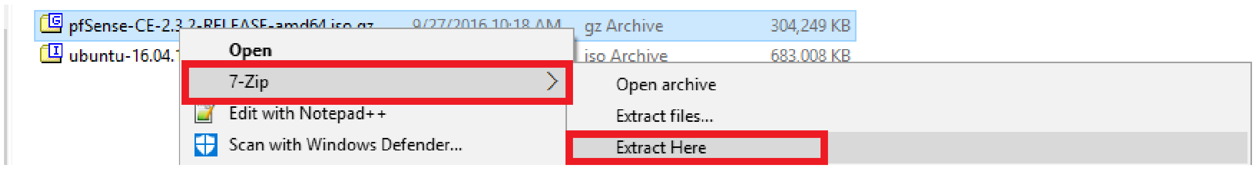
Use this option to attach an existing virtual hard disk, either VHD or VHDX format.

Location: D:\VMs\Disks\

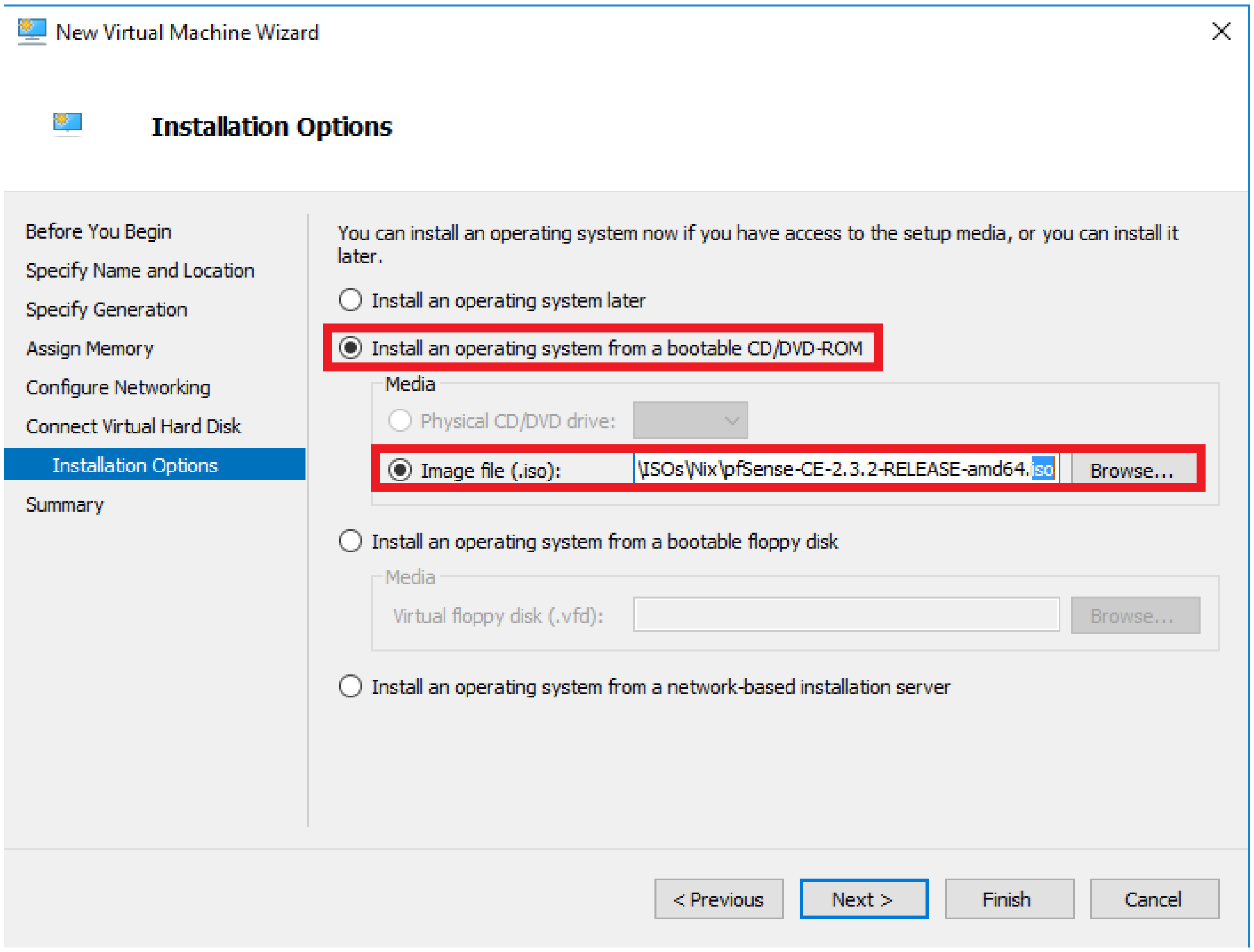
Attach a virtual hard disk later

Use this option to skip this step now and attach an existing virtual hard disk later.

Before we move on to the next screen, if you haven't already downloaded your pfSense ISO I would suggest doing so now. Please be aware that the pfSense ISO is `gzip` compressed and will need to be extracted/decompressed for Hyper-V to use it. To decompress your pfSense ISO using 7-Zip, open file explorer, and navigate to where you download the pfSense ISO file. Right click on the file, choose the *7-Zip* from the context menu, and the *Extract Here* option. This will extract the gzipped ISO for you and place in the same directory as the original gzipped file you downloaded.



On the next screen in the New Virtual Machine Wizard, labeled *Installation Options*, Click the radio button *Install an operating system from a bootable CD/DVD-ROM*. This will make the *Media* portion of the screen active. Click on the radio button labeled *Image file (.iso):* then click browse to open the file explorer and locate the unzipped pfSense ISO. After you have select the pfSense ISO, click Next.

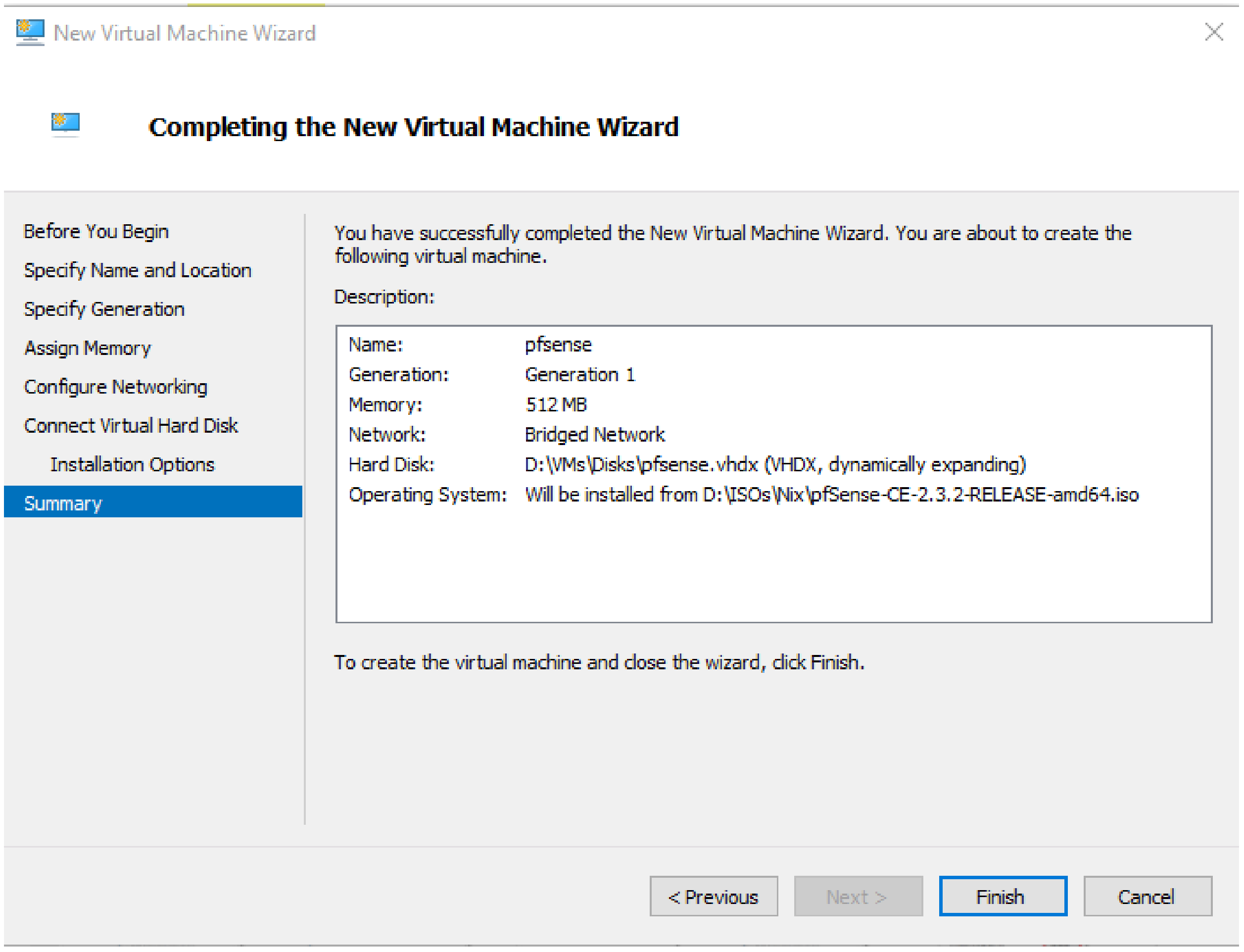

New Virtual Machine Wizard

Installation Options

You can install an operating system now if you have access to the setup media, or you can install it later.

- ○ Install an operating system later
- ● Install an operating system from a bootable CD/DVD-ROM
  - Media
  - ○ Physical CD/DVD drive:
  - ● Image file (.iso): \ISOs\Nix\pfSense-CE-2.3.2-RELEASE-amd64.iso
- ○ Install an operating system from a bootable floppy disk
  - Media
  - Virtual floppy disk (.vfd):
- ○ Install an operating system from a network-based installation server

The final screen verifies the settings we used to create our first VM. Confirm that your VM's configuration looks similar to what is presented on this screen, then click *Finish*.



Completing the New Virtual Machine Wizard

You have successfully completed the New Virtual Machine Wizard. You are about to create the following virtual machine.

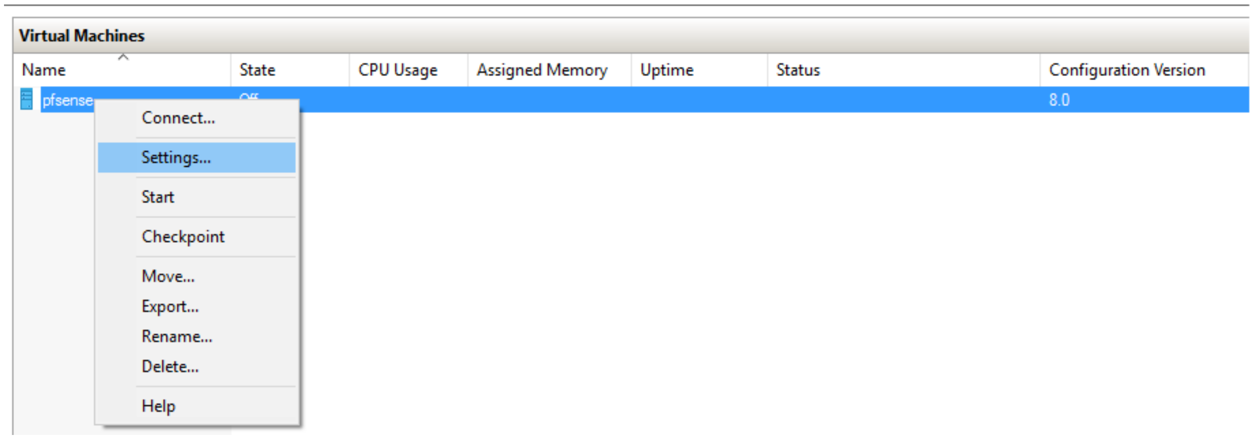
Description:

Name:	pfsense
Generation:	Generation 1
Memory:	512 MB
Network:	Bridged Network
Hard Disk:	D:\VMs\Disks\pfsense.vhdx (VHDX, dynamically expanding)
Operating System:	Will be installed from D:\ISOs\Nix\pfSense-CE-2.3.2-RELEASE-amd64.iso

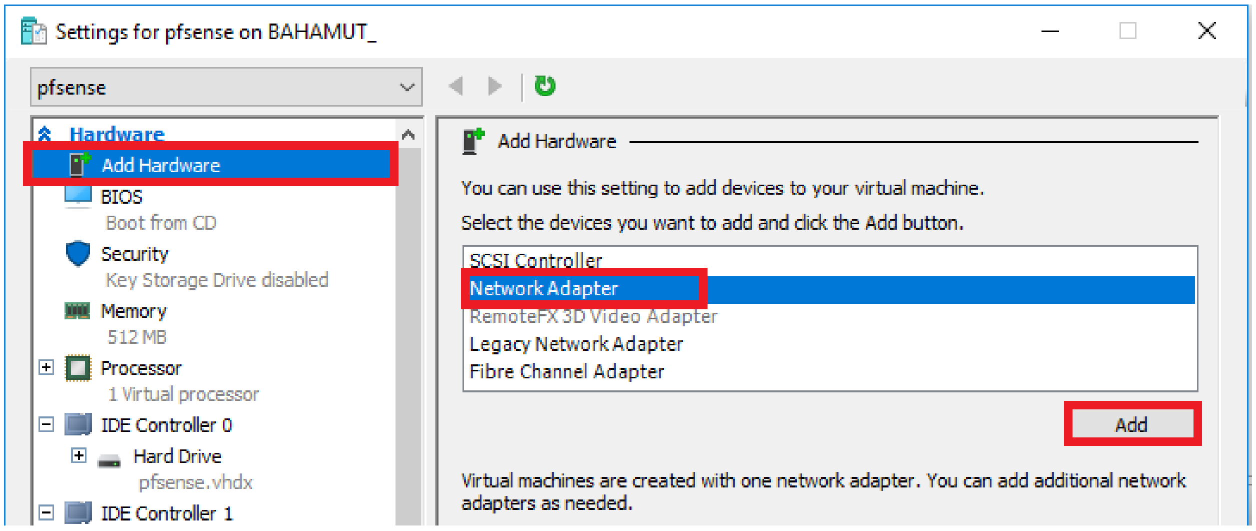
To create the virtual machine and close the wizard, click Finish.

Initial VM Settings

So now we have our first VM created. However, before we start it to install its operating system, there are a few more adjustments we have to make. In the Hyper-V Manager, under the *Virtual Machines* pane, right click on pfSense, and select *Settings*.



In the settings window for pfSense, The first thing we're going to do is add new hardware. Specifically, we have to add two more network adapters. Under *Hardware* in the left pane, click on *Add Hardware* to highlight, then on the right pane, select *Network Adapter* so that it is highlighted, then click the *Add* button.

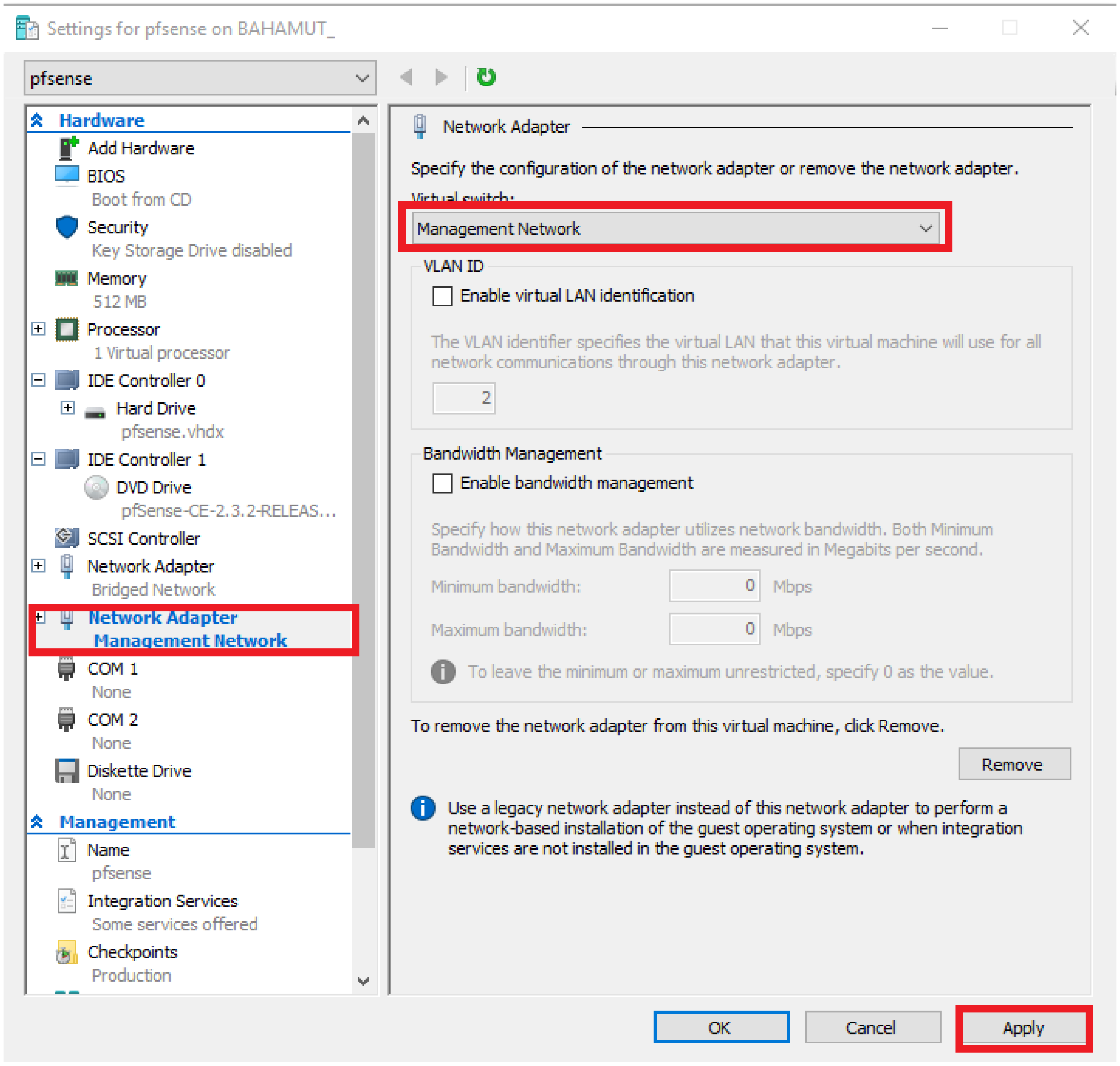


You can use this setting to add devices to your virtual machine.

Select the devices you want to add and click the Add button.

Virtual machines are created with one network adapter. You can add additional network adapters as needed.

A new network adapter should be added to your VM underneath the first network adapter we connected to the *Bridged Network* when we created this virtual machine. You should automatically be brought to this new network adapter to configure it. On the right side of the screen, on the drop-down menu labeled *Virtual Switch:*, select *Management Network* then click *Apply*.


Settings for pfsense on BAHAMUT_

pfsense

Hardware
- Add Hardware
- BIOS
  - Boot from CD
- Security
  - Key Storage Drive disabled
- Memory
  - 512 MB
- Processor
  - 1 Virtual processor
- IDE Controller 0
  - Hard Drive
    - pfsense.vhdx
- IDE Controller 1
  - DVD Drive
    - pfSense-CE-2.3.2-RELEAS...
- SCSI Controller
- Network Adapter
  - Bridged Network
- **Network Adapter**
  - **Management Network**
- COM 1
  - None
- COM 2
  - None
- Diskette Drive
  - None

Management
- Name
  - pfsense
- Integration Services
  - Some services offered
- Checkpoints
  - Production

Network Adapter

Specify the configuration of the network adapter or remove the network adapter.

Virtual switch: Management Network

VLAN ID

☐ Enable virtual LAN identification

The VLAN identifier specifies the virtual LAN that this virtual machine will use for all network communications through this network adapter.

2

Bandwidth Management

☐ Enable bandwidth management

Specify how this network adapter utilizes network bandwidth. Both Minimum Bandwidth and Maximum Bandwidth are measured in Megabits per second.

Minimum bandwidth: 0 Mbps

Maximum bandwidth: 0 Mbps

To leave the minimum or maximum unrestricted, specify 0 as the value.

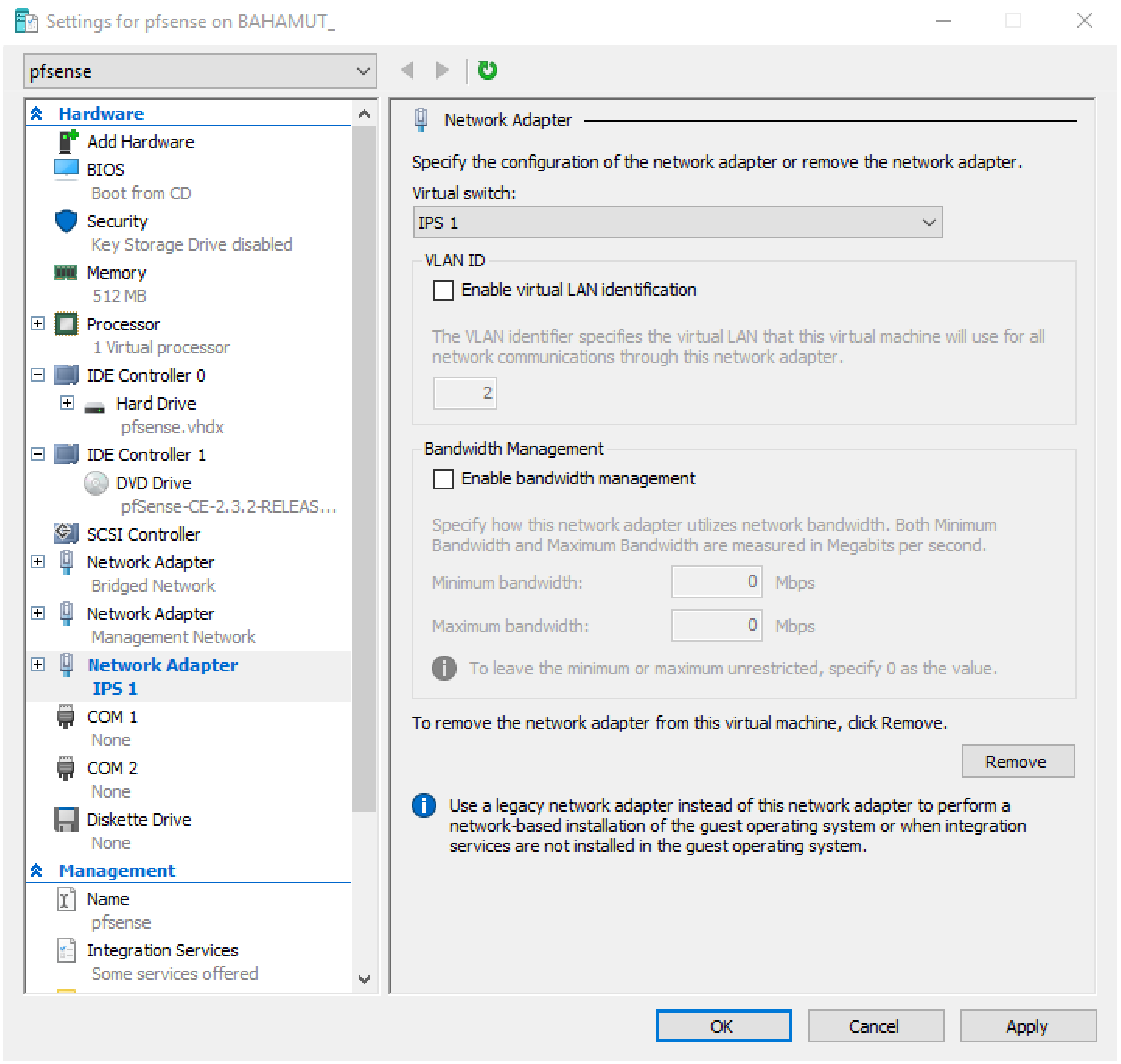
To remove the network adapter from this virtual machine, click Remove.

Remove

Use a legacy network adapter instead of this network adapter to perform a network-based installation of the guest operating system or when integration services are not installed in the guest operating system.

OK | Cancel | Apply

Click on *Add Hardware* again to highlight, select *Network Adapter*, then click the *Add* button again to create the third and final network card. This time we're going to select *IPS 1* for the *Virtual switch:* drop-down. Click *Apply* once more.



Network Adapter

Specify the configuration of the network adapter or remove the network adapter.

Virtual switch: IPS 1

VLAN ID

☐ Enable virtual LAN identification

The VLAN identifier specifies the virtual LAN that this virtual machine will use for all network communications through this network adapter.

2

Bandwidth Management

☐ Enable bandwidth management

Specify how this network adapter utilizes network bandwidth. Both Minimum Bandwidth and Maximum Bandwidth are measured in Megabits per second.

Minimum bandwidth: 0 Mbps

Maximum bandwidth: 0 Mbps

To leave the minimum or maximum unrestricted, specify 0 as the value.

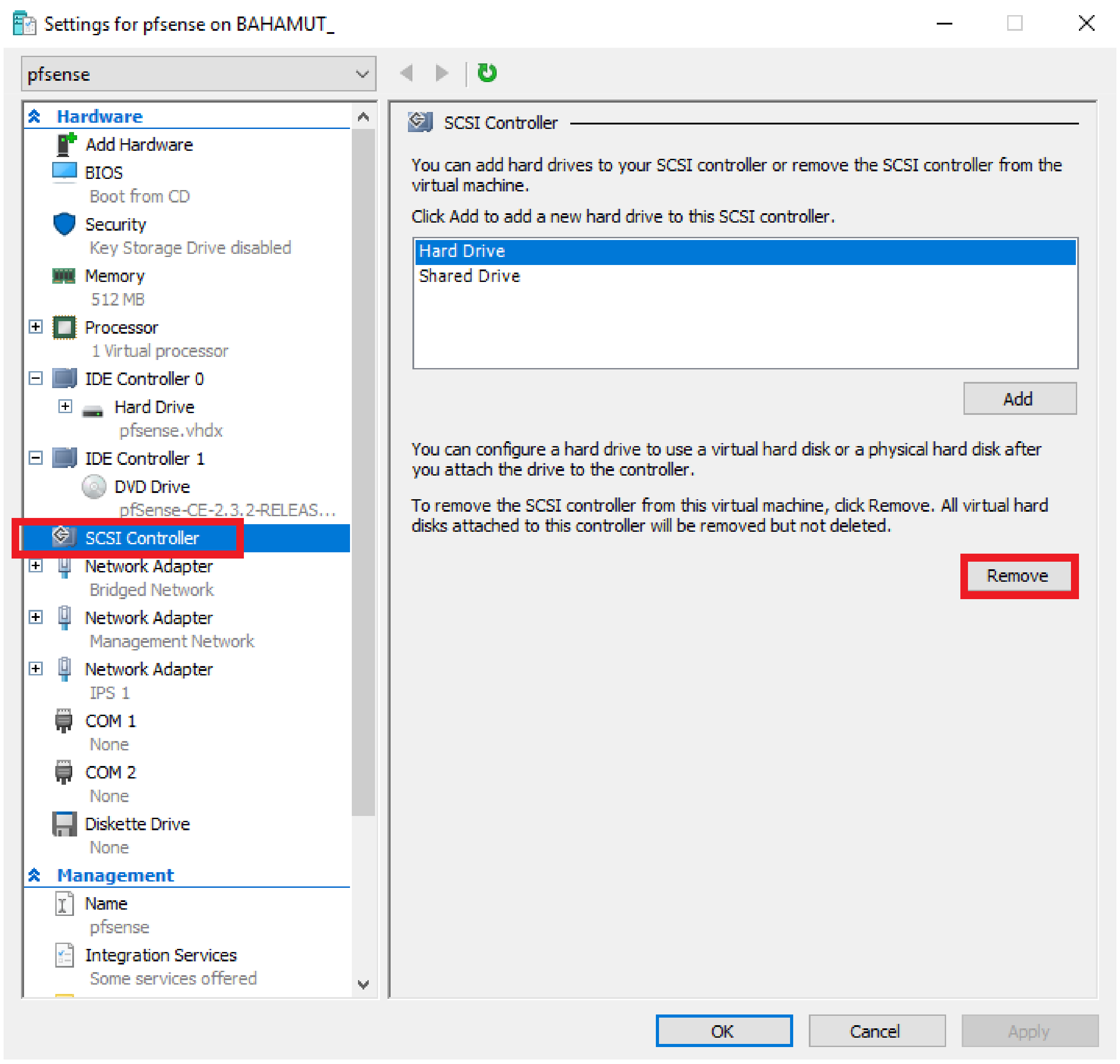
To remove the network adapter from this virtual machine, click Remove.

Remove

Use a legacy network adapter instead of this network adapter to perform a network-based installation of the guest operating system or when integration services are not installed in the guest operating system.

At this point, our pfSense VM should have three network cards attached to three different virtual switches.

Next, click on *SCSI Controller* under the *Hardware* list on the left pane, and click the *Remove* button that appears. We're removing the SCSI controller because our VM has no use for it. Afterwards, click *Apply*.



Settings for pfsense on BAHAMUT_

pfsense

SCSI Controller

You can add hard drives to your SCSI controller or remove the SCSI controller from the virtual machine.

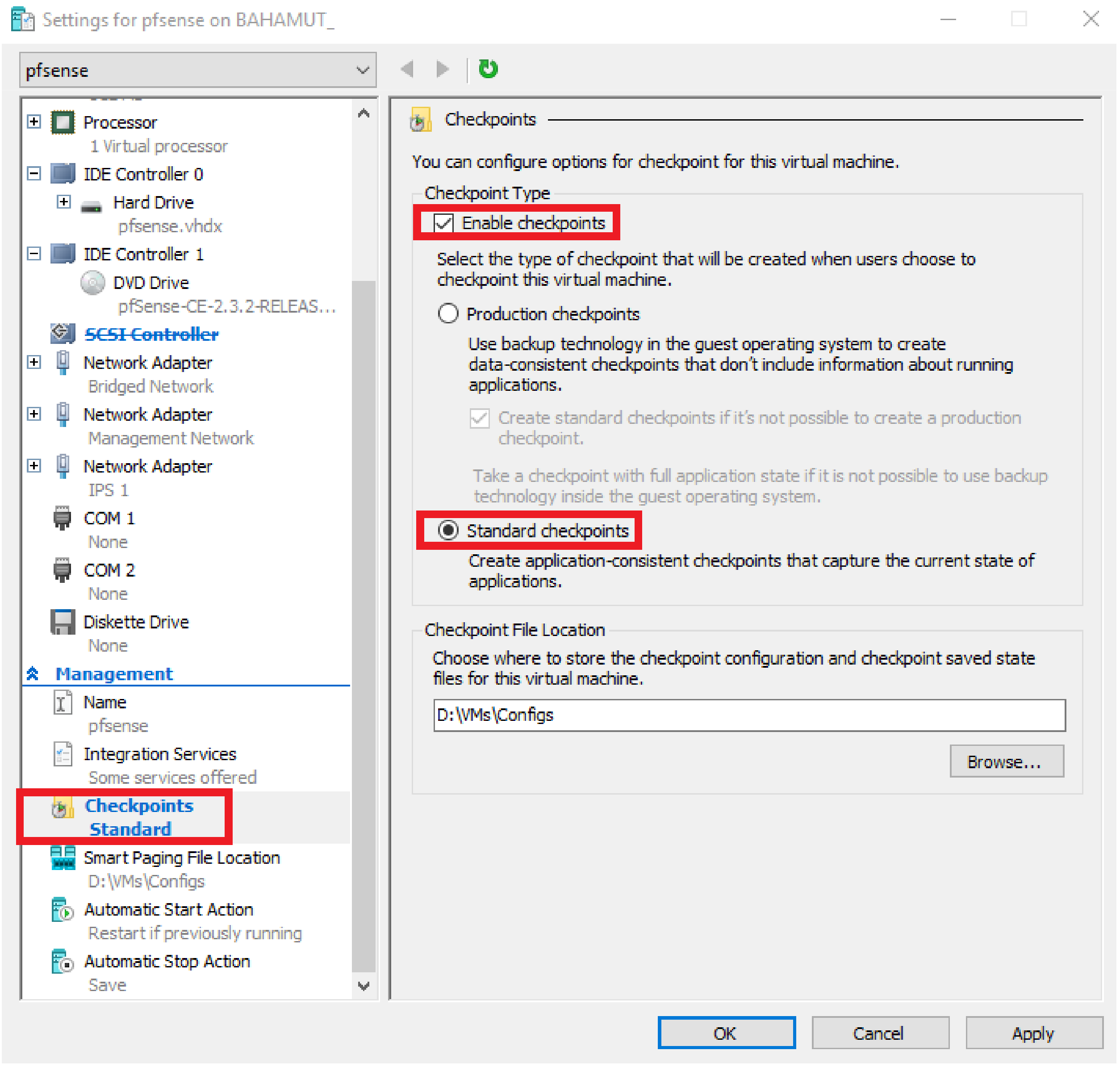
Click Add to add a new hard drive to this SCSI controller.

Hard Drive
Shared Drive

You can configure a hard drive to use a virtual hard disk or a physical hard disk after you attach the drive to the controller.

To remove the SCSI controller from this virtual machine, click Remove. All virtual hard disks attached to this controller will be removed but not deleted.

Scroll down on the left pane under *Management*, and click on *Checkpoints*. Under the section labeled *Checkpoint Type*, ensure the *Enable checkpoints* checkbox is checked, click on the *Standard checkpoints* radio button, then click *Apply*.


Settings for pfsense on BAHAMUT_

pfsense

- Processor: 1 Virtual processor
- IDE Controller 0
  - Hard Drive: pfsense.vhdx
- IDE Controller 1
  - DVD Drive: pfSense-CE-2.3.2-RELEAS...
- ~~SCSI Controller~~
- Network Adapter: Bridged Network
- Network Adapter: Management Network
- Network Adapter: IPS 1
- COM 1: None
- COM 2: None
- Diskette Drive: None
- Management
  - Name: pfsense
  - Integration Services: Some services offered
  - Checkpoints: Standard
  - Smart Paging File Location: D:\VMs\Configs
  - Automatic Start Action: Restart if previously running
  - Automatic Stop Action: Save

Checkpoints

You can configure options for checkpoint for this virtual machine.

Checkpoint Type

[✓] Enable checkpoints

Select the type of checkpoint that will be created when users choose to checkpoint this virtual machine.

( ) Production checkpoints

Use backup technology in the guest operating system to create data-consistent checkpoints that don't include information about running applications.

[✓] Create standard checkpoints if it's not possible to create a production checkpoint.

Take a checkpoint with full application state if it is not possible to use backup technology inside the guest operating system.

[● Standard checkpoints]

Create application-consistent checkpoints that capture the current state of applications.

Checkpoint File Location

Choose where to store the checkpoint configuration and checkpoint saved state files for this virtual machine.

D:\VMs\Configs

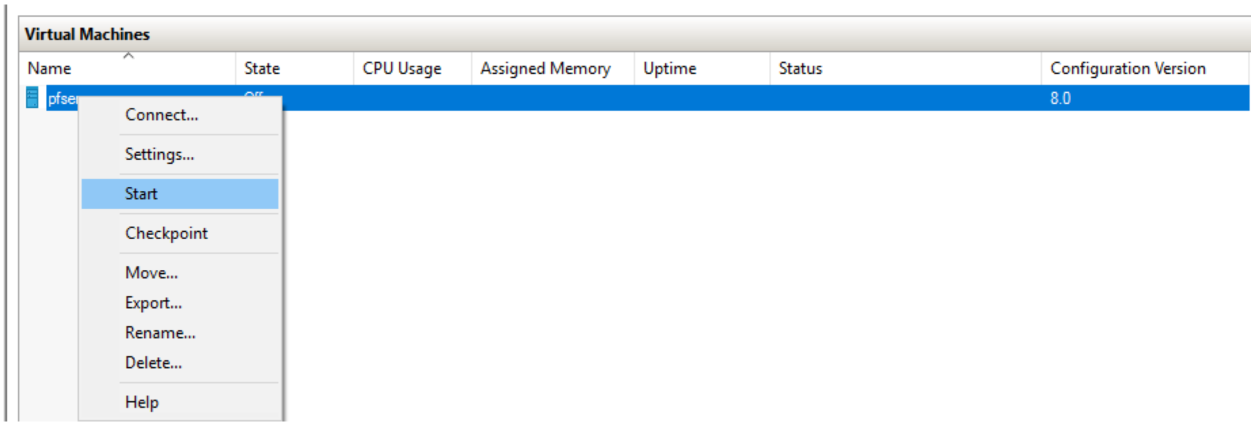
Without going too deep into it, checkpoints are essentially Hyper-V's terminology for snapshots, or a way to restore the VM back to a particular configuration in an instant. We will be using standard checkpoints for all of our VMs from here on out.

Note: If you are running Client Hyper-V on Windows 8.x, you may not have a choice between

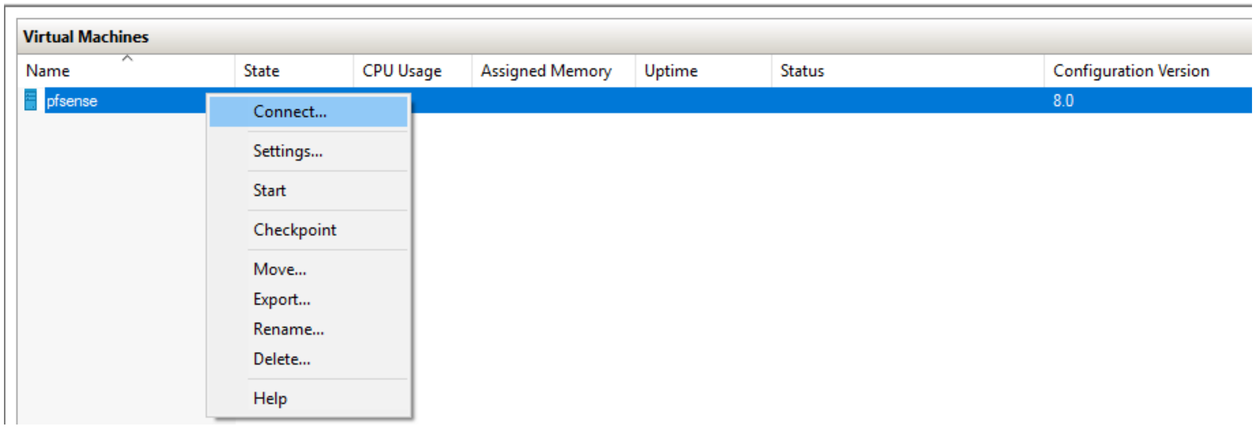
Production and *Standard* Checkpoints. You can disregard this section. Production checkpoints were added in with the Windows 10 version of Client Hyper-V. If you want to find out more about the differences between production and standard checkpoints, go here: https://technet.microsoft.com/en-us/windows-server-docs/compute/hyper-v/manage/choose-between-standard-or-production-checkpoints-in-hyper-v.

Installing pfSense

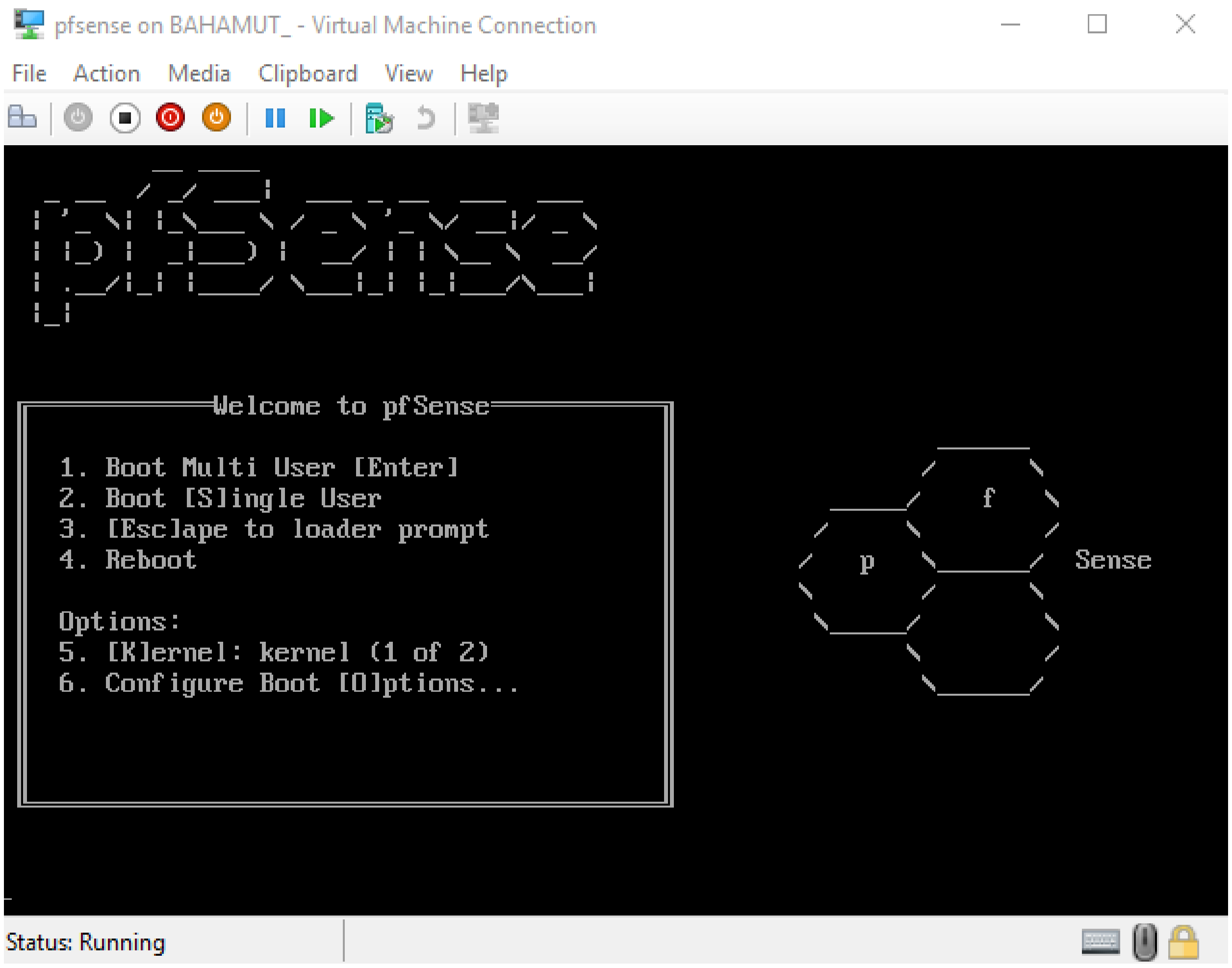
With our changes in place, the VM is ready to be powered on to have the operating system installed. Power on the VM by right clicking it, and selecting *Start* from the menu.



You should see the VM's *State* in the Hyper-V Manager console change from *Off* to *Running*. Right click on the VM again, and select *Connect…* to enable a console session to the virtual machine.

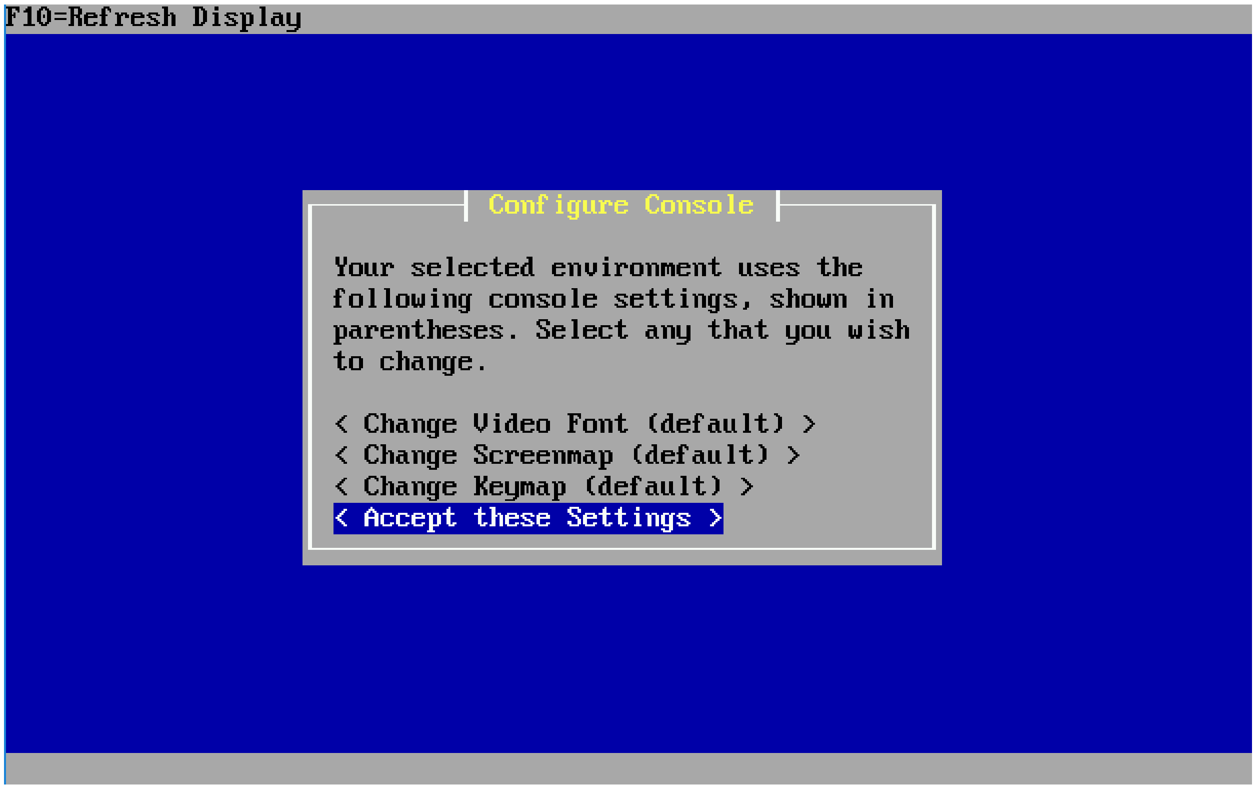


After booting the VM and attaching to the virtual machine's console, select option 1, *Boot Multi User [Enter]* by hitting the enter key.


pfsense on BAHAMUT_ - Virtual Machine Connection
File Action Media Clipboard View Help

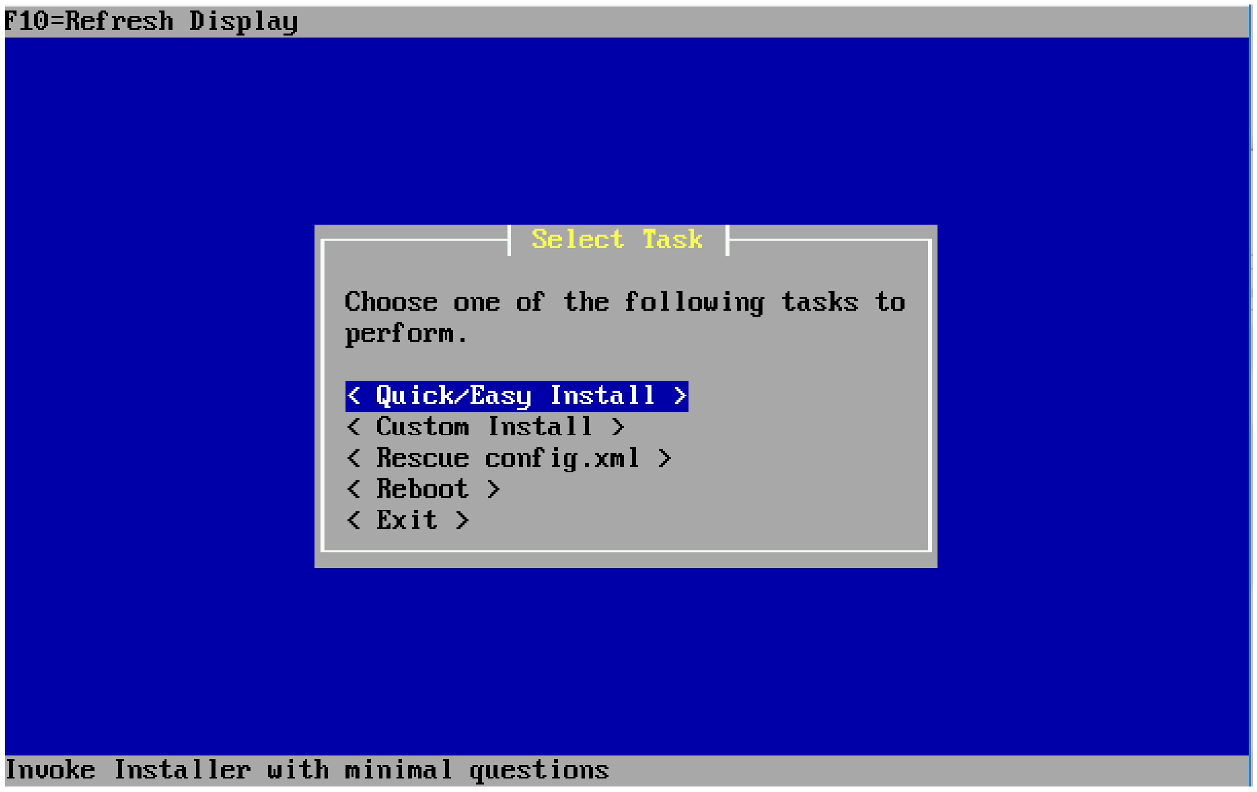
```
Welcome to pfSense

1. Boot Multi User [Enter]
2. Boot [S]ingle User
3. [Esc]ape to loader prompt
4. Reboot

Options:
5. [K]ernel: kernel (1 of 2)
6. Configure Boot [O]ptions...
```

Let pfSense boot up, and the system should automatically run the installer. Adjust your video, screenmap, and keymap settings as necessary, then select < *Accept these Settings* >.



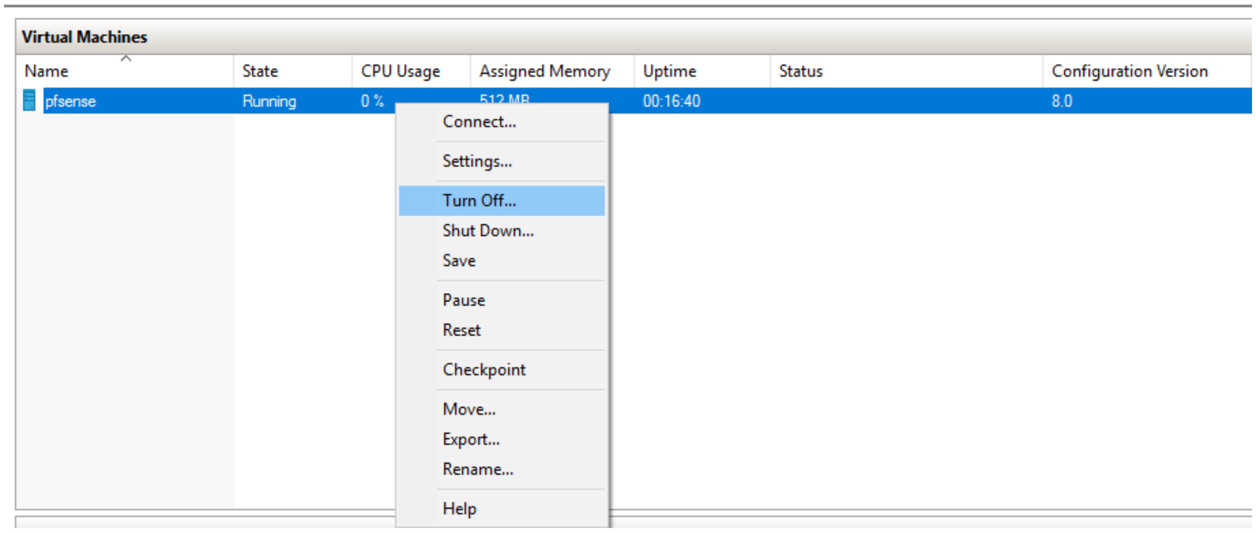
```
F10=Refresh Display

                 Configure Console

   Your selected environment uses the
   following console settings, shown in
   parentheses. Select any that you wish
   to change.

   < Change Video Font (default) >
   < Change Screenmap (default) >
   < Change Keymap (default) >
   < Accept these Settings >
```

On the next screen, select < *Quick/Easy Install* >, and let pfSense perform its installation routines. The next screen will inform you that the install will erase the contents of the hard disk. Since our virtual disk is already empty, this doesn't matter in the least. Select *OK*, and let the installer run.



```
F10=Refresh Display

                    Select Task

        Choose one of the following tasks to
        perform.

        < Quick/Easy Install >
        < Custom Install >
        < Rescue config.xml >
        < Reboot >
        < Exit >

Invoke Installer with minimal questions
```

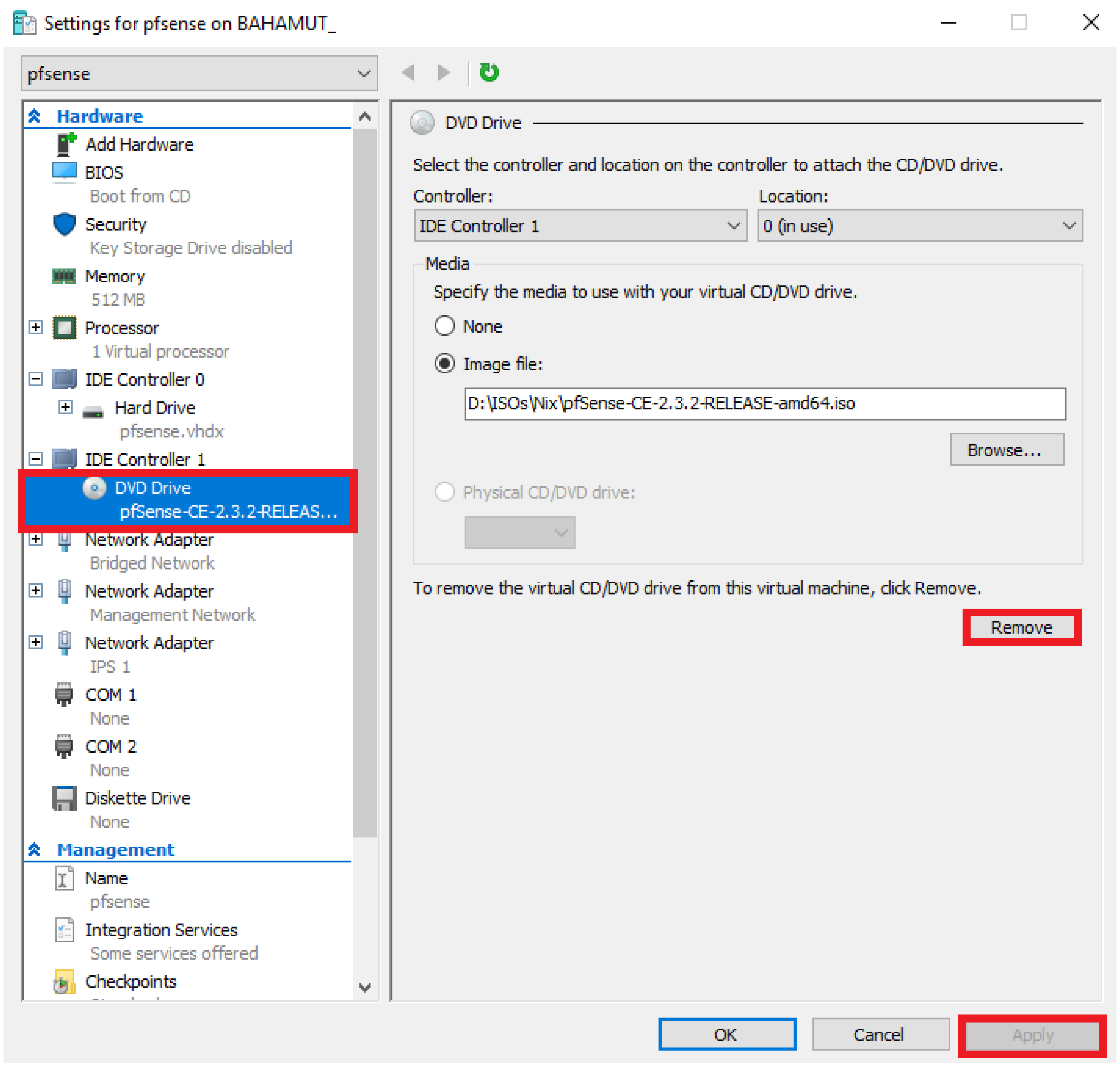
Eventually, you will be asked whether the standard kernel or the embedded one should be used. Be sure to select < *Standard Kernel* >. Finally, the installer informs you to reboot the machine to boot from the hard drive. Allow the virtual machine to reboot in order to finish the installation procedure. After system has rebooted, right click on the VM in the Hyper-V Manager, and select *Turn Off...*


Virtual Machines

Name	State	CPU Usage	Assigned Memory	Uptime	Status	Configuration Version
pfsense	Running	0 %	512 MB	00:16:40		8.0

Final VM Settings

There is one last configuration change we want to make to the pfSense VM's settings before we continue. Make sure that the VM has been turned off and that the State reads as *Off* in the Hyper-V Manager. Next, right click on the VM in the Virtual Machines pane, and select *Settings…* to bring up the settings menu. On the left pane under *Hardware*, select *DVD Drive*. If you don't see it, double click on *IDE Controller* 1 to bring it up under the IDE controller, then click on it. On the *DVD Drive* screen, select the *Remove* button to uninstall the virtual DVD drive entirely, then click *Apply*.


Settings for pfsense on BAHAMUT_

pfsense

Hardware
- Add Hardware
- BIOS: Boot from CD
- Security: Key Storage Drive disabled
- Memory: 512 MB
- Processor: 1 Virtual processor
- IDE Controller 0
  - Hard Drive: pfsense.vhdx
- IDE Controller 1
  - DVD Drive: pfSense-CE-2.3.2-RELEAS...
- Network Adapter: Bridged Network
- Network Adapter: Management Network
- Network Adapter: IPS 1
- COM 1: None
- COM 2: None
- Diskette Drive: None

Management
- Name: pfsense
- Integration Services: Some services offered
- Checkpoints

DVD Drive

Select the controller and location on the controller to attach the CD/DVD drive.

Controller: IDE Controller 1

Location: 0 (in use)

Media

Specify the media to use with your virtual CD/DVD drive.

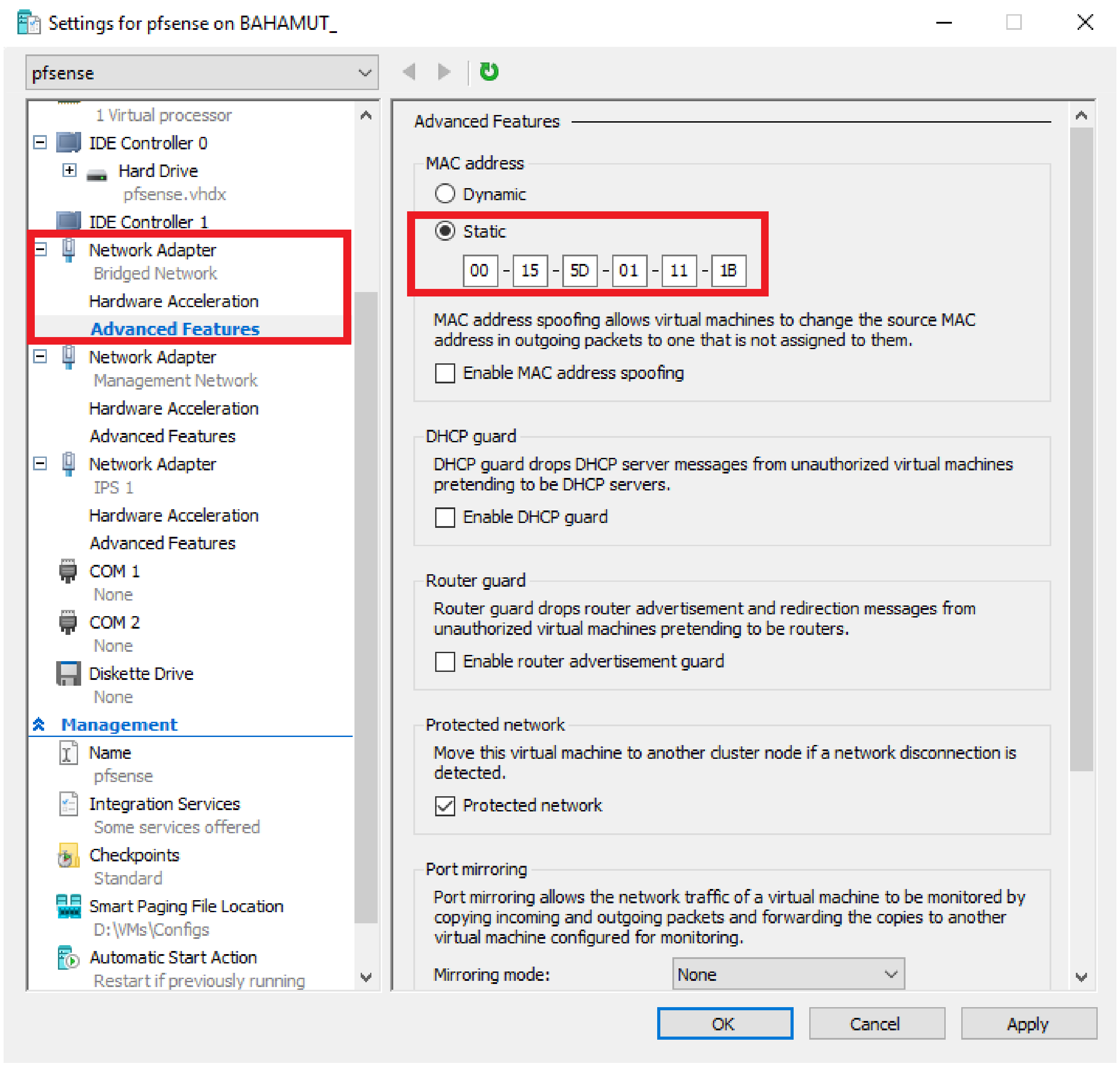
( ) None

(●) Image file: D:\ISOs\Nix\pfSense-CE-2.3.2-RELEASE-amd64.iso

( ) Physical CD/DVD drive:

To remove the virtual CD/DVD drive from this virtual machine, click Remove.

Before closing the settings menu for the pfSense VM, We need to record the MAC addresses for each of our network adapters. Under *Hardware*, on the left pane, look for the network adapter labeled *Bridged Network*. Next to the adapter, you'll notice a + sign. Either, click on the symbol, or double click on *Network Adapter* to display the *Hardware Acceleration*, and *Advanced Features* options. Select *Advanced Features* to display advanced settings for the network adapter. Under the section labeled *MAC address*, make sure to document or write down the MAC address displayed in the input boxes.


Settings for pfsense on BAHAMUT_

pfsense

- 1 Virtual processor
- IDE Controller 0
  - Hard Drive
    - pfsense.vhdx
- IDE Controller 1
- Network Adapter
  - Bridged Network
  - Hardware Acceleration
  - **Advanced Features**
- Network Adapter
  - Management Network
  - Hardware Acceleration
  - Advanced Features
- Network Adapter
  - IPS 1
  - Hardware Acceleration
  - Advanced Features
- COM 1
  - None
- COM 2
  - None
- Diskette Drive
  - None

Management

- Name
  - pfsense
- Integration Services
  - Some services offered
- Checkpoints
  - Standard
- Smart Paging File Location
  - D:\VMs\Configs
- Automatic Start Action
  - Restart if previously running

Advanced Features

MAC address

○ Dynamic

● Static

00 - 15 - 5D - 01 - 11 - 1B

MAC address spoofing allows virtual machines to change the source MAC address in outgoing packets to one that is not assigned to them.

☐ Enable MAC address spoofing

DHCP guard

DHCP guard drops DHCP server messages from unauthorized virtual machines pretending to be DHCP servers.

☐ Enable DHCP guard

Router guard

Router guard drops router advertisement and redirection messages from unauthorized virtual machines pretending to be routers.

☐ Enable router advertisement guard

Protected network

Move this virtual machine to another cluster node if a network disconnection is detected.

☑ Protected network

Port mirroring

Port mirroring allows the network traffic of a virtual machine to be monitored by copying incoming and outgoing packets and forwarding the copies to another virtual machine configured for monitoring.

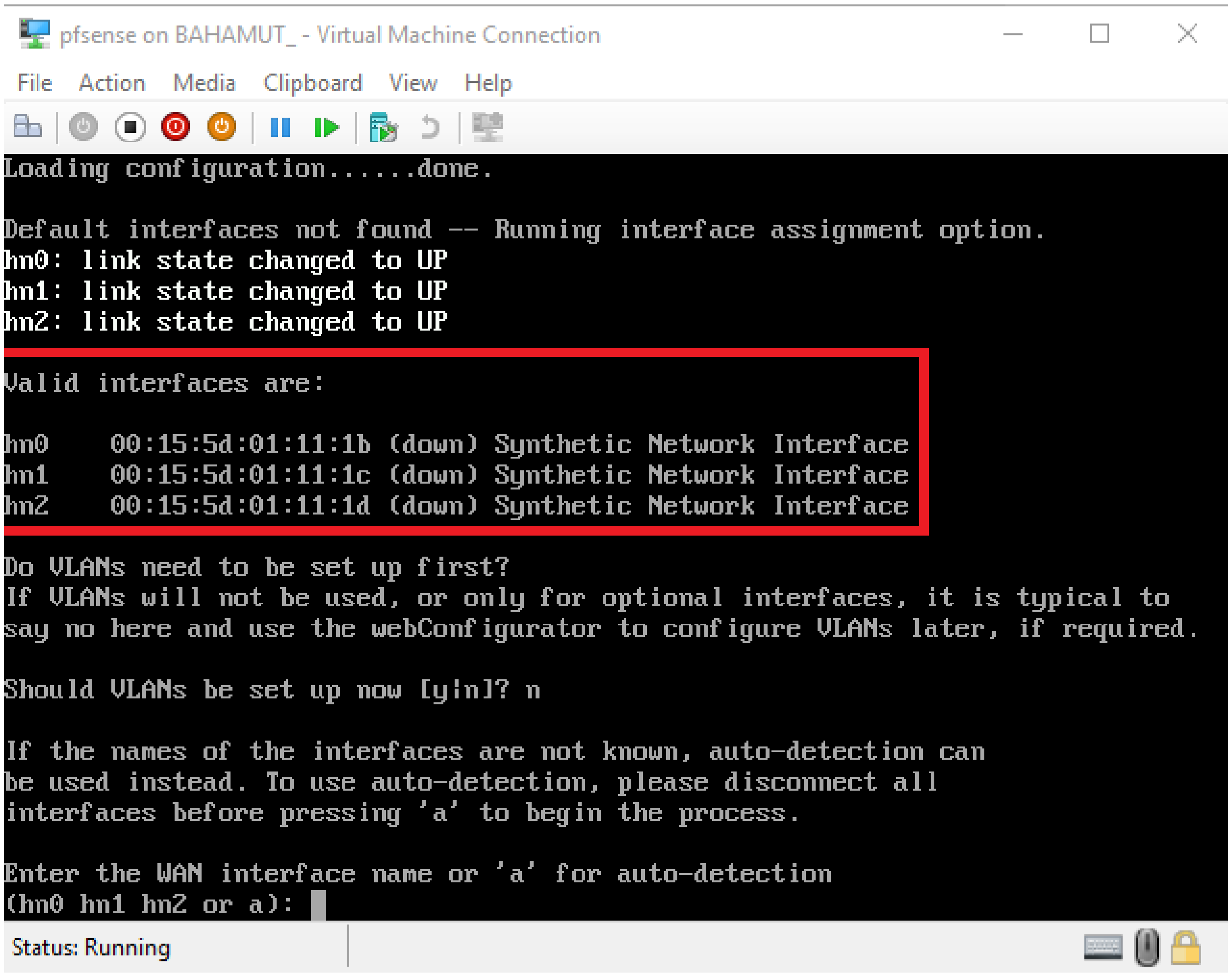
Mirroring mode: None

Repeat this process for the *Management Network* and *IPS 1* network adapters and record the MAC addresses for these interfaces.

Note: In the illustration above, I opted to change the MAC address setting from *Dynamic* To *Static*. You are not required to do this. When you first boot a VM in Hyper-V, the hypervisor will automatically generate a MAC address for you that begins with 00-15-5D (the OUI for “Microsoft”). If you select *Static*, You can overwrite the MAC address that Hyper-V assigns to the interface, to one of your choosing. You can use this to define a MAC address with a custom OUI. You can use this to feature to fool some malware and/or applications that attempt to check the MAC address to identify whether or not a system is a VM or not. **Keep this in mind for systems in which you plan on doing malware analysis.** For the time being however, I recommend leaving the “MAC address” radio button set to “Dynamic”, and recording the MAC address that appears.

Network Configuration

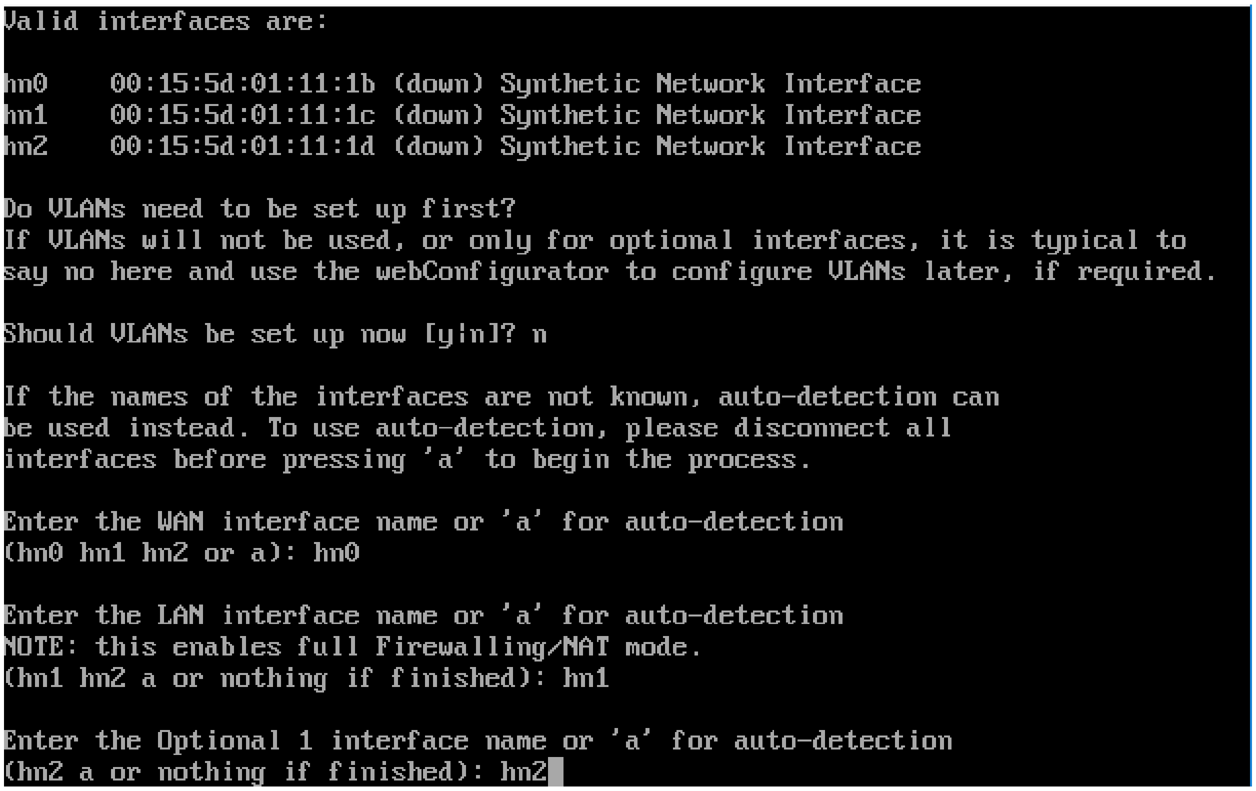
Now, we need to set up networking in the VM itself. In Hyper-V Manager, start the VM if it isn't already running, then connect to the console. Wait for the VM to finish booting. You'll be greeted by the *Assign Interfaces* wizard. The wizard will start by asking if you want to set up VLANS. Say no, because our virtual lab will NOT be using VLANs. Next, the wizard will ask you to define which interface should be assigned to what role. There should be three network interfaces displayed:


pfsense on BAHAMUT_ - Virtual Machine Connection
File Action Media Clipboard View Help
Status: Running

```
Loading configuration......done.

Default interfaces not found -- Running interface assignment option.
hn0: link state changed to UP
hn1: link state changed to UP
hn2: link state changed to UP

Valid interfaces are:

hn0    00:15:5d:01:11:1b (down) Synthetic Network Interface
hn1    00:15:5d:01:11:1c (down) Synthetic Network Interface
hn2    00:15:5d:01:11:1d (down) Synthetic Network Interface

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y|n]? n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(hn0 hn1 hn2 or a):
```

When we performed the Initial VM Settings configuration, I had you record the MAC addresses for each of the network adapters, and document which vswitch they are connected to. - Bridged, Management and IPS 1. Your goal right now is determine which network adapter in pfSense (hn0, hn1, and hn2) corresponds to the network adapter configuration in Hyper-V.

For example, in the illustration above, hn0 with a MAC address of 00:15:5d:01:11:1b has the MAC address of the network adapter connected to the “Bridged Network”. The 1c MAC address of hn1 corresponds to the network adapter attached to the “Management Network” and 1d MAC address of hn2 corresponds to the network adapter attached to “IPS 1”. For our lab, the Bridged Network is our connection to the outside world, and therefore will serve as the “WAN” interface in pfSense. In my case, that means that hn0 is the “WAN” interface. The hn1 interface is connected to the “Management Network” and will serve as the “LAN” interface, and finally, hn2 connected to the “IPS 1 Network” will serve as the “OPT1” interface.

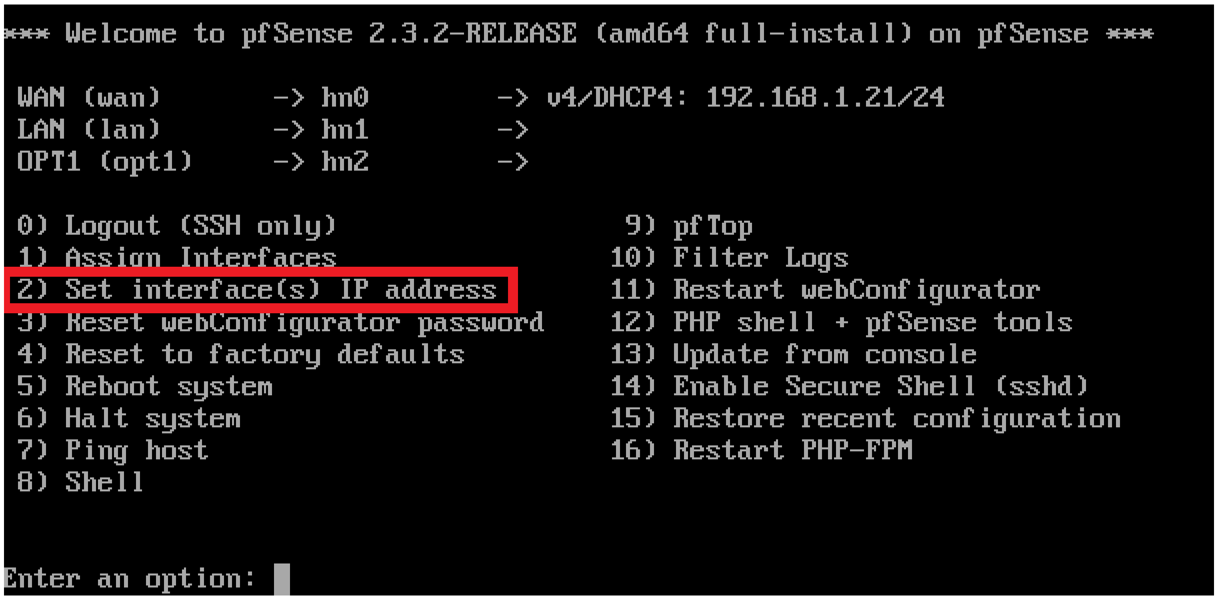


```
Valid interfaces are:

hn0    00:15:5d:01:11:1b (down) Synthetic Network Interface
hn1    00:15:5d:01:11:1c (down) Synthetic Network Interface
hn2    00:15:5d:01:11:1d (down) Synthetic Network Interface

Do VLANs need to be set up first?
If VLANs will not be used, or only for optional interfaces, it is typical to
say no here and use the webConfigurator to configure VLANs later, if required.

Should VLANs be set up now [y|n]? n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(hn0 hn1 hn2 or a): hn0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(hn1 hn2 a or nothing if finished): hn1

Enter the Optional 1 interface name or 'a' for auto-detection
(hn2 a or nothing if finished): hn2
```

The setup script will ask you to confirm your interface choices. If you are satisfied, type “y” to let pfSense configure the interfaces and initial system settings. When everything is done, you’ll be dropped into a menu system.

Note: If for some reason the “Assign Interfaces” wizard was not run automatically when you first booted your pfSense VM, or you made a mistake and you need to correct it, option 1 on the pfSense main menu, *Assign Interfaces* will run the wizard, and/or allow you to correct interface assignment errors as necessary.

Now that we have assigned the network interfaces, we have to configure IP addresses for these interfaces. In most cases, the WAN interface will automatically get an IP address from the device on your physical network that provides DHCP services. If this is not happening, you may have to troubleshoot your physical network. This is beyond the scope of our guide. As for the LAN and OPT1 networks, we have to manually set the IP address, subnet mask, and DHCP scopes for these networks. Select option 2 in the pfSense main menu to get started.



```
*** Welcome to pfSense 2.3.2-RELEASE (amd64 full-install) on pfSense ***

 WAN (wan)       -> hn0        -> v4/DHCP4: 192.168.1.21/24
 LAN (lan)       -> hn1        ->
 OPT1 (opt1)     -> hn2        ->

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell


Enter an option:
```

The configuration wizard starts by asking for the interface the settings should be changed for. We have to go through the process two times, for the LAN and the OPT1 interface individually. Here are the settings I recommend:

LAN IP: **172.16.1.1**
LAN Subnet bit count: **24**
LAN upstream gateway address: **<empty>**
LAN IPv6 address: **<empty>**
Do you want to enable the DHCP server on LAN? **y**
LAN DHCP start address: **172.16.1.10**
LAN DHCP end address: **172.16.1.254**
Do you want to revert to HTTP as the webConfigurator protocol? **N**

OPT1 IP: **172.16.2.1**
OPT1 Subnet bit count: **24**
OPT1 upstream gateway address: **<empty>**
OPT1 IPv6 address: **<empty>**
Do you want to enable the DHCP server on OPT1? **y**

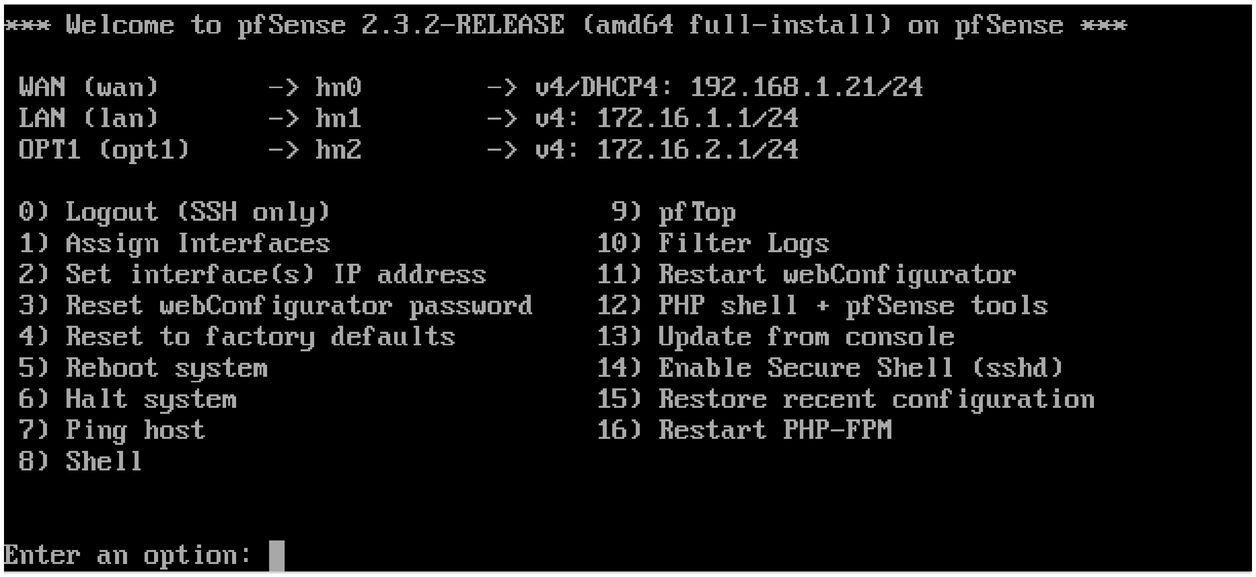
OPT1 DHCP start address: **172.16.2.10**
OPT1 DHCP end address: **172.16.2.254**
Do you want to revert to HTTP as the webConfigurator protocol? **n**

The reason we choose 1.10 and 2.10 as the DHCP start addresses is to reserve a few IP addresses for static DHCP allocations. Static DHCP allocations allow you to configure a DHCP server to always serve the same IP address when requested from a particular MAC address. We'll talk about this a little bit more later. For now, we should be done mucking around in the pfSense CLI. From here on out, we get to use the web UI to configure pfSense.

Note: If you are using 172.16.1.0/24 or the 172.16.2.0/24 network ranges in your physical network, choose another network range to assign to the LAN and OPT1 interfaces to avoid network conflicts.

Note: When you are configuring the LAN and OPT1 interface IP addresses, the configuration script will ask, *Do you want to revert to HTTP as the webConfigurator protocol?* **Always say no to this.** This allows the web UI to accept HTTP logins, and makes your firewall credentials vulnerable to sniffing over the network.

Your pfSense main menu should look something like this when you're all done:



```
*** Welcome to pfSense 2.3.2-RELEASE (amd64 full-install) on pfSense ***

 WAN (wan)       -> hn0        -> v4/DHCP4: 192.168.1.21/24
 LAN (lan)       -> hn1        -> v4: 172.16.1.1/24
 OPT1 (opt1)     -> hn2        -> v4: 172.16.2.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell


Enter an option: 
```

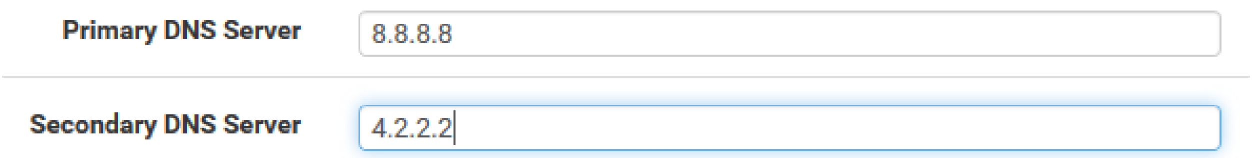
If so, let's move on to setting up pfSense from the webConfigurator. Before doing so however, you must configure the vEthernet adapter attached to the *Management Network* on your Windows host. If you haven't already, you'll want to visit the Unbinding Network Protocols on Windows Virtual Adapters at a minimum. This document will guide you through configuring the vEthernet adapter attached to the *Management Network* with an IP address as well as unbinding network protocols on this adapter to increase the security of your hypervisor host.

You may also want to consider reading Using Windows Firewall to Limit Exposure of Windows Hypervisor Hosts to further enhance the security of the Windows host when interacting with your lab VMs.

webConfigurator - Initial Setup

On the Windows Client Hyper-V host, open your favorite web browser (I prefer Firefox - https://www.mozilla.org/en-US/firefox/new/) and navigate to https://172.16.1.1 (or the IP address you assigned to the LAN interface of your pfSense system). **The default credentials for access to the web interface are admin/pfsense.** If this is your first time logging in, pfSense takes you through a nice little setup wizard. I'll highlight some of the important things to take note of, and/or change as necessary:

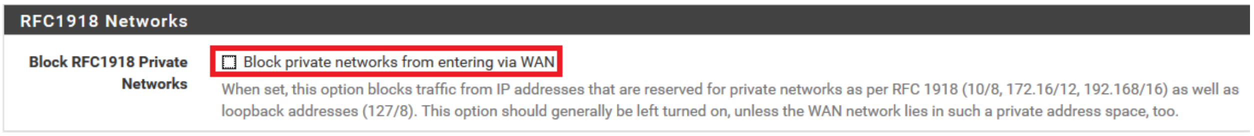
Set the primary and secondary DNS servers you plan on using. I typically use 8.8.8.8 (Google public DNS) and 4.2.2.2 (Level 3 public DNS). If you're using this lab at work, your workplace may have restrictions and you may have to use their DNS servers instead.



Primary DNS Server: 8.8.8.8

Secondary DNS Server: 4.2.2.2

Uncheck the checkbox next to *Block private networks from entering via WAN* rule. Uncheck the checkbox next to *Block Bogon Networks* on this page as well.



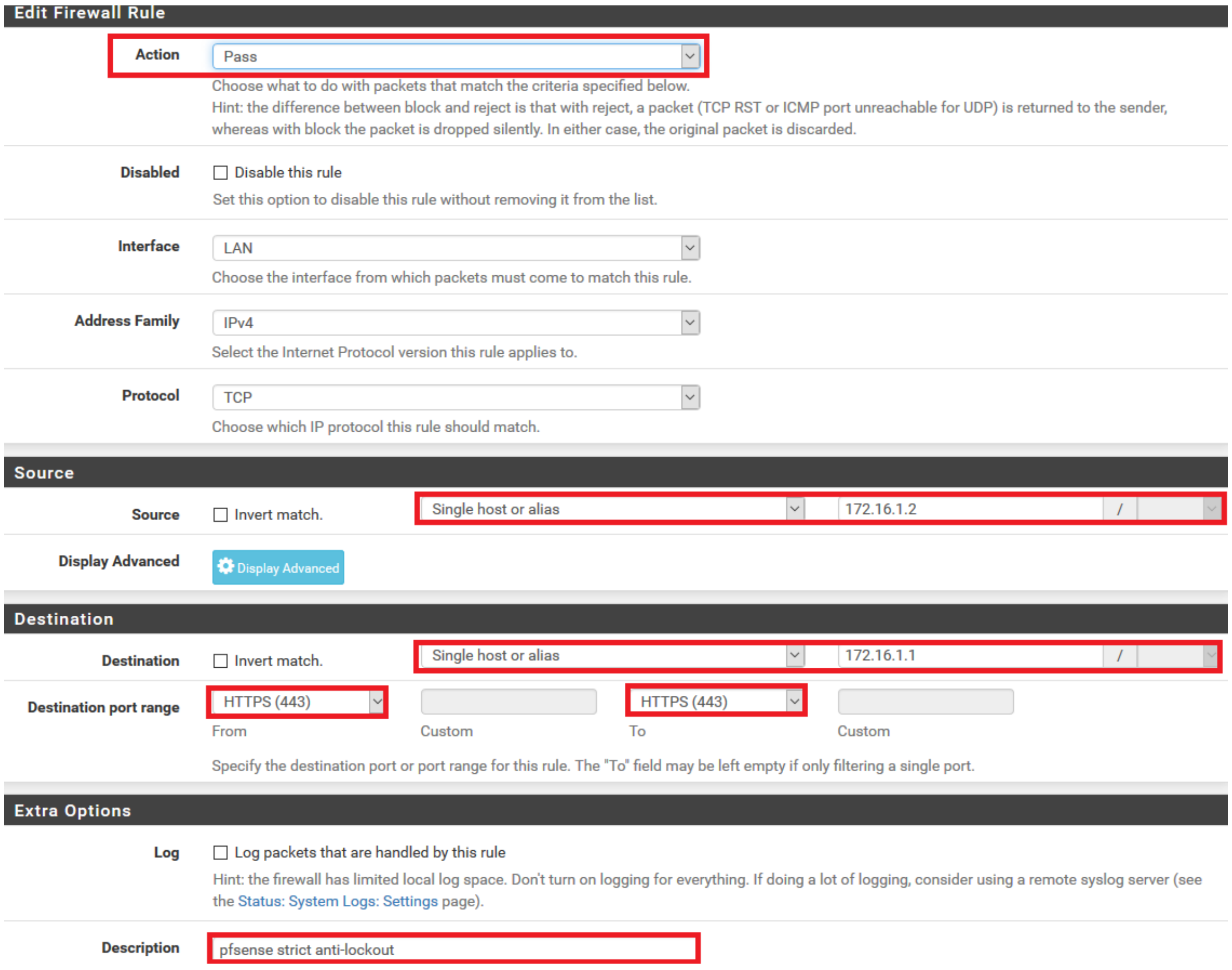
RFC1918 Networks

Block RFC1918 Private Networks: ☐ Block private networks from entering via WAN

When set, this option blocks traffic from IP addresses that are reserved for private networks as per RFC 1918 (10/8, 172.16/12, 192.168/16) as well as loopback addresses (127/8). This option should generally be left turned on, unless the WAN network lies in such a private address space, too.

The other default settings on the first time setup wizard should be fine. Please note that you will be asked to change the password for the admin account as a part of the setup wizard. Document the password, and keep it somewhere safe.

Our next order of business is to restrict access to the web interface of the firewall to the machine with the IP address 172.16.1.2. On the menu bar at the top of the page, select *Firewall > Rules*. Click on *LAN* to modify the firewall policy for the *LAN* interface. Click the *Add* button with the arrow facing up. This will add the firewall rule to the top of the firewall policy to where it will evaluated first. The *Edit Firewall Rule* page is fairly straightforward. I've highlighted the options to be aware of.



Edit Firewall Rule

Action	Pass Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.
Disabled	☐ Disable this rule Set this option to disable this rule without removing it from the list.
Interface	LAN Choose the interface from which packets must come to match this rule.
Address Family	IPv4 Select the Internet Protocol version this rule applies to.
Protocol	TCP Choose which IP protocol this rule should match.

Source

Source	☐ Invert match. Single host or alias 172.16.1.2 /
Display Advanced	Display Advanced

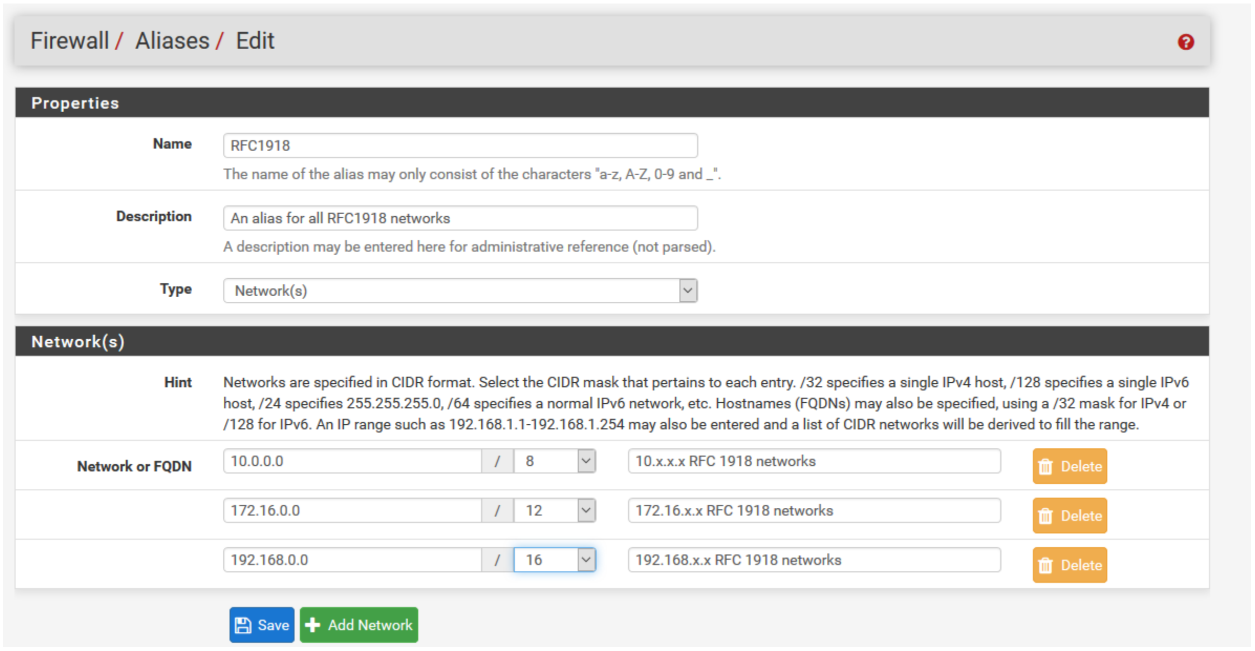
Destination

Destination	☐ Invert match. Single host or alias 172.16.1.1 /
Destination port range	From: HTTPS (443) Custom: To: HTTPS (443) Custom: Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.

Extra Options

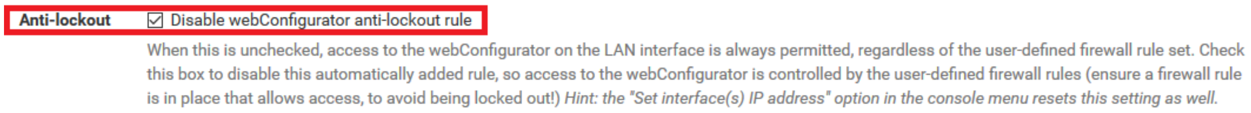
Log	☐ Log packets that are handled by this rule Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the Status: System Logs: Settings page).
Description	pfsense strict anti-lockout

When you are done, click the *Save* icon at the bottom of the page. This will take you back to the previous page. A yellow dialogue box with a green button called *Apply Changes* will appear. Click this button to apply this new firewall rule. Next, we want to an alias. Navigate to *Firewall > Aliases*. On the Firewall Aliases page, click on *IP*, then click *Add*. Create an alias with the following settings:



Name	RFC1918
	The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".
Description	An alias for all RFC1918 networks
	A description may be entered here for administrative reference (not parsed).
Type	Network(s)
Hint	Networks are specified in CIDR format. Select the CIDR mask that pertains to each entry. /32 specifies a single IPv4 host, /128 specifies a single IPv6 host, /24 specifies 255.255.255.0, /64 specifies a normal IPv6 network, etc. Hostnames (FQDNs) may also be specified, using a /32 mask for IPv4 or /128 for IPv6. An IP range such as 192.168.1.1-192.168.1.254 may also be entered and a list of CIDR networks will be derived to fill the range.
Network or FQDN	10.0.0.0 / 8 — 10.x.x.x RFC 1918 networks
	172.16.0.0 / 12 — 172.16.x.x RFC 1918 networks
	192.168.0.0 / 16 — 192.168.x.x RFC 1918 networks

Click Save to be brought back to the previous page, then click *Apply Changes*. We just created an alias for RFC1918 networks (local networks that are not routable through the public internet) This will come in handy later for creating firewall rules around these networks. Our final action for now is to navigate to *System > Advanced*, and click to fill in the checkbox next to the option *Disable webConfigurator anti-lockout rule*, and click *Save* on the bottom of the page. **Be aware that if you did not create the firewall rule above, this <u>will</u> lock you out of the webConfigurator as soon as you click *Save*.**



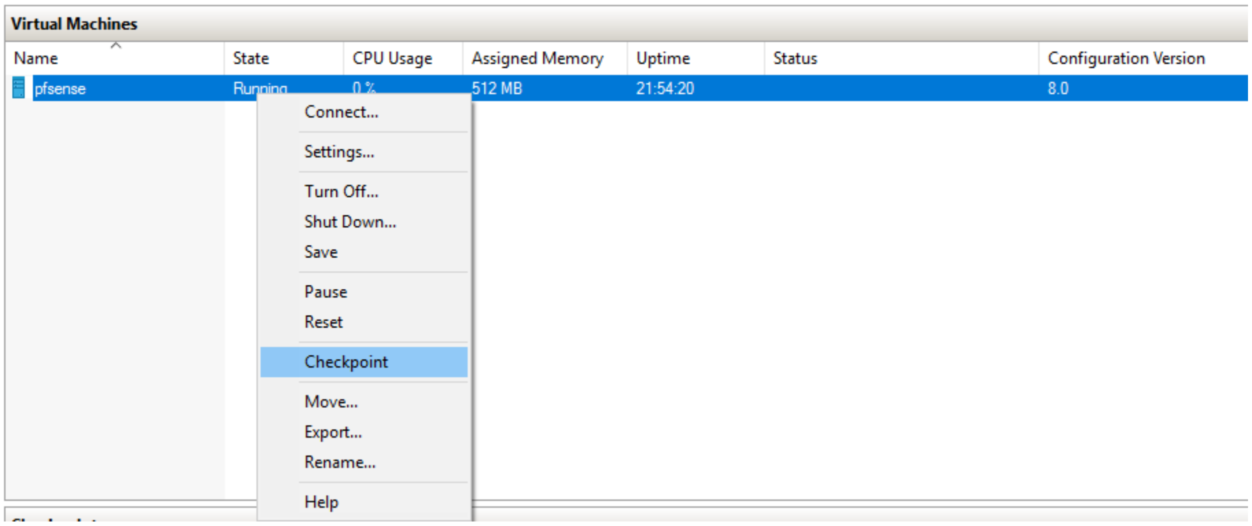
Anti-lockout ☑ Disable webConfigurator anti-lockout rule

When this is unchecked, access to the webConfigurator on the LAN interface is always permitted, regardless of the user-defined firewall rule set. Check this box to disable this automatically added rule, so access to the webConfigurator is controlled by the user-defined firewall rules (ensure a firewall rule is in place that allows access, to avoid being locked out!) *Hint: the "Set interface(s) IP address" option in the console menu resets this setting as well.*

Making Checkpoints

Snapshot management is an important part of any virtual lab. In Hyper-V terminology, snapshots are known as checkpoints. Our next step will be to create a checkpoint for the pfSense VM, so that if we make a mistake in the configuration, you have a known good, somewhat functional baseline to fall back to.

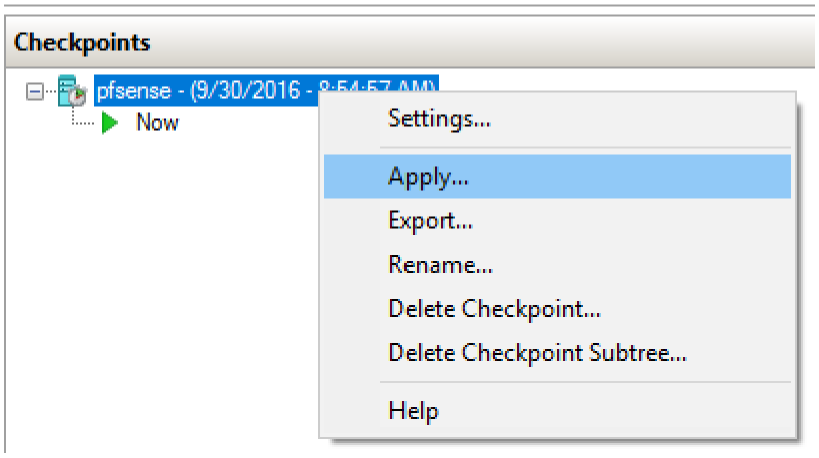
I’m going to show you how to create a snapshot for your pfSense VM. Before doing so, ensure that in the VM’s settings that checkpoints are enabled, and that if you are running Client Hyper-V on Windows 10, that *Standard* checkpoints have been selected. In the Hyper-V Manager, under the Virtual Machines pane, right click on pfSense, and choose the Checkpoint option.



Virtual Machines

Name	State	CPU Usage	Assigned Memory	Uptime	Status	Configuration Version
pfsense	Running	0 %	512 MB	21:54:20		8.0

After a moment or two, you’ll notice under the *Checkpoints* pane you now have a new checkpoint available. Underneath that check point you should see a green triangle and the word *Now*. If at any point you needed to revert back to this checkpoint, simply right click on the checkpoint you just made, and select *Apply* to restore the checkpoint.



pfSense Summary

At the end of all this, you should have a pfSense VM with the following settings:

- 512MB of RAM
- 5GB of disk
- 1 Virtual CPU
- 3 Network interfaces:
    - 1 Bridged Network (WAN interface)
    - 1 Management Network (LAN interface)
    - 1 IPS 1 Network (OPT1 interface)

pfSense should be configured as followed:

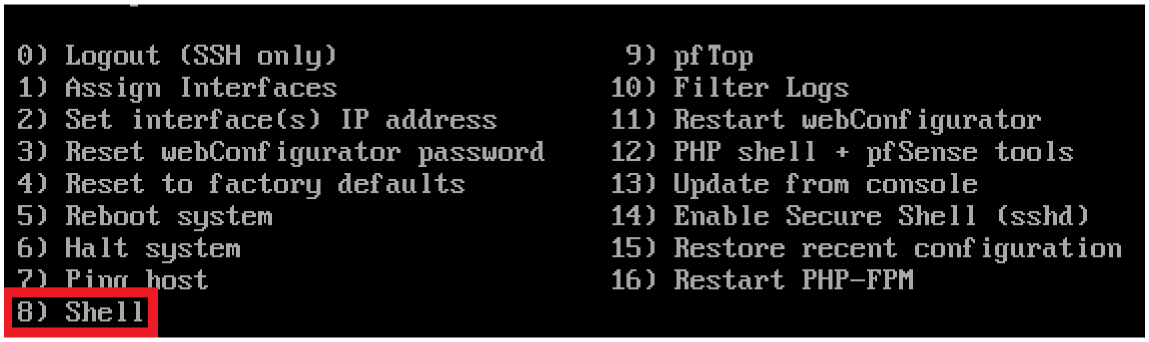
- The WAN interface should be the Hyper-V network adapter connected to the Bridged Network. This interface should automatically get an IP address via DHCP. If this isn't happening, get some help with whoever administers your local network, or start troubleshooting. **Alternatively, configure a static IP address**
- The LAN interface should have an IP address of 172.16.1.1.
- The LAN network should be 172.16.1.0 with a /24 (255.255.255.0) subnet mask
- DHCP should be configured for this network with a scope of 172.16.1.10-172.16.1.254
    - 172.16.1.3-172.16.1.9 are available for setting static DHCP mappings
- The OPT1 interface should have an IP address of 172.16.2.1
- The OPT1 network should be 172.16.2.0 with a /24 (255.255.255.0) subnet mask
- DHCP should be configured for this network with a scope of 172.16.2.10-172.16.2.254
    - 172.16.2.2-172.16.2.9 are available for static DHCP mappings
- 172.16.1.2 should have a firewall rule configured to allow it access to the firewall over HTTPS; this is the anti-lockout firewall rule
- You should have an alias for RFC1918 networks configured for use with the firewall policy.
- You should have at least one good checkpoint you can revert to if you need to redo a step or somehow got lost/confused

What's Next?

Now that you have an operational pfSense VM, your next steps are to read the pfSense Firewall Rules and Network Services Guide. Make sure the firewall policies match those illustrated in the segmentation guide, and make sure that the pfSense VM is hosting all the services mentioned in the Core Network Services guide to make your firewall fully functional and ready to handle your lab network.

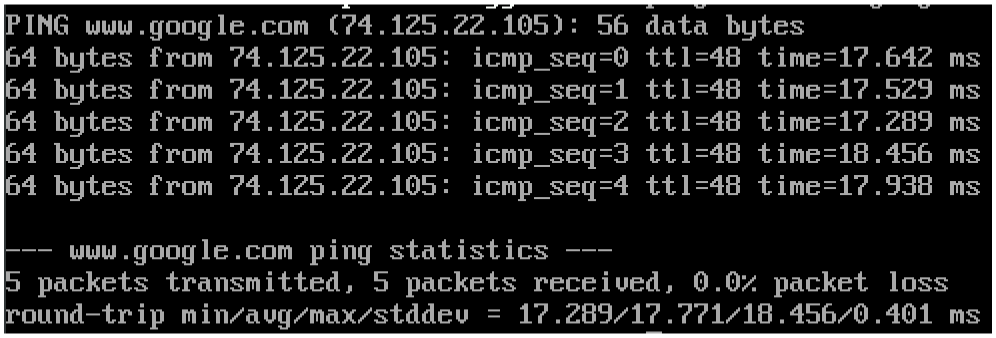
Final Connectivity Checks and Troubleshooting

Once you have finished setting up your pfSense VM according to the instructions laid out, Open a console session to the pfSense VM, and attach your mouse and keyboard by clicking on the console. If the screen is blank, hit the *Enter* key to bring up the pfSense main menu. Select option *8) Shell* to start a shell session on pfSense.



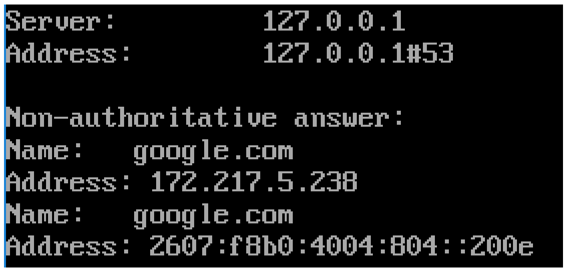
```
0) Logout (SSH only)                   9) pfTop
1) Assign Interfaces                  10) Filter Logs
2) Set interface(s) IP address        11) Restart webConfigurator
3) Reset webConfigurator password     12) PHP shell + pfSense tools
4) Reset to factory defaults          13) Update from console
5) Reboot system                      14) Enable Secure Shell (sshd)
6) Halt system                        15) Restore recent configuration
7) Ping host                          16) Restart PHP-FPM
8) Shell
```

The first, run the command `ping -c 5 www.google.com`



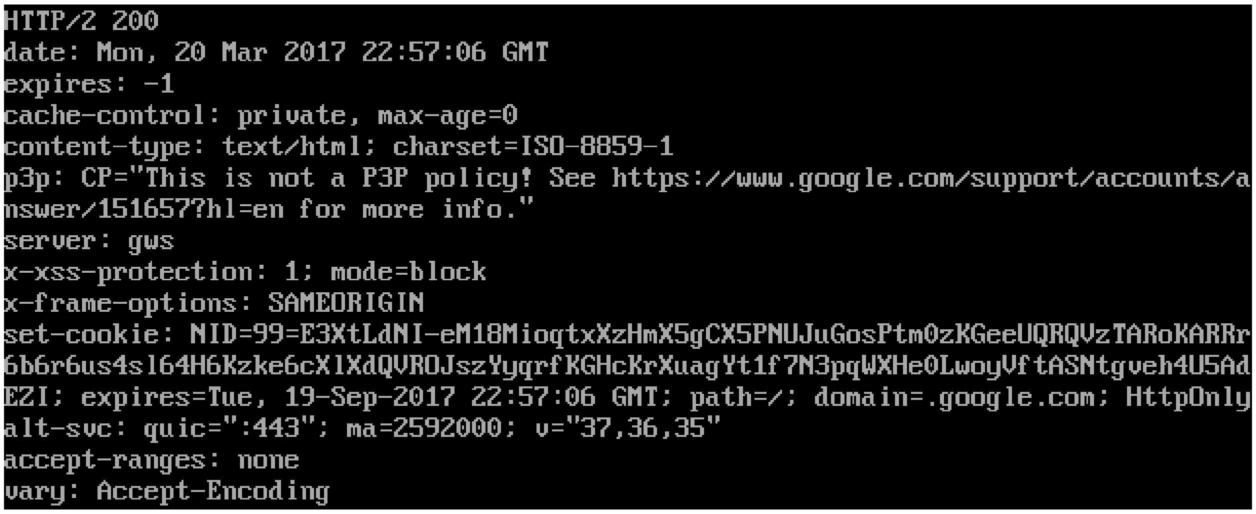
```
PING www.google.com (74.125.22.105): 56 data bytes
64 bytes from 74.125.22.105: icmp_seq=0 ttl=48 time=17.642 ms
64 bytes from 74.125.22.105: icmp_seq=1 ttl=48 time=17.529 ms
64 bytes from 74.125.22.105: icmp_seq=2 ttl=48 time=17.289 ms
64 bytes from 74.125.22.105: icmp_seq=3 ttl=48 time=18.456 ms
64 bytes from 74.125.22.105: icmp_seq=4 ttl=48 time=17.938 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 17.289/17.771/18.456/0.401 ms
```

The output from the `ping` command should look similar to the output above. Next, run the command `nslookup google.com`



```
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.5.238
Name:   google.com
Address: 2607:f8b0:4004:804::200e
```

Again, you should have output similar to the illustration above. Finally, run `curl -I https://www.google.com`



```
HTTP/2 200
date: Mon, 20 Mar 2017 22:57:06 GMT
expires: -1
cache-control: private, max-age=0
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See https://www.google.com/support/accounts/a
nswer/151657?hl=en for more info."
server: gws
x-xss-protection: 1; mode=block
x-frame-options: SAMEORIGIN
set-cookie: NID=99=E3XtLdNI-eM18MioqtxXzHmX5gCX5PNUJuGosPtm0zKGeeUQRQVzTARoKARRr
6b6r6us4sl64H6Kzke6cXlXdQVROJszYyqrfKGHcKrXuagYt1f7N3pqWXHe0LwoyVftASNtgveh4U5Ad
EZI; expires=Tue, 19-Sep-2017 22:57:06 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: quic=":443"; ma=2592000; v="37,36,35"
accept-ranges: none
vary: Accept-Encoding
```

The output of your `curl` command should be similar to the screen capture above. If so, you may type `exit` to leave the shell, and close your connection to the pfSense console. If you had any problems with any of the commands above, you've got troubleshooting to do. When it comes to troubleshooting, start at layer 1 of the OSI model (Physical) and work your way up. Here are some questions to help the troubleshooting process:

- Which of the commands above failed? Did they all fail?
- Have you checked your physical cabling or wireless network connectivity?
- Does your company/LAN have MAC filtering/port security on? You might need to modify switch settings for your drop, or talk to your network administrator.
- Have you checked your upstream gateway/SOHO router? Trying pinging the gateway's IP address to see if your pfSense VM can reach the gateway. Try running the `traceroute` command to google.com to see if and where your ping command is failing.

- Do you have an upstream firewall in place? Is 443/TCP allowed outbound? Is 53/UDP allowed outbound? Is ICMP allowed?
- Did you input the commands correctly? Try entering them again. No, I’m not patronizing you.

Once you have verified that your firewall’s connectivity, policies, and services are properly configured, **SNAPSHOT THE VM <u>BEFORE</u>** moving on and trying your hand at setting up the remaining VMs.

Your Turn

The pfSense virtual machine required a lot of custom configuration, both in Hyper-V, as well as in the VM itself once the OS had been installed. Now that we have done that together, I'm going to have you create the rest of the virtual machines on your own -- with the exception of the Metasploitable 2 VM, since it's a special case for reasons you will see momentarily. Below are a set of spec sheets to help you create the remaining virtual machines for the lab on your own. Think of them as checklists you should be able to run through unaccompanied.

All of our Linux-based virtual machines are built on Debian Linux based distros (both, Ubuntu and Kali Linux are derived from Debian), so setup and configuration of all of our lab VMs should be very similar.

Kali Linux VM

Kali Linux is a popular penetration testing distro. Popular because hackers and script kiddies are lazy, and practically everything you need for performing a penetration test or joining anonymous is installed by default. Kali is going to be our loud, obnoxious attacker that we're going to use as a noise generator for the express purpose of generating events on our IPS VM, and to test the AFPACKET bridge's functionality. Perform the following tasks to create a Kali Linux VM:

- Download Kali Linux 64-bit here: https://www.kali.org/downloads/
- Create a VM with the following settings:
    - Name the VM "Kali"
    - Make the VM *Generation 1*
    - Allocate 4GB of RAM
        - Uncheck the *Use Dynamic Memory for this virtual machine* feature
    - Connect the VM's network adapter to the *IPS 1* virtual switch
    - Allocate 80 GB of space for the vhdx file
    - Under *Installation Options*, select the *Install from a bootable CD/DVD-ROM* radio button, then *select the Image file (.iso)* option, and browse to the Kali Linux ISO you downloaded
- Once the VM is created, adjust the following settings:
    - 1-2 virtual CPUs (virtual CPUs can be adjusted under *Settings > Hardware > Processor* and adjust *Number of Virtual Processors:* to 2)
    - Remove the SCSI Controller
    - If you are running Windows 10, change the *Checkpoints* option (Under *Settings > Management > Checkpoints*) to *Standard*
- Install Kali Linux

    - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.2.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy.
- After the installation is completed, shut down, or turn off the VM and adjust the following settings:
    - Remove the DVD drive located under IDE Controller 1
    - Under *Network Adapter* > *Advanced Features* in the MAC address section, **document the MAC address of this network adapter.**
- Log in to the pfSense web UI, navigate to *Services* > *DHCP Server*, and add a static mapping to the OPT1 interface DHCP, for the MAC address of adapter 1 for the Kali Linux VM; assign it the IP address 172.16.2.2, making sure to enter a description that tells you which VM this static mapping belongs to!
- Power on the virtual machine and do the following:
    - Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense
        - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
    - Verify the virtual machine can reach the internet
        - In a terminal/shell run the command `curl -I https://www.google.com`
            - This will confirm DNS can resolve domain names, and that the VM can reach the internet
            - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense
                - **Note:** The `ping` command, in this case, will work from this virtual machine, due to the unique firewall policy on the OPT1 network.
    - Update/patch the VM
        - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
            - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
    - Create a snapshot for the VM

SIEM VM

SIEM is shorthand for Security Intrusion Events Manager. This is fancy security nomenclature for "log aggregator"; we will be having our IPS VM log its events here. We're going to be running

Splunk on this VM. Splunk is a commercial program for managing logs on a pretty large scale. By default, and with no licensing, Splunk only allows you to collect or "index" 500MB worth of logs per day, however, there are /ways/ around this that we'll talk about later. Perform the following tasks to set up the SIEM VM:

- Download the latest Ubuntu Server 64-bit LTS release. In our case, this is Ubuntu Server 16.04.1, which can be found here: http://www.ubuntu.com/download/server
- Create a VM with the following settings:
  - Name the VM "SIEM"
  - Make the VM *Generation 1*
  - Allocate 4GB of RAM
    - Uncheck the *Use Dynamic Memory for this virtual machine* feature
  - Connect the VM's network adapter to the *Management Network* virtual switch
  - Allocate 80 GB of space for the vhdx file
  - Under *Installation Options*, select the *Install from a bootable CD/DVD-RO*M radio button, then select the *Image file (.iso)* option, and browse to the Ubuntu Server ISO you downloaded
- Once the VM is created, adjust the following settings:
  - 1-2 virtual CPUs (virtual CPUs can be adjusted under *Settings > Hardware > Processor* and adjust *Number of Virtual Processors:* to 2)
  - Remove the SCSI Controller
  - If you are running Windows 10, change the *Checkpoints* option (Under *Settings > Management > Checkpoints*) to *Standard*
- Install Ubuntu Server
  - Be sure to install the *Standard System Utilities* and *OpenSSH Server* role when asked. You should not need to install any other roles or features for this server.
  - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.1.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy.
- After the installation is completed, shut down, or turn off the VM and adjust the following settings:
  - Remove the DVD drive located under IDE Controller 1
  - Under *Network Adapter > Advanced Features* in the MAC address section, **document the MAC address for this network adapter.**
- Log in to the pfSense web UI, navigate to *Services > DHCP Server*, and add a static mapping to the LAN interface DHCP, for the MAC address of adapter 1 for the SIEM VM; assign it the IP address 172.16.1.3, making sure to enter a description that tells you which VM this static mapping belongs to!
- Power on the virtual machine and do the following:
  - Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever

the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense

    - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
  - Verify the virtual machine can reach the internet
    - In a terminal/shell run the command `curl -I https://www.google.com`
      - This will confirm DNS can resolve domain names, and that the VM can reach the internet
      - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense
        - **Note:** The `ping` command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail
  - Update/patch the VM
    - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
      - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
  - Create a snapshot for the VM

IPS VM

This VM is going to be responsible for running our IPS Software, as well as the AFPACKET bridge between the IPS 1 and IPS 2 virtual switches. Perform the following tasks to install the IPS VM:

- If you installed the SIEM VM already, you should already have Ubuntu Server downloaded. If not, download the latest Ubuntu Server 64-bit LTS release. In our case, this is Ubuntu Server 16.04.1, which can be found here: http://www.ubuntu.com/download/server
- Create a VM with the following settings:
  - Name the VM "IPS"
  - Make the VM *Generation 1*
  - Allocate 2GB of RAM
    - Uncheck the *Use Dynamic Memory for this virtual machine* feature
  - Connect the VM's first network adapter to the *Management Network* virtual switch
  - Allocate 80 GB of space for the vhdx file
  - Under *Installation Options*, select the *Install from a bootable CD/DVD-ROM* radio button, then select the *Image file (.iso)* option, and browse to the Ubuntu Server ISO you downloaded

- Once the VM is created, adjust the following settings:
  - 1-2 virtual CPUs (virtual CPUs can be adjusted under *Settings* > *Hardware* > *Processor* and adjust *Number of Virtual Processors:* to 2)
  - Remove the SCSI Controller
  - Under *Add New Hardware*, We need to add two more Network Adapters. For now, ***<u>DO NOT</u>*** connect either of these network adapters to a virtual switch; leave them as “Not connected” for right now
    - The only interface that should be connected to a virtual switch right now should be the first network adapter that was connected to the *Management Network*
  - If you are running Windows 10, change the *Checkpoints* option (Under *Settings* > *Management* > *Checkpoints*) to *Standard*
- Install Ubuntu Server
  - The installer will reach a section titled *Configure the network* and ask you which network card is the primary network interface for this system
    - Usually, these interfaces are labeled `eth0`, `eth1`, and `eth2`, but may be labeled differently in your case.
    - In almost all cases, the first network adapter added to a VM in Client Hyper-V will line up with the first network interface listed on the *Configure the network* screen (which is almost always `eth0`). Hit Enter, and wait for it to try and configure the interface via DHCP.
    - The installer will try to get an IP address for the network interface via DHCP. If this is not successful, select the `<Go Back>` option, and select one of the other two network interfaces, and try again.
    - If none of the interfaces work, verify the pfSense VM is running, connected to the *Management Network*, and that the DHCP service is enabled, then try again until DHCP configuration is successful.
  - Be sure to install the *Standard System Utilities* and *OpenSSH Server* role when asked. You should not need to install any other roles or features for this server.
  - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be asked if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.1.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy.
- After the install is completed, shut down, or turn off the VM and adjust the following settings:
  - Remove the DVD drive located under IDE Controller 1
  - Select the second and third network adapters, and connect them to the *IPS 1* and *IPS 2* networks respectively.
  - Under *Network Adapter* > *Advanced Features* in the MAC address section, **document the MAC address for all three network adapters on this VM.**
- Log in to the pfSense web UI, navigate to *Services* > *DHCP Server*, and add a static

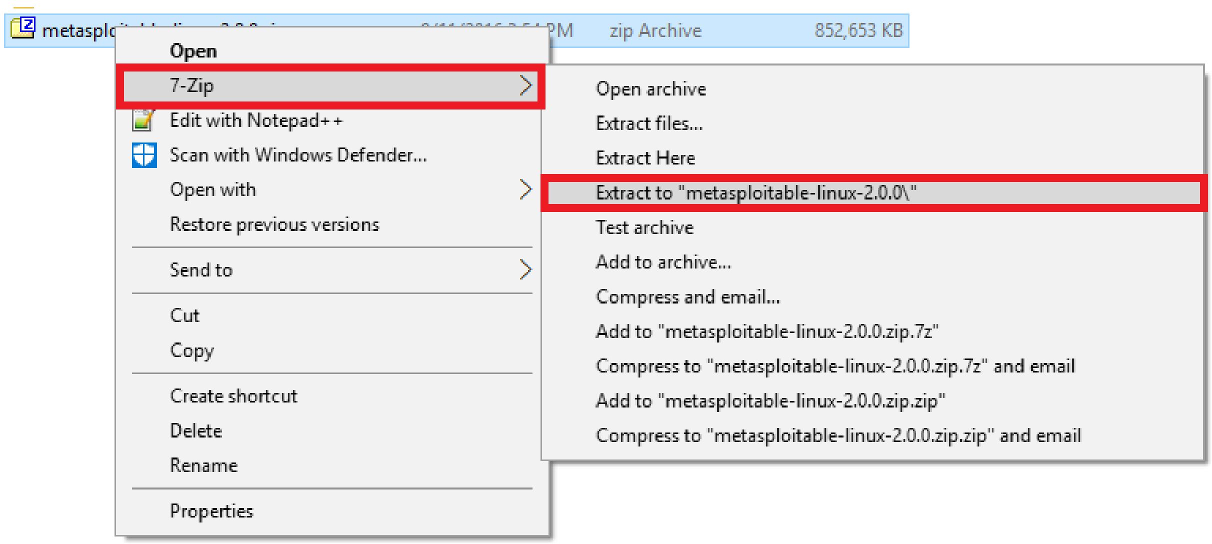
mapping to the LAN interface DHCP, for the MAC address of the network adapter attached to the *Management Network* virtual switch; assign it the IP address 172.16.1.4, making sure to enter a description that tells you which VM this static mapping belongs to!

- Power on the virtual machine and do the following:
  - Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run `ifconfig -a | less` and check the MAC addresses of all three interfaces to verify which interface is connected to the *Management Network*
  - Verify that the VM has an IP address on the *Management Network* (172.16.1.4) and *only* on that network.
    - If the virtual machine does not have an IP address, run the `dhclient -i [interface connected to the management network]` to have the VM request an IP address from the DHCP server.
    - If for some reason the VM has an IP address from the 172.16.2.0/24 network, you can run `ifdown [interface name];ifconfig [interface name] up` to fix this.
  - Verify the virtual machine can reach the internet
    - In a terminal/shell run the command `curl -I https://www.google.com`
      - This will confirm DNS can resolve domain names, and that the VM can reach the internet
      - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense
        - **Note:** The `ping` command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail
  - Update/patch the VM
    - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
      - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
  - Create a snapshot for the VM

Metasploitable 2

Setting up Metasploitable 2 on Hyper-V is slightly more complex than the just creating a virtual machine. In order to install Metasploitable 2, you'll need a special powershell script in order to convert the VMware VMDK disk file that it is distributed as into a VHD file that Hyper-V can make sense of.

First and foremost, go and download Metasploitable 2 from https://sourceforge.net/projects/metasploitable/files/Metasploitable2/metasploitable-linux-2.0.0.zip/download. Metasploitable 2 is distributed as a zip archive, so you need a compression utility to extract it. If you followed the instructions for decompressing the pfSense ISO earlier, you should have 7-Zip installed. Once downloaded, I moved the zip file to `D:\VMs`, where the rest of my Hyper-V VMs are stored. We need to extract the zip archive to get access to the VMDK file. Right click on the zip file and select *7-Zip* > *Extract to `metasploitable-linux-2.0.0\`*.



zip Archive
852,653 KB
Open
7-Zip
Edit with Notepad++
Scan with Windows Defender...
Open with
Restore previous versions
Send to
Cut
Copy
Create shortcut
Delete
Rename
Properties
Open archive
Extract files...
Extract Here
Extract to "metasploitable-linux-2.0.0\"
Test archive
Add to archive...
Compress and email...
Add to "metasploitable-linux-2.0.0.zip.7z"
Compress to "metasploitable-linux-2.0.0.zip.7z" and email
Add to "metasploitable-linux-2.0.0.zip.zip"
Compress to "metasploitable-linux-2.0.0.zip.zip" and email

Next, open your favorite web browser, and visit https://www.microsoft.com/en-us/download/details.aspx?id=42497. Since Microsoft has a habit of occasionally reorganizing their website, and breaking links, You want to look for the *Microsoft Virtual Machine Converter* (Specifically, I'm using version 3.0, but as always, use the latest version available). Download the msi file and install it.

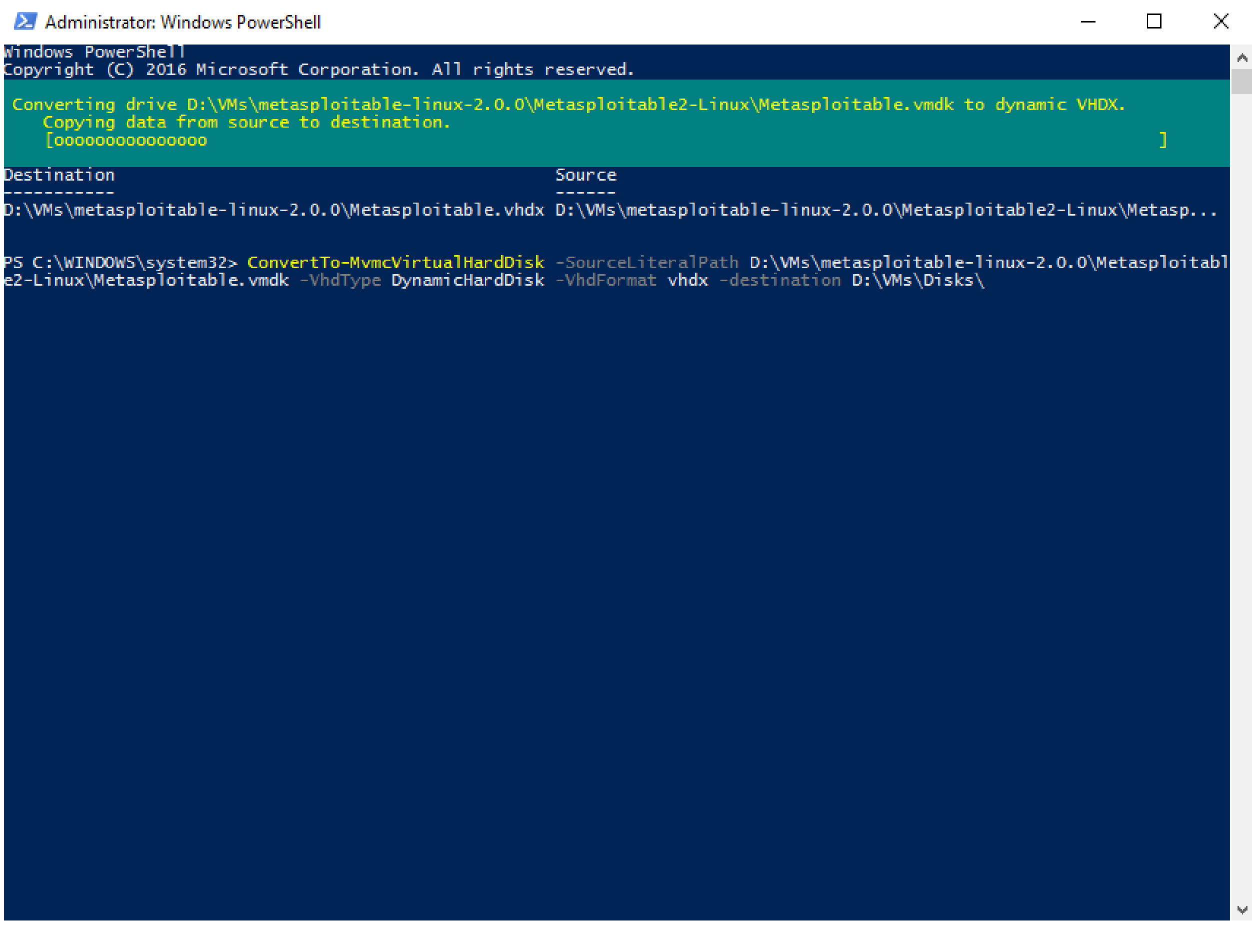
Please note, that depending on your system configuration, you may need to open a powershell.exe shell as administrator and run `Set-ExecutionPolicy Bypass` in order to run the powershell module this application relies on. For more information, please review this MS Technet article: https://technet.microsoft.com/en-us/library/hh849812.aspx.

One everything is installed, open an elevated powershell prompt (right click on the icon and select *Run as Administrator*) and run the following command: `Import-Module ‘C:\Program Files\Microsoft Virtual Machine Converter\MvmcCmdlet.psd1’`

If everything ran correctly you should get a new prompt back with no errors. The following command will extract the VMDK file from `D:\VMs\metasploitable-linux-2.0.0\Metasploitable2-Linux\Metasploitable.vmdk`, and dump the converted VHDX file to `D:\VMs\Disks`

```
ConvertTo-MvmcVirtualHardDisk -SourceLiteralPath D:\VMs\metasploitable-linux-
2.0.0\Metasploitable2-Linux\Metasploitable.vmdk -VhdType DynamicHardDisk -
VhdFormat vhdx -destination D:\VMs\Disks\
```

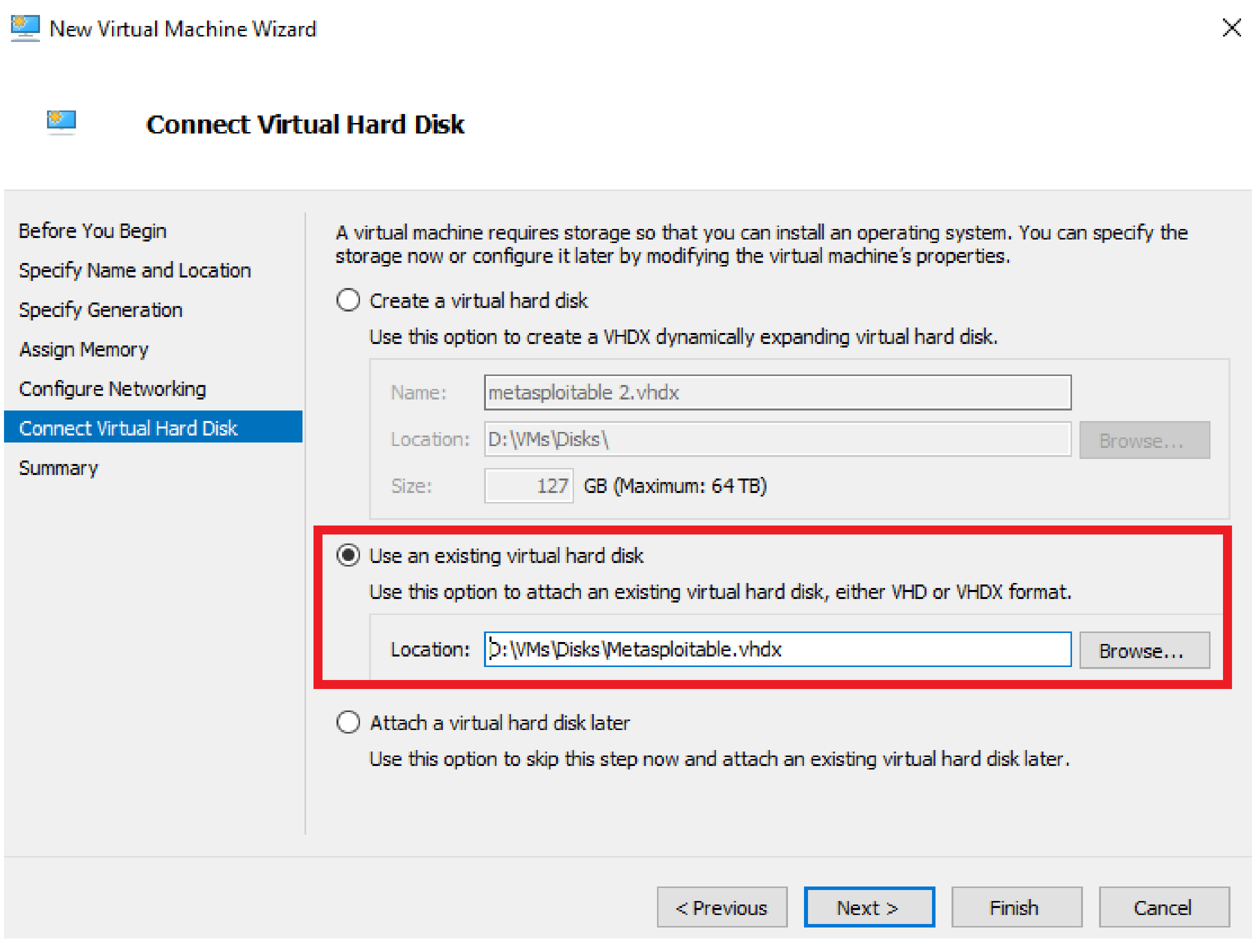
Adjust the drive letters and directory paths as necessary for your Windows host. If everything was executed properly, you should be greeted with this in the powershell prompt, as it converts the disk format into something Hyper-V can use:



```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

 Converting drive D:\VMs\metasploitable-linux-2.0.0\Metasploitable2-Linux\Metasploitable.vmdk to dynamic VHDX.
    Copying data from source to destination.
    [oooooooooooooooo                                                                                       ]

Destination                                          Source
-----------                                          ------
D:\VMs\metasploitable-linux-2.0.0\Metasploitable.vhdx D:\VMs\metasploitable-linux-2.0.0\Metasploitable2-Linux\Metasp...


PS C:\WINDOWS\system32> ConvertTo-MvmcVirtualHardDisk -SourceLiteralPath D:\VMs\metasploitable-linux-2.0.0\Metasploitabl
e2-Linux\Metasploitable.vmdk -VhdType DynamicHardDisk -VhdFormat vhdx -destination D:\VMs\Disks\
```

Now that we have the VHDX, we need to create the VM to run it. Open Hyper-V Manager and create a new VM, just like with the previous VMs. Give it the following settings:

- Name: Metasploitable 2
- Generation: 1
- RAM: 512MB
- Uncheck *Use Dynamic Memory for this virtual machine*
- Leave the *Network Adapter* in its default state (Not Connected)
- On the *Connect Virtual Hard Disk* screen, click the *Use an existing virtual hard disk* radio button, then browse to `D:\VMs\Disks\Metasploitable.vhdx` (or the location your Metasploitable vhdx is located)


New Virtual Machine Wizard

Connect Virtual Hard Disk

Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Summary

A virtual machine requires storage so that you can install an operating system. You can specify the storage now or configure it later by modifying the virtual machine's properties.

○ Create a virtual hard disk

Use this option to create a VHDX dynamically expanding virtual hard disk.

Name: metasploitable 2.vhdx

Location: D:\VMs\Disks\

Size: 127 GB (Maximum: 64 TB)

● Use an existing virtual hard disk

Use this option to attach an existing virtual hard disk, either VHD or VHDX format.

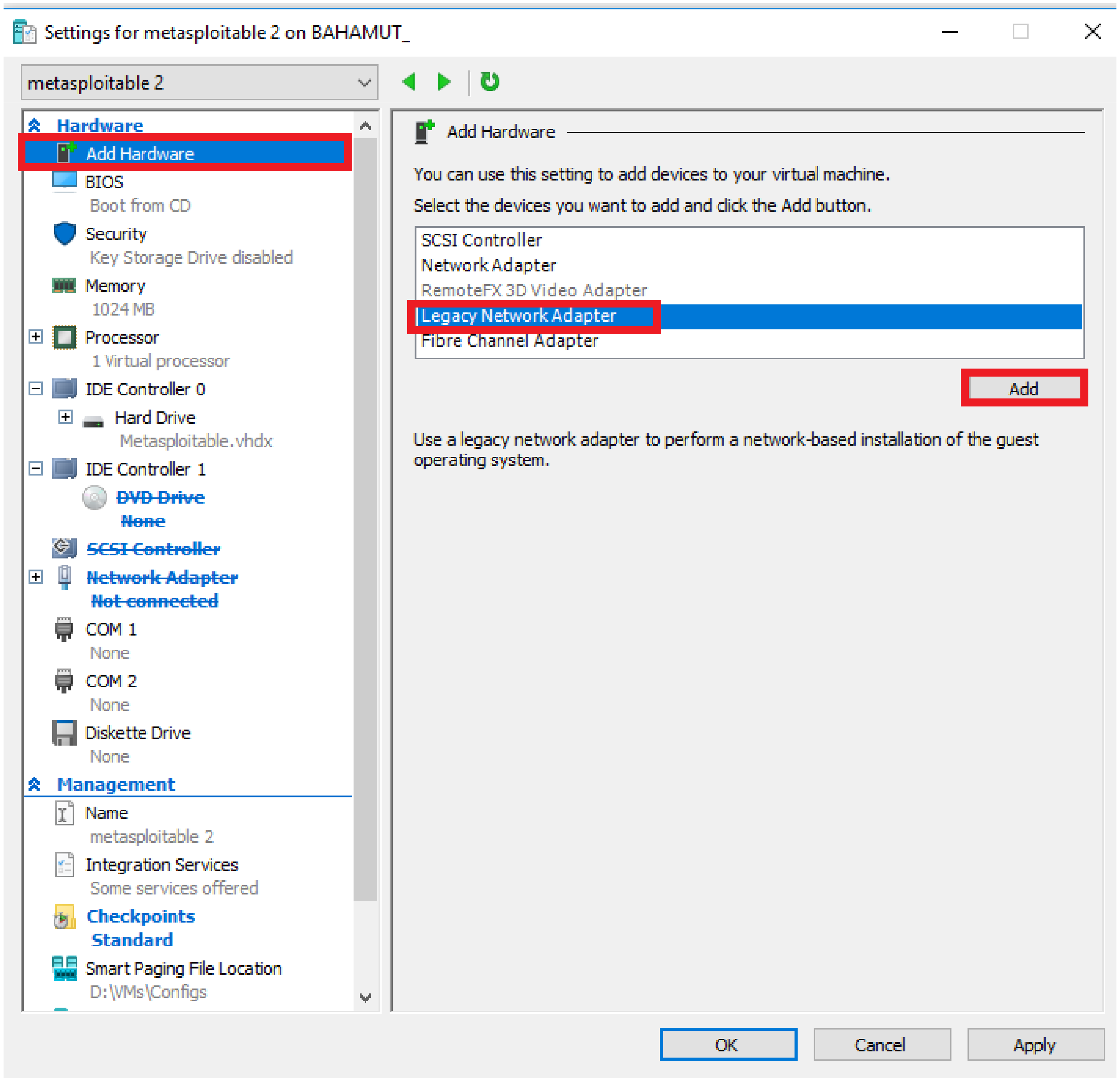
Location: D:\VMs\Disks\Metasploitable.vhdx

○ Attach a virtual hard disk later

Use this option to skip this step now and attach an existing virtual hard disk later.

The installer will skip the installation options screen, so click *Finish* to create the VM. In the Hyper-V Manager, open the settings for the "Metasploitable 2" VM and make the following changes:

- Remove the SCSI Controller
- Remove the DVD Drive
- Remove the Network Adapter
- Reconfigure *Checkpoints* from *Production* to *Standard*
- Choose *Add New Hardware* add the *Legacy Network Adapter*



Settings for metasploitable 2 on BAHAMUT_

metasploitable 2

Hardware
- Add Hardware
- BIOS
  Boot from CD
- Security
  Key Storage Drive disabled
- Memory
  1024 MB
- Processor
  1 Virtual processor
- IDE Controller 0
  - Hard Drive
    Metasploitable.vhdx
- IDE Controller 1
  - ~~DVD Drive~~
    ~~None~~
- ~~SCSI Controller~~
- ~~Network Adapter~~
  ~~Not connected~~
- COM 1
  None
- COM 2
  None
- Diskette Drive
  None

Management
- Name
  metasploitable 2
- Integration Services
  Some services offered
- Checkpoints
  Standard
- Smart Paging File Location
  D:\VMs\Configs

Add Hardware

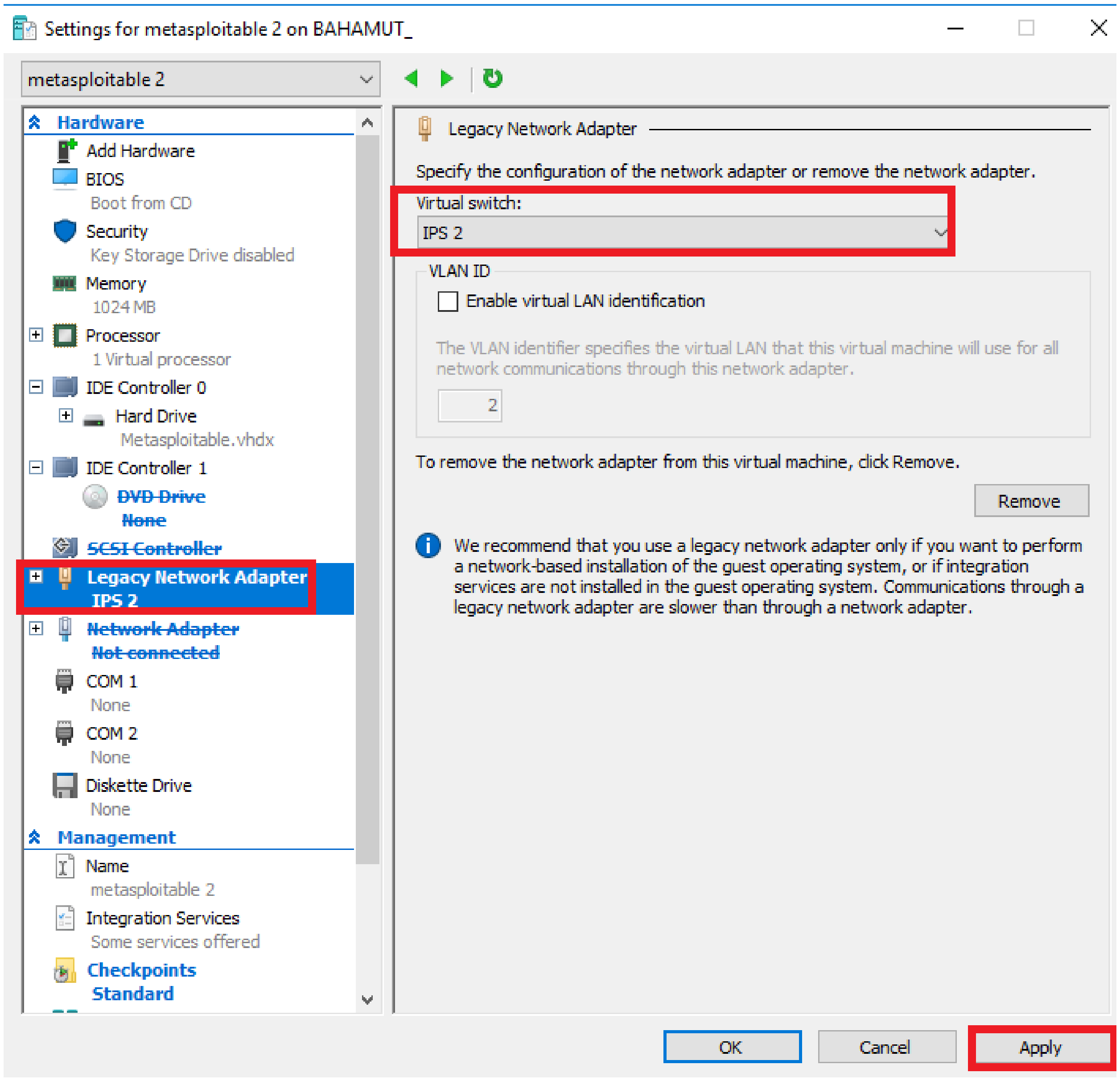
You can use this setting to add devices to your virtual machine.

Select the devices you want to add and click the Add button.

SCSI Controller
Network Adapter
RemoteFX 3D Video Adapter
Legacy Network Adapter
Fibre Channel Adapter

Use a legacy network adapter to perform a network-based installation of the guest operating system.

Click on the *Legacy Network Adapte*r and make sure it is connected to the *IPS 2* virtual switch. When you're all done, click *Apply* to confirm these settings..


Settings for metasploitable 2 on BAHAMUT_

metasploitable 2

Hardware
- Add Hardware
- BIOS
  Boot from CD
- Security
  Key Storage Drive disabled
- Memory
  1024 MB
- Processor
  1 Virtual processor
- IDE Controller 0
  - Hard Drive
    Metasploitable.vhdx
- IDE Controller 1
  - ~~DVD Drive~~
    ~~None~~
- ~~SCSI Controller~~
- Legacy Network Adapter
  IPS 2
- ~~Network Adapter~~
  ~~Not connected~~
- COM 1
  None
- COM 2
  None
- Diskette Drive
  None

Management
- Name
  metasploitable 2
- Integration Services
  Some services offered
- Checkpoints
  Standard

Legacy Network Adapter

Specify the configuration of the network adapter or remove the network adapter.

Virtual switch:
IPS 2

VLAN ID

☐ Enable virtual LAN identification

The VLAN identifier specifies the virtual LAN that this virtual machine will use for all network communications through this network adapter.

2

To remove the network adapter from this virtual machine, click Remove.

Remove

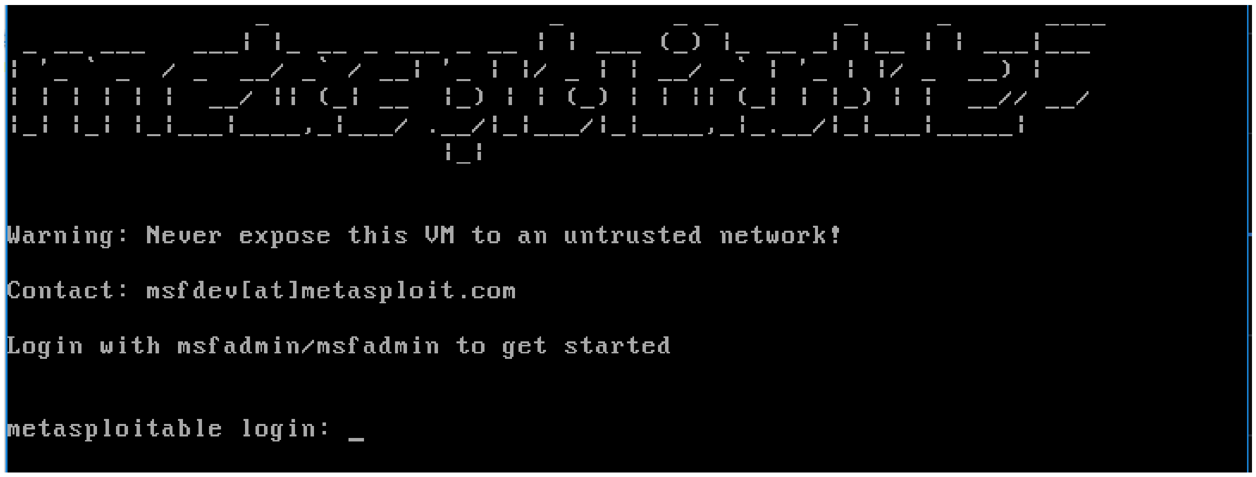
We recommend that you use a legacy network adapter only if you want to perform a network-based installation of the guest operating system, or if integration services are not installed in the guest operating system. Communications through a legacy network adapter are slower than through a network adapter.

OK Cancel Apply

The reason we're installing the *Legacy Network Adapter* is because Metasploitable 2 is very, *VERY* old. Its based on Ubuntu 8.04, which as of writing this, was released over 8 years ago. It doesn't have the drivers for the standard Hyper-V network adapter available. However, the legacy network adapter works just fine for our purposes, so we will be using that instead.

Now, power on the VM. we only need to make sure that it reaches the login prompt. Once you verify that the VM boots properly, Turn it off, then enter the VM's settings again. Select

Advanced Features under the *Legacy Network Adapter*, and **record the MAC address**, so that we can create a static DHCP mapping under OPT1 DHCP in pfSense. To do this, log in to the pfSense webConfigurator, navigate to *Services > DHCP Server*, and add a static mapping to the OPT1 interface DHCP, for the MAC address of the *Legacy Network Adapter* for the Metasploitable 2 VM ; assign it the IP address 172.16.2.3, making sure to enter a description that tells you which VM this static mapping belongs to. We are now finished configuring the Metasploitable 2 virtual machine.



```
                _                  _       _ _        _     _      ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: _
```

Note: This VM is located on *IPS 2* network, which we haven’t bridged to *IPS 1* network with the “IPS” VM yet. So in spite of having a static DHCP mapping, it has no way of reach out to the “pfSense” virtual machine for an IP address yet. We are going to fix this soon.

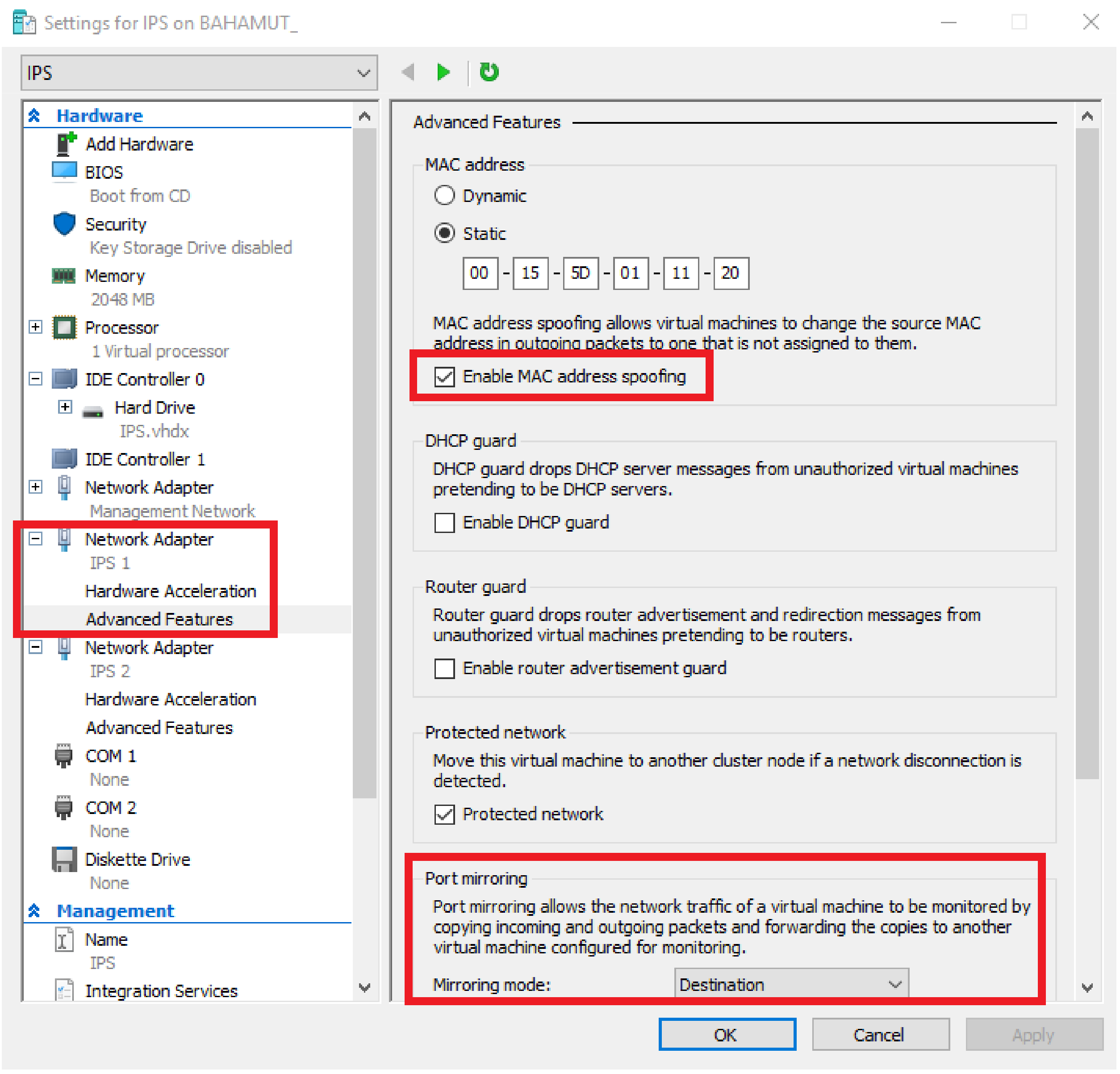
Port Mirroring and MAC spoofing

Mirroring network traffic is the process of taking normal network traffic from one or more network ports on a network device, and copying the network traffic from those source network ports to another port on the network device. Usually, this destination network port is some sort of a network inspection device, configured to passively analyze this traffic for threats, among other purposes. Port mirroring is typically associated with physical network switches, however Hyper-V mimics this functionality through its virtual switches. Hyper-V requires you to specify whether or not a VM’s network adapter is a source to mirror network traffic from, or a destination for receiving mirrored network traffic. The system that will be collecting network traffic must have it’s network adapter specified as a destination, and the hosts we want to to collect traffic from must be specified as sources.

In order for the AFPACKET bridge we will be implementing on our IPS VM to work properly, the network adapters of all the other VMs connected to the *IPS 1* and *IPS 2* networks must be defined as port mirroring sources, while our IPS VM network adapters on these networks must be specified as port mirroring destinations. In addition to that, those network adapters used to bridge the *IPS 1* and *IPS 2* networks must have MAC address spoofing enabled for the traffic to be passed across the bridge correctly. We will walk through configuring the IPS VM as a destination, and the pfSense VM as a source together, then I will have you configure the remaining VMs on your own.

Configuring the IPS VM as a Port Mirroring Destination

In Hyper-V Manager, open up the Settings menu for the “IPS” VM. Click on the *Network Adapter* connected to the *IPS 1* virtual switch. Double click on it, or click the + sign to bring up the *Advanced Features* menu. In the *MAC address* section, click the *Enable MAC address spoofing* checkbox. In the section labeled *Port mirroring*, Select *Destination* in the *Mirroring mode:* drop-down.



Advanced Features

MAC address

○ Dynamic

● Static

00 - 15 - 5D - 01 - 11 - 20

MAC address spoofing allows virtual machines to change the source MAC address in outgoing packets to one that is not assigned to them.

☑ Enable MAC address spoofing

DHCP guard

DHCP guard drops DHCP server messages from unauthorized virtual machines pretending to be DHCP servers.

☐ Enable DHCP guard

Router guard

Router guard drops router advertisement and redirection messages from unauthorized virtual machines pretending to be routers.

☐ Enable router advertisement guard

Protected network

Move this virtual machine to another cluster node if a network disconnection is detected.

☑ Protected network

Port mirroring

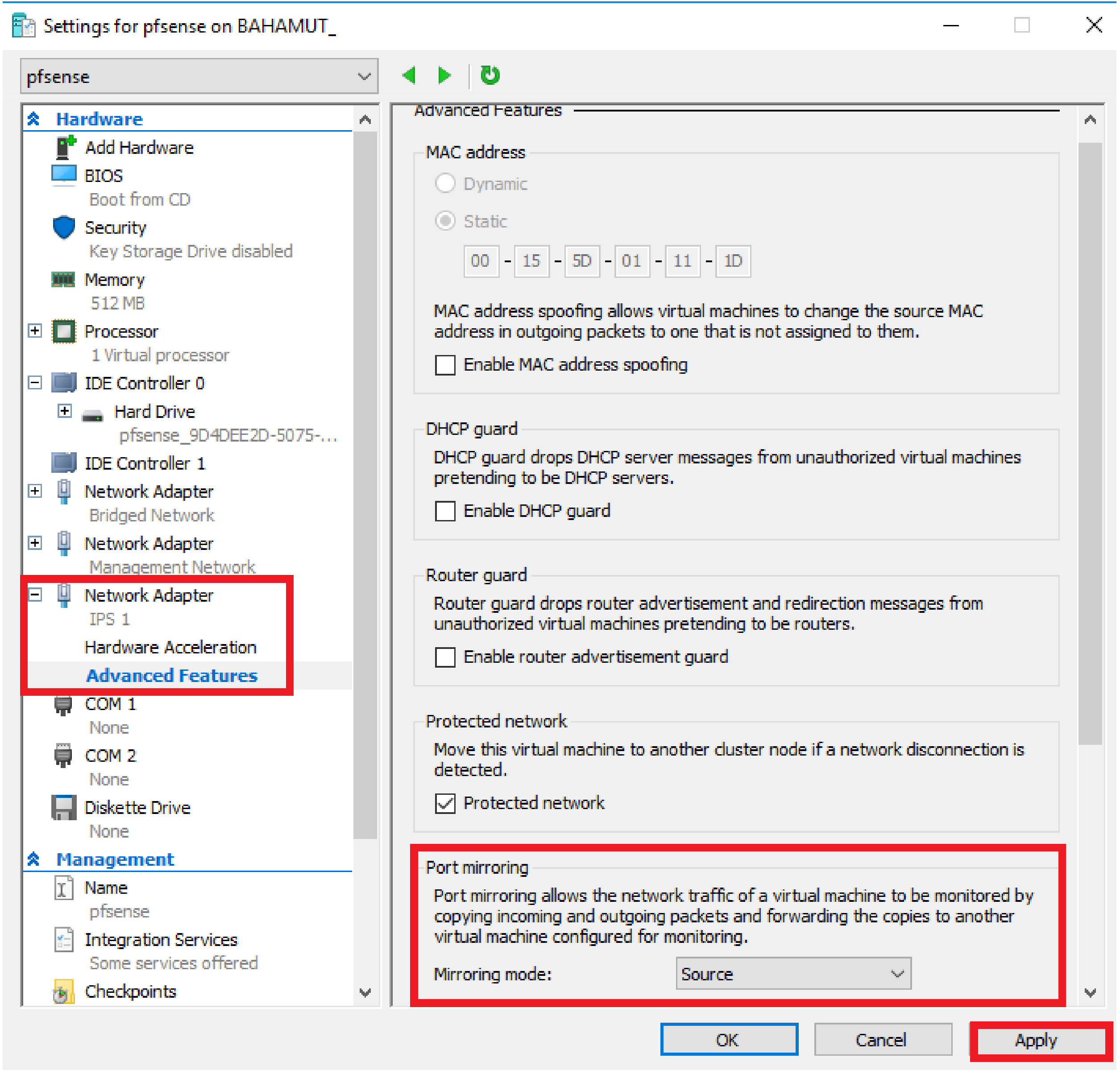
Port mirroring allows the network traffic of a virtual machine to be monitored by copying incoming and outgoing packets and forwarding the copies to another virtual machine configured for monitoring.

Mirroring mode: Destination

Repeat this process for the *IPS 2 Network Adapter* (enable MAC address spoofing, and set the adapter as a port mirroring destination), then click *Apply*. At this point, we have the "IPS" VM set up and ready to go, but now, we have to configure all the remaining VMs connected to the *IPS 1* and *IPS 2* networks as sources to send their traffic to the IPS VM's network adapters.

Configuring the pfSense VM as a Port Mirroring Source

In the Hyper-V Manager, right click on pfSense and choose settings to open the VM's settings. Double click on the *Network Adapter* connected to the *IPS 1* virtual switch, or click the + sign to display the *Advanced Features* menu. Under the *Port mirroring* section, select *Source* in the Mirroring mode: drop-down. *Apply* these changes.



Settings for pfsense on BAHAMUT_

pfsense

Hardware
- Add Hardware
- BIOS
  Boot from CD
- Security
  Key Storage Drive disabled
- Memory
  512 MB
- Processor
  1 Virtual processor
- IDE Controller 0
  - Hard Drive
    pfsense_9D4DEE2D-5075-...
- IDE Controller 1
- Network Adapter
  Bridged Network
- Network Adapter
  Management Network
- Network Adapter
  IPS 1
  - Hardware Acceleration
  - **Advanced Features**
- COM 1
  None
- COM 2
  None
- Diskette Drive
  None

Management
- Name
  pfsense
- Integration Services
  Some services offered
- Checkpoints

Advanced Features

MAC address

Dynamic

[✓ Static]

00 - 15 - 5D - 01 - 11 - 1D

MAC address spoofing allows virtual machines to change the source MAC address in outgoing packets to one that is not assigned to them.

☐ Enable MAC address spoofing

DHCP guard

DHCP guard drops DHCP server messages from unauthorized virtual machines pretending to be DHCP servers.

☐ Enable DHCP guard

Router guard

Router guard drops router advertisement and redirection messages from unauthorized virtual machines pretending to be routers.

☐ Enable router advertisement guard

Protected network

Move this virtual machine to another cluster node if a network disconnection is detected.

☑ Protected network

Port mirroring

Port mirroring allows the network traffic of a virtual machine to be monitored by copying incoming and outgoing packets and forwarding the copies to another virtual machine configured for monitoring.

Mirroring mode: Source

Port Mirroring for the Remaining VMs

Now that you understand how to configure port mirroring, you will need to configure the network adapters on the Kali Linux and Metasploitable 2 VMs as port mirroring sources. Simply follow the same directions I laid out above for the pfSense VM to configure the remaining VMs as port mirroring sources. After you have completed this task, all that's left is to install and configure Snort or Suricata on the IPS VM to pass traffic between the *IPS 1* and *IPS 2* virtual switches.

Note: Please be aware that if you choose to customize your lab network, and add additional VMs to the *IPS 1* or *IPS 2* networks, that in order for those VMs to utilize the AFPACKET bridge provided by the IPS VM, each new VM will need to be specified as a port mirroring source.

Next Steps

The Client Hyper-V configuration is all but done at this point. However, you're not quite done yet. Here is a checklist of tasks to complete:

- If you haven't completed the chapter Defense in Depth for Windows Hosted Hypervisors, I would very highly suggest doing so in order to harden your hypervisor host.
- While not strictly necessary, you may also want to complete the Remote Lab Management guide. Specifically, the sections related to Windows Remote Access, Enabling SSH on Kali Linux, and if you're lazy like me, the section on Securing root SSH access. This will allow you to remotely manage all of your lab VMs, as well as finish the IPS and Splunk setup guides much more easily than through the Hyper-V console.
- You still need to install IDS/IPS software on the IPS VM to get a functioning AFPACKET bridge. Check out the IPS Installation Guide to learn how to do this with either Snort or Suricata as your IPS software of choice.
- The SIEM VM needs to have Splunk installed and configured. You'll need to complete the Splunk Installation Guide.
- You may want to consider reading the Automated Patching for Linux Lab VMs chapter, and implementing the `updater.sh` script for your Linux VMs.
- Do you want some ideas on where to take your lab? Check out the chapter In Your Own Image for some tips on how to mold your VM lab to better suit your needs.

Setup - Oracle VirtualBox

Note: Please be aware that this guide was written using VirtualBox version 5.1.6, with Windows 10 as the host operating system. The host OS should not matter since the graphical interface of VirtualBox is fairly consistent across operating systems. Also, for the sake of sanity, we're not going to cover `VBoxManage` (a command line management interface for VirtualBox) in this guide.

This guide will instruct you on how to create the our lab environment on Oracle's VirtualBox hosted hypervisor. You will learn how to install and customize VirtualBox, create your first virtual machine, and finally, create and configure the remaining VMs for your lab environment.

Installation

VirtualBox's biggest draws are that it's 100% free, and that it's multi-platform: VirtualBox can be installed on practically every major operating system out there today, and it can virtualize practically any operating system today, provided you meet the hardware requirements. To get started, open your favorite web browser and navigate to https://www.virtualbox.org/wiki/Downloads and download the latest version of VirtualBox for the host OS you are using, then run the installer.

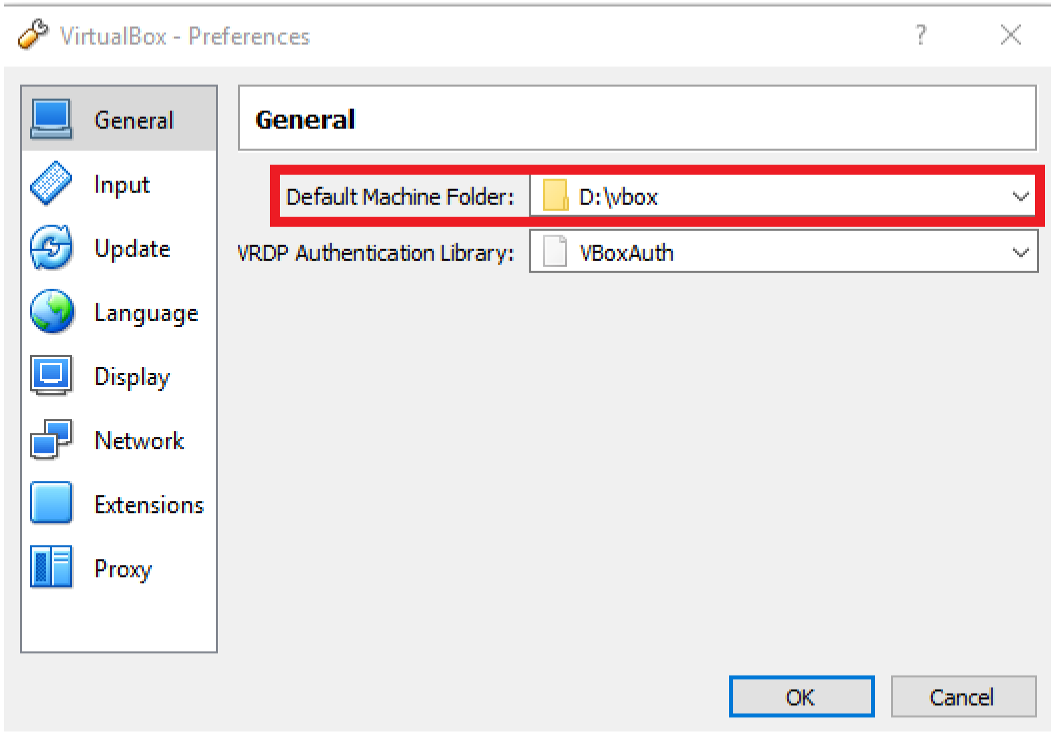
Note: Linux and BSD hosts can install VirtualBox through their respective package manager (e.g. `yum`, `apt-get`, `pkg`, etc.). For instructions on how to install VirtualBox on Linux via yum or apt-get visit https://www.virtualbox.org/wiki/Linux_Downloads. For instructions on how to install VirtualBox on FreeBSD, visit https://www.freebsd.org/doc/handbook/virtualization-host-virtualbox.html.

No matter what operating system you choose to run VirtualBox on, the installation is usually straightforward; simply proceed with the default installation options and reboot the host as necessary.

Hypervisor Preferences

So now that we have VirtualBox installed, let's get some general settings sorted before we create our first virtual machine. Open VirtualBox, then open the VirtualBox preferences menu. This can be done on Windows and Linux by clicking *File > Preferences*. On OS X, Click on *VirtualBox > Preferences*.

The *General* screen should be displayed. Pay attention to the setting *Default Machine Folder:*; If you have multiple drives or partitions and you wanted to have your virtual machine files stored on a particular drive, create a folder on that drive and edit the default path by clicking the drop-down arrow, and selecting *Other…* VirtualBox will then open an explorer window for you to navigate to the folder you want to use to store your VM files and folders. In the case of my setup, I chose to have my VirtualBox files stored in `D:\vbox`.

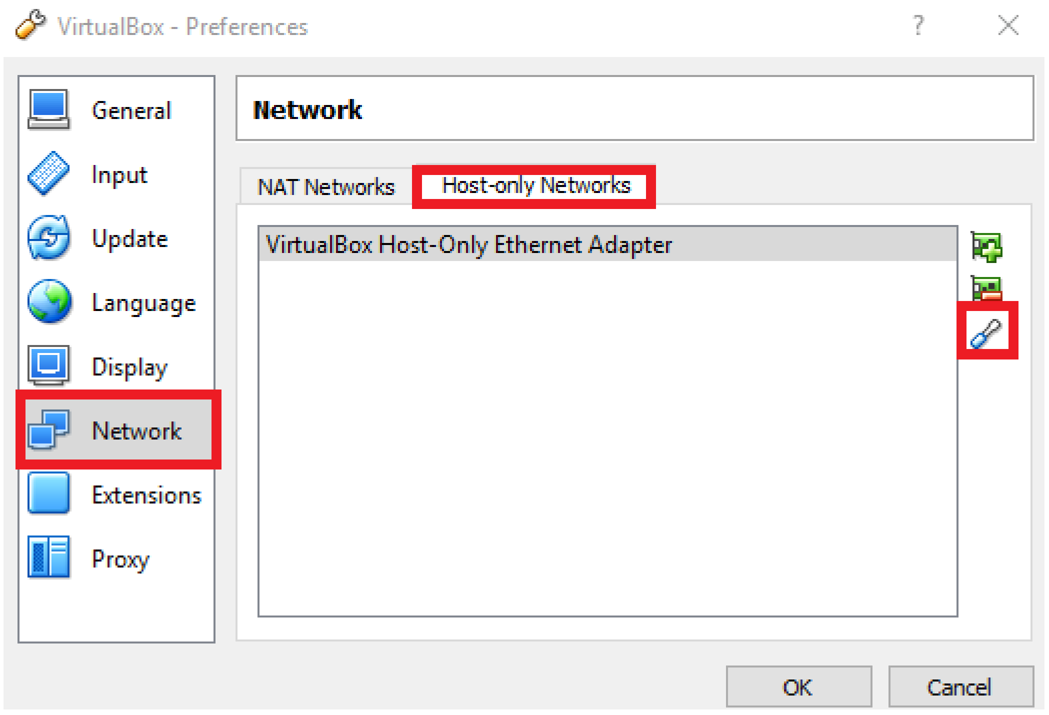


Default Machine Folder: D:\vbox

VRDP Authentication Library: VBoxAuth

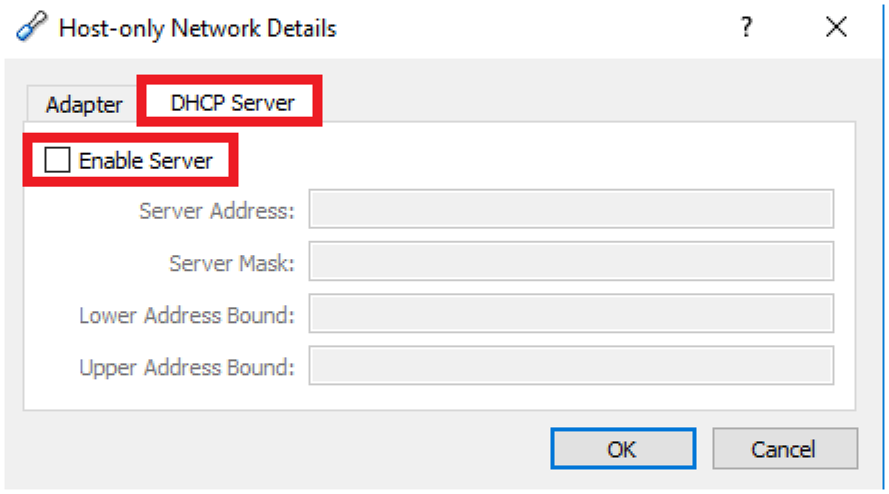
As discussed in the Hardware Considerations section, storing your VMs on a separate physical drive is generally a good idea to ensure that your VMs are NOT competing for disk I/O with the host operating system.

The next order of business is to disable DHCP for the host-only network, since we want the pfSense VM to handle DHCP services. To do this, click on *Network*, then click on the *Host-only Networks* tab in the new pane that appears. If you are running Windows, VirtualBox creates your first host-only network, and names it `VirtualBox Host-Only Ethernet Adapter`. If you are running Linux, OS X or BSD, you will first need to click the network card icon with a + sign on it to the right of the *Host-Only Networks* pane to create your system's first host-only network. In this case, this first host-only network will be named *vboxnet0* on non-Windows operating systems. Ensure the host-only network is highlighted, then click third icon on the right in the window that looks like a screwdriver.



VirtualBox Host-Only Ethernet Adapter

In the next window, click on the *DHCP Server* tab, then uncheck the *Enable Server* Checkbox. Click OK to exit.


Host-only Network Details

Adapter | DHCP Server

☐ Enable Server

Server Address:

Server Mask:

Lower Address Bound:

Upper Address Bound:

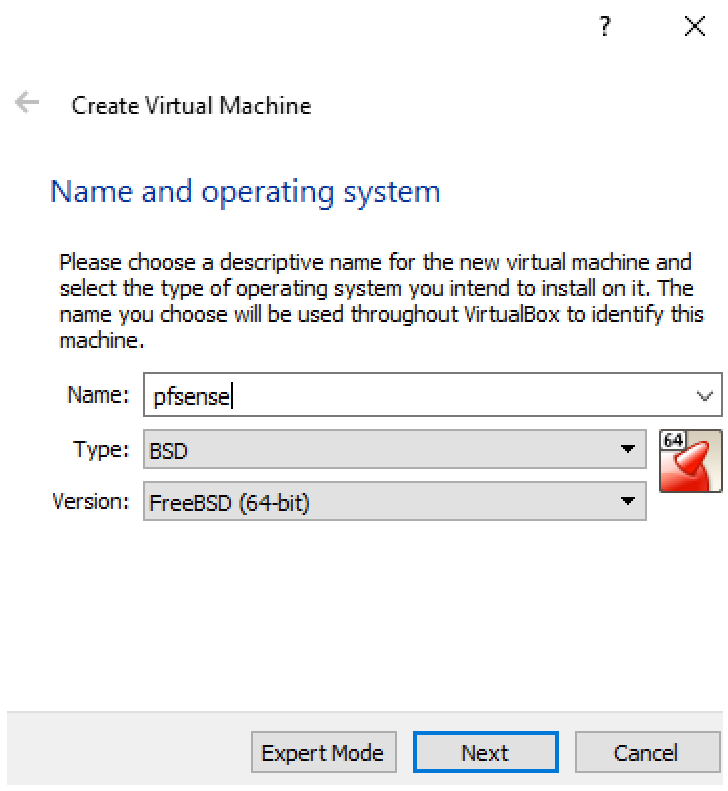
Creating the first VM, pfSense

pfSense is the keystone holding this entire configuration together. Personally, it's my favorite firewall distro due to ease of use, the amount of functionality it includes out of the box, combined with a plugin/add-on system for additional functionality. If you have the CPU, RAM, and disk, pfSense can easily be converted into a so-called “Next-Generation” firewall.

Make your way to https://www.pfsense.org/download/. Download the latest installation ISO for the amd64 architecture. While you’re at it, you may want to download a compression utility. The pfSense maintainers distribute pfSense as an ISO image file compressed with `gzip`. This means that we’ll need to decompress the ISO file at some point. On Windows, I prefer 7-Zip (http://www.7-zip.org/) as my compression utility of choice for decompressing files, since 7-Zip can handle zip, gzip, rar, and 7z files (among others) easily. Linux, BSD, and OS X usually include the `gzip` or `gunzip` command line utilities for decompressing gzipped files.

Adding a New VM

In VirtualBox, click the *New* icon to be taken through the *Create Virtual Machine* wizard. The first page will have you choose a name for your new VM, as well as assigning the type of OS you'll be installing and the version. In our case, I named the VM pfSense. pfSense is BSD derived, and is based on FreeBSD, so I chose *BSD* as the type, and *FreeBSD (64-bit)* as the Version. Click the *Next* button to continue.



Name and operating system

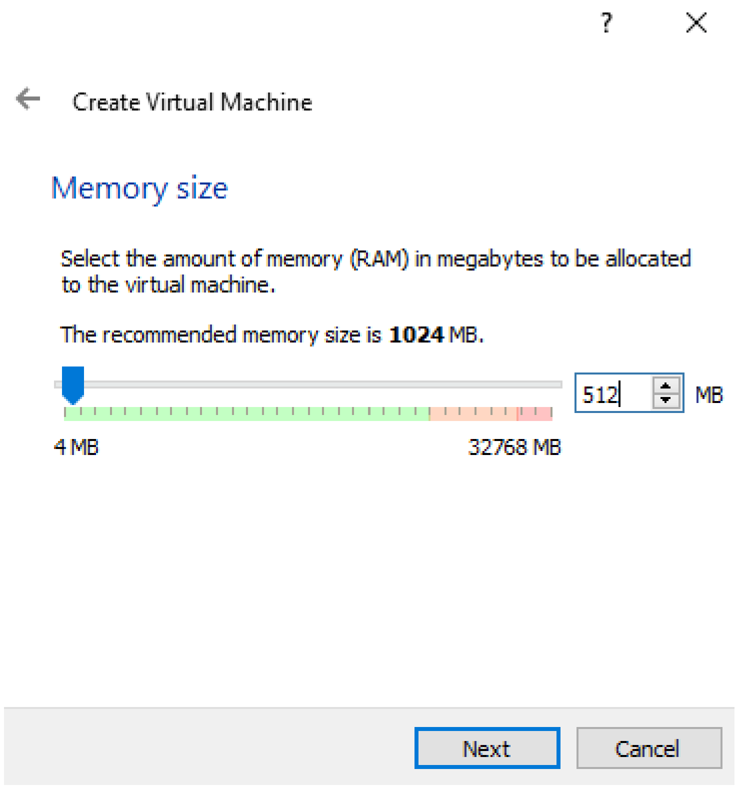
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: pfsense

Type: BSD

Version: FreeBSD (64-bit)

On the next screen, you'll be asked how much RAM you wish to allocate to this VM. 512MB is sufficient for this VM. After inputting 512MB, click *Next*.



Memory size

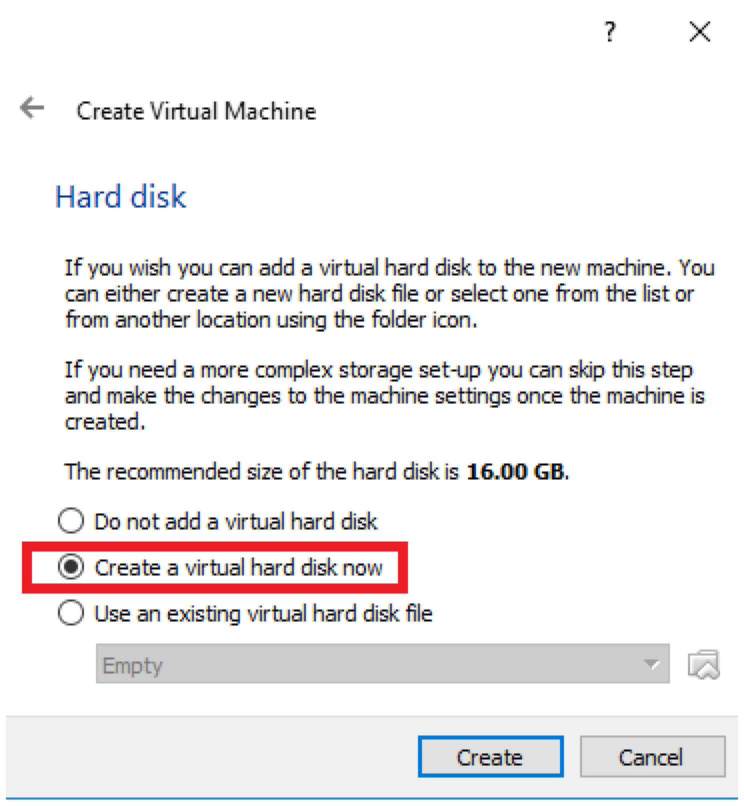
Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

512 MB

4 MB	32768 MB

The next screen is where we allocate hard drive space for the new VM. Click the *Create a virtual hard disk now* radio button, then click *Create*.



Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

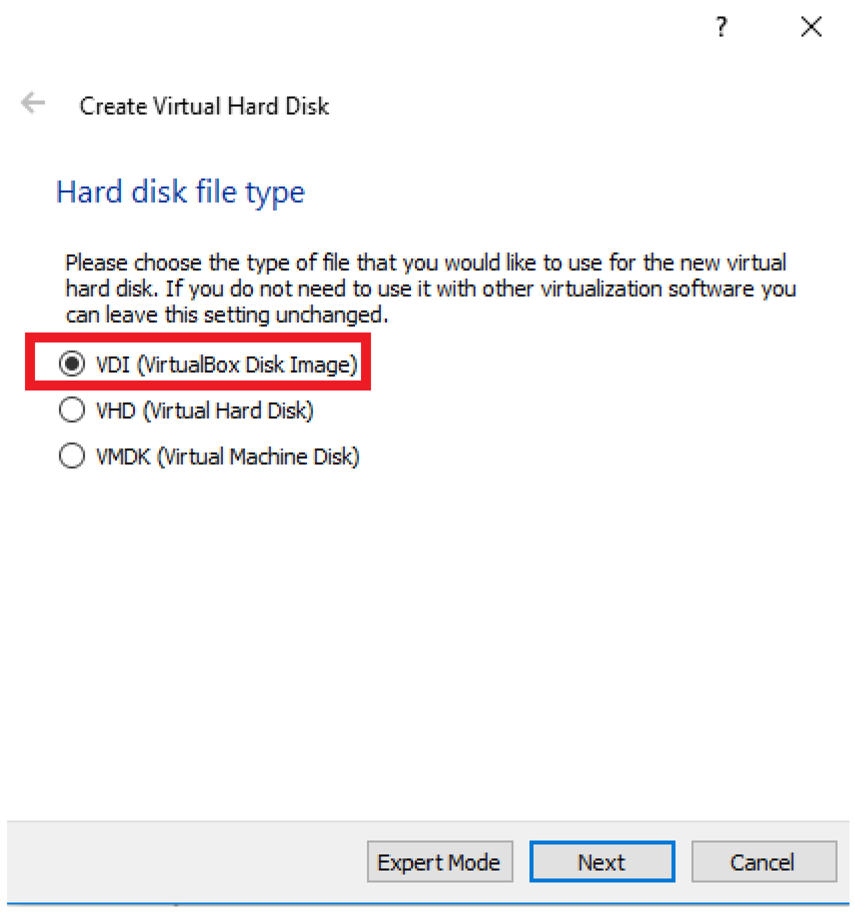
If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **16.00 GB**.

- ○ Do not add a virtual hard disk
- ● Create a virtual hard disk now
- ○ Use an existing virtual hard disk file

Empty

On the next screen, select the *VDI (VirtualBox Disk Image)* radio button and click *Next*.

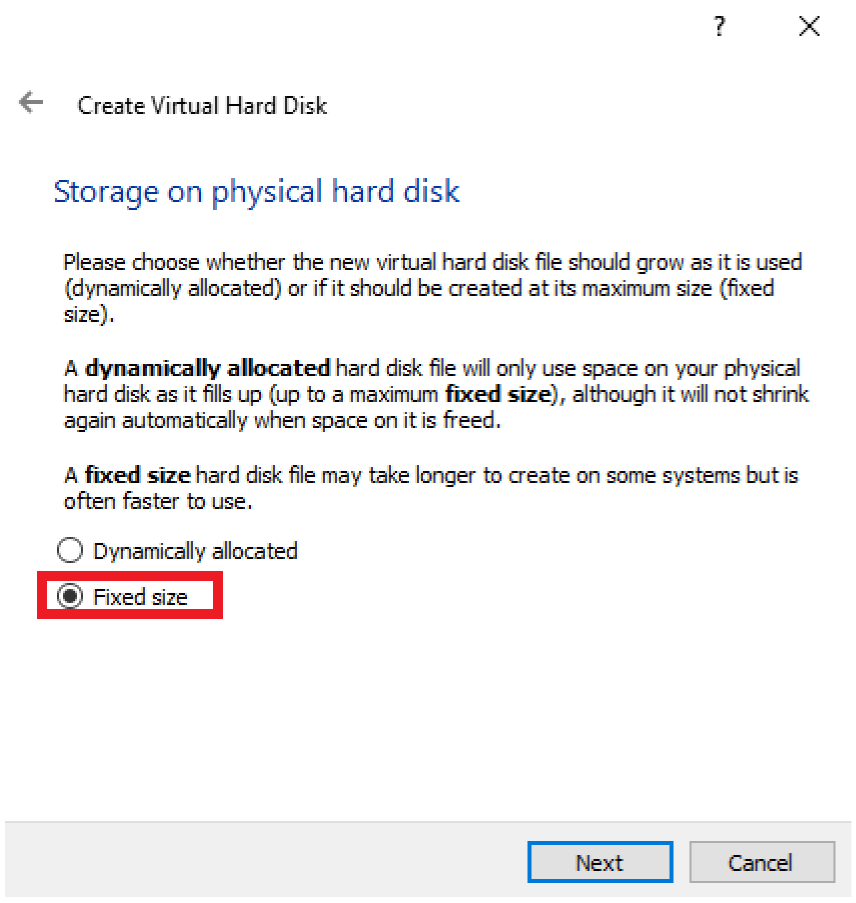

Create Virtual Hard Disk

Hard disk file type

Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

[✓ VDI (VirtualBox Disk Image)]
VHD (Virtual Hard Disk)
VMDK (Virtual Machine Disk)

This screen asks if you want to use a fixed or dynamically allocated disk. Dynamic allocation allows the VDI file to expand in file size up to the maximum size we configure for it. This may lead to some I/O and/or CPU penalties as the file allocation is expanded, and more disk space is utilized in the VM. Choosing fixed means that the full amount of space we request for the VM is immediately allocated to the corresponding VDI file. Click the *Fixed size* radio button, then click *Next*.



Storage on physical hard disk

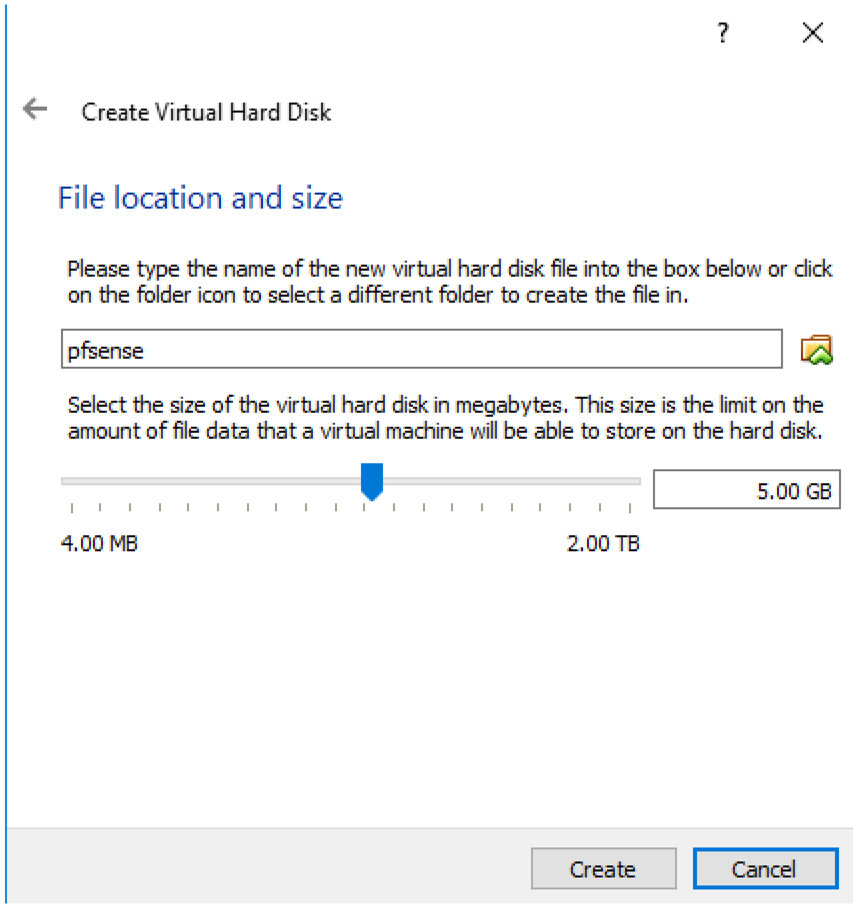
Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

Dynamically allocated [✓ Fixed size]

On the next screen, we allocate disk space that we'll be using to install pfSense to our VDI container. 5GB is what I recommend, but if you plan on utilizing more features, adjust the the size of your VM's virtual hard disk accordingly. Click *Create* and wait for VirtualBox to write the initial VDI file.



File location and size

Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

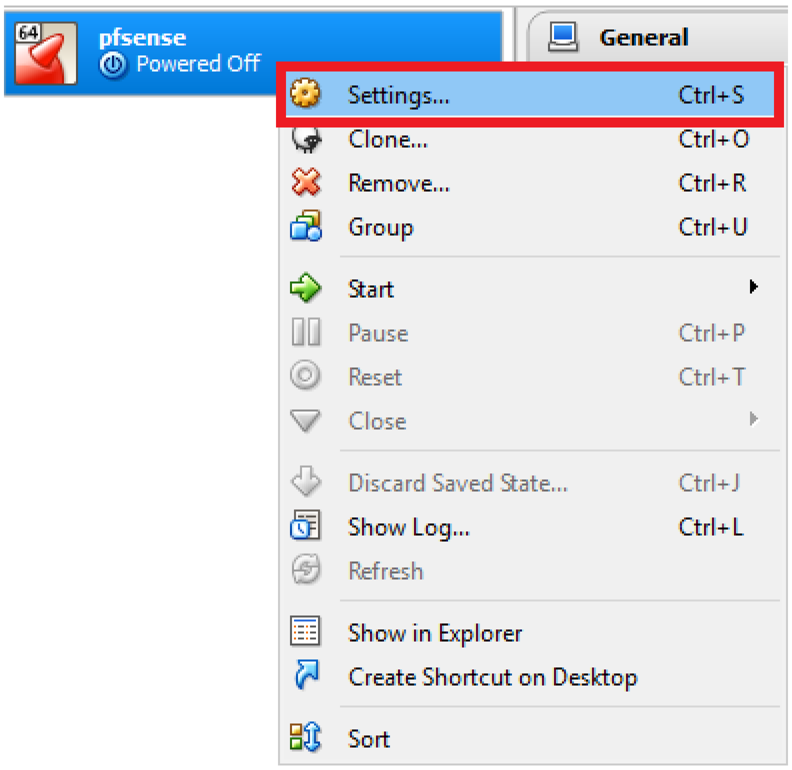
pfsense

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

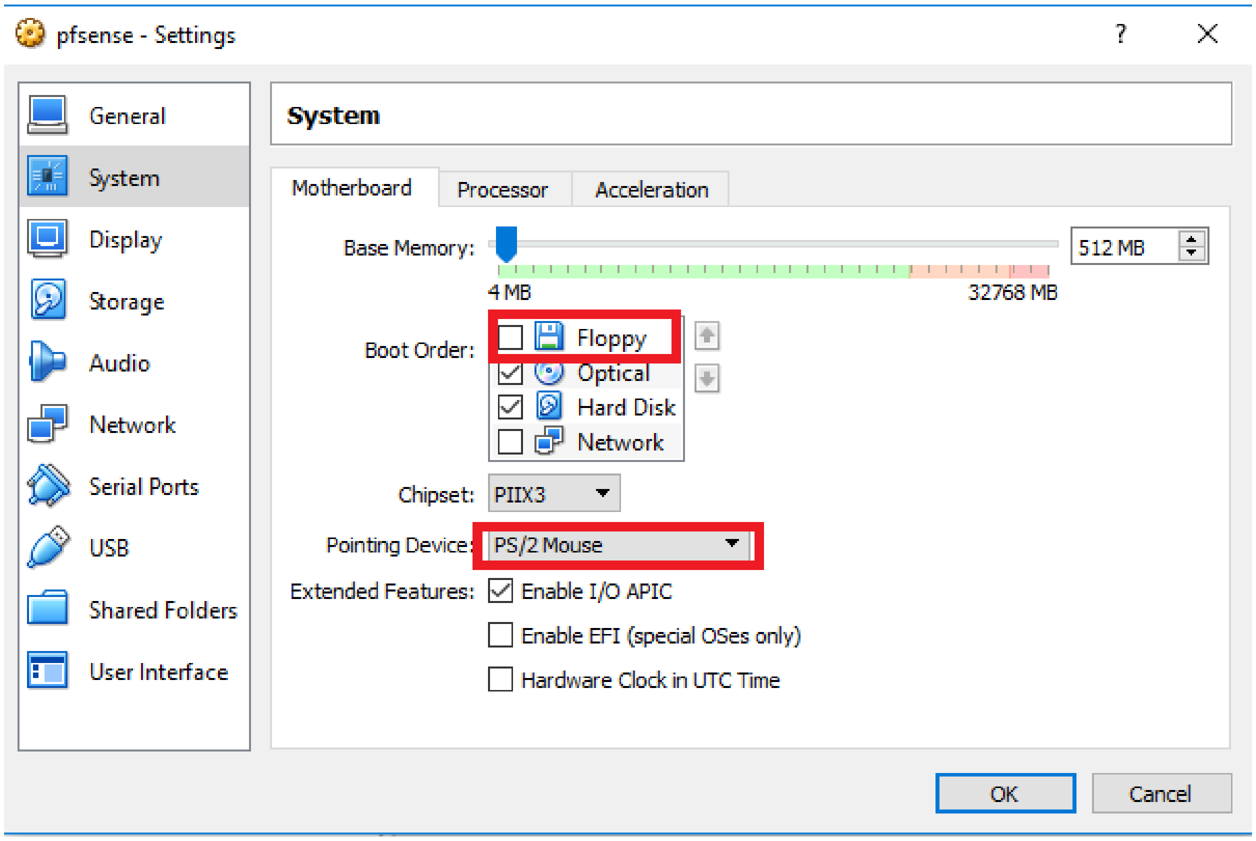
5.00 GB

Initial VM Settings

At this point, there should be a VM in the VirtualBox menu labeled “pfSense” listed in the main menu. It should be displayed as *Powered Off*. Before we power it on, there are a number of configuration changes we need to make. Right click on “pfSense”, and select the *Settings…* option.



The settings menu for the VM will be displayed. Let's begin by clicking on *System*. This causes a new pane to appear with a variety of system hardware settings. On the *Motherboard* tab, In the *Boot Order:* selection, uncheck the *Floppy* checkbox, then navigate to the *Pointing Device:* drop-down menu, and make sure that *PS/2 Mouse* is selected.



pfsense - Settings

System

Motherboard | Processor | Acceleration

Base Memory: 512 MB (4 MB – 32768 MB)

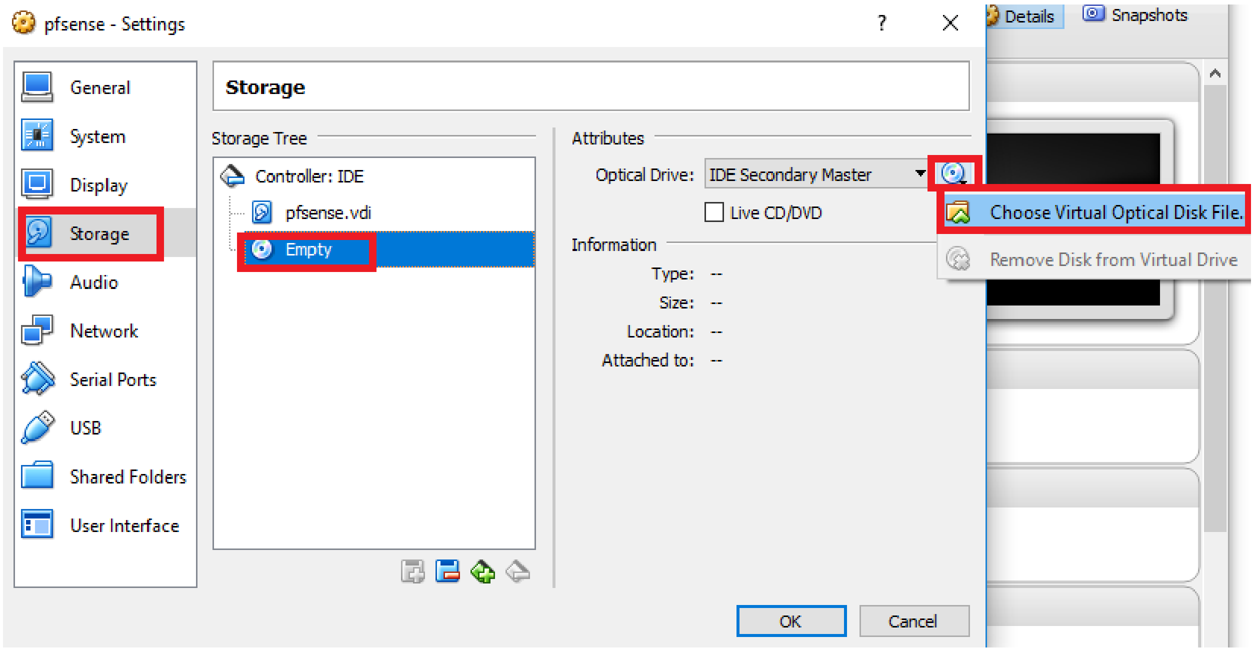
Boot Order:
- [ ] Floppy
- [✓] Optical
- [✓] Hard Disk
- [ ] Network

Chipset: PIIX3

Pointing Device: PS/2 Mouse

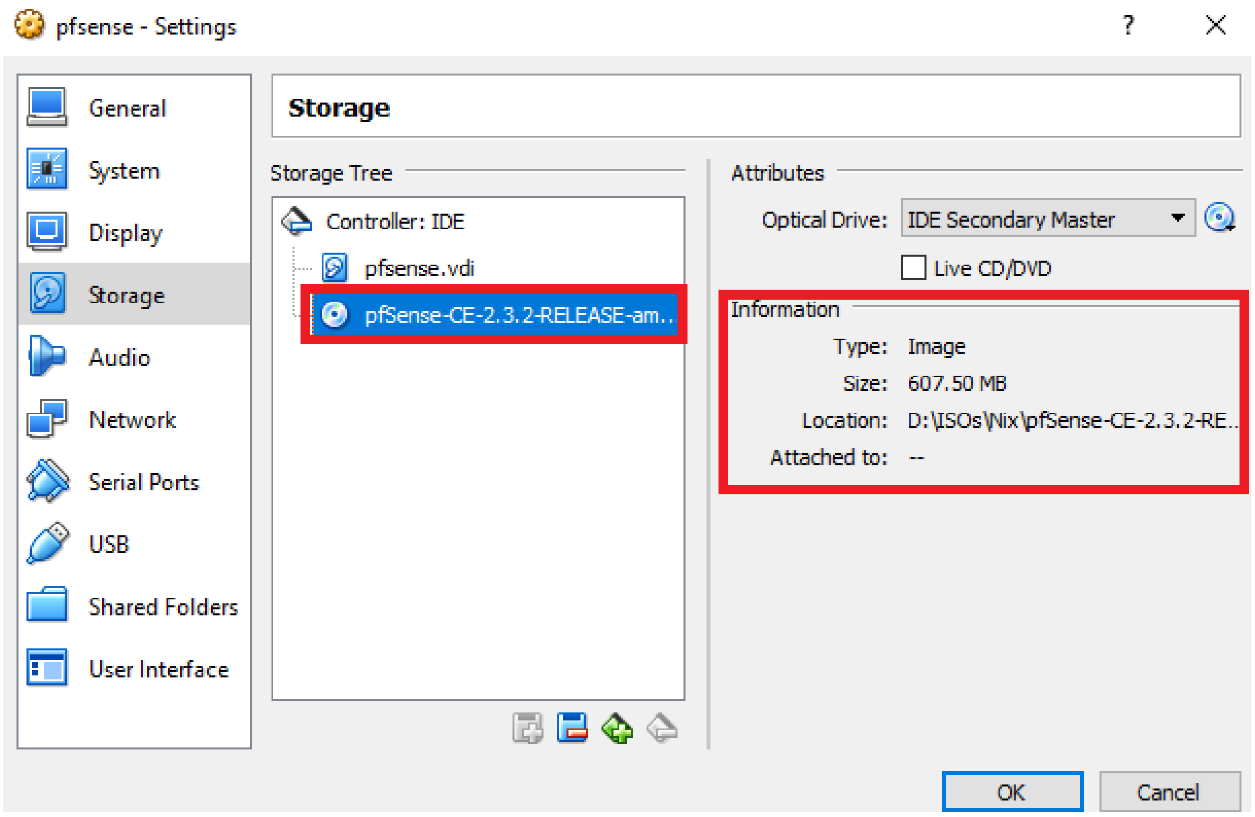
Extended Features:
- [✓] Enable I/O APIC
- [ ] Enable EFI (special OSes only)
- [ ] Hardware Clock in UTC Time

Next, click on *Storage*, then click on the icon that looks like a CD under the *Storage Tree* pane. Then, under the *Attributes* portion of the screen, click on the small icon that looks like a CD. On the drop-down menu that appears, then click on the *Choose Virtual Optical Disk File...* option.This will cause an explorer window to pop up. Navigate to the directory where your pfSense ISO is located and double click on the .iso file to select it.



Note: VirtualBox is NOT capable of reading the pfSense ISO while it is compressed (.gz). You will need to decompress it first. As stated previously, I recommend using 7-Zip on Windows, or the `gzip` command on Linux, BSD, or OS X to decompress the ISO.

The reason we are doing this is that we need to tell the VM to boot from this ISO file (that we're treating as though it is a virtual CD/DVD drive) in order to install an operating system to the VDI container. The *Storage* pane should look like this after finding and selecting the pfSense ISO:



Optical Drive: IDE Secondary Master

Live CD/DVD

Information

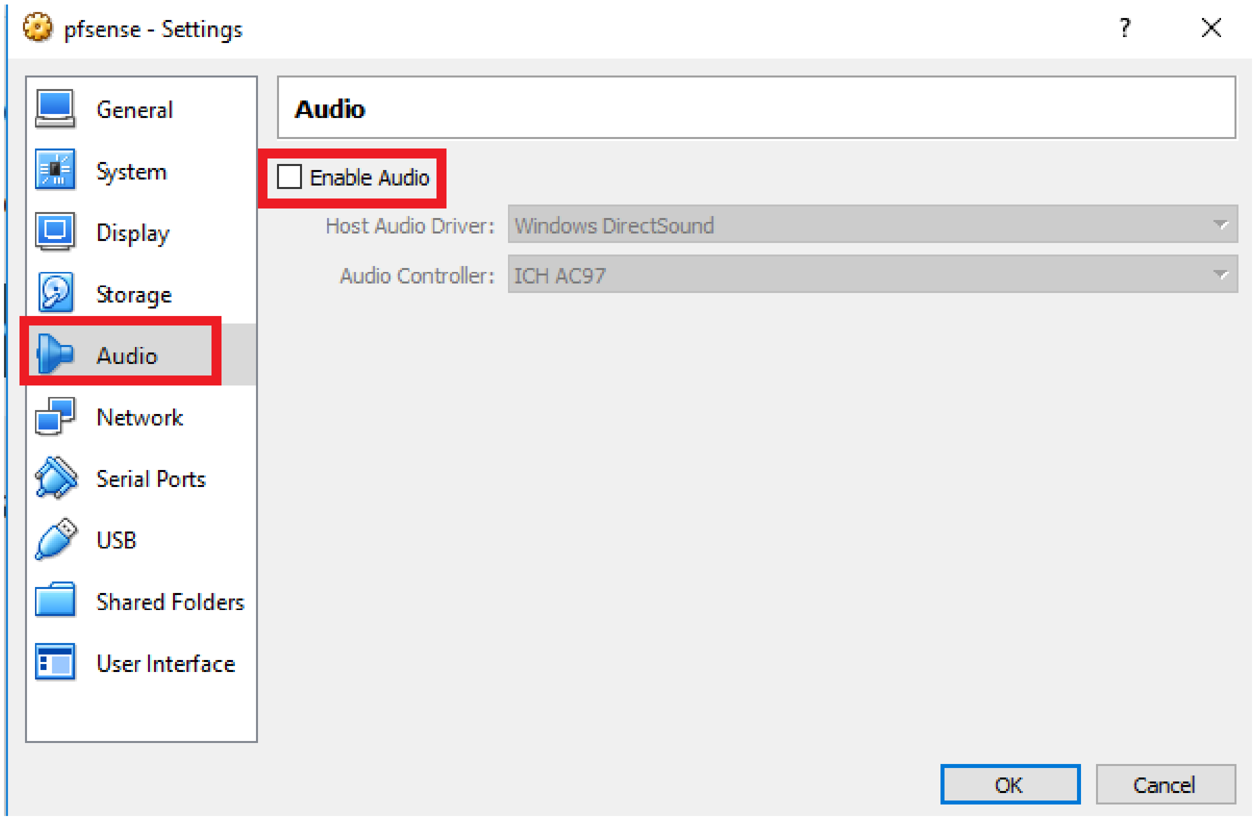
Type: Image

Size: 607.50 MB

Location: D:\ISOs\Nix\pfSense-CE-2.3.2-RE..

Attached to: --

Next, click on *Audio*, and uncheck the *Enable Audio* checkbox.



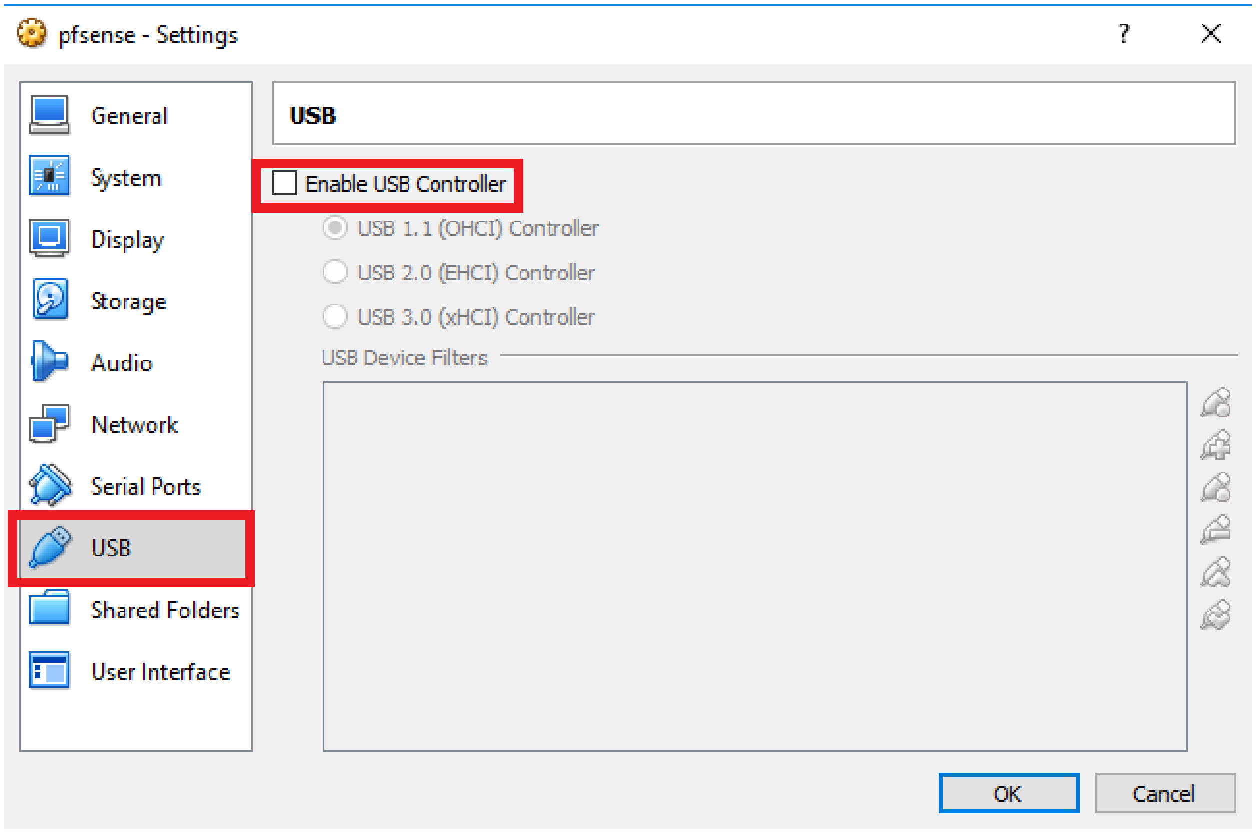
Audio

Enable Audio [ ]

Host Audio Driver: Windows DirectSound

Audio Controller: ICH AC97

Next, click on *USB* (OS X users should click on *Ports*, then click on the the *USB* tab, instead). Uncheck the *Enable USB Controller* checkbox to disable USB support for the VM. We are disabling the audio and USB controllers for our VM in order to reduce possible attack surface for virtual machine escapes, as well as ensure that host and VM are as isolated from one another as reasonably possible.



USB

☐ Enable USB Controller

● USB 1.1 (OHCI) Controller

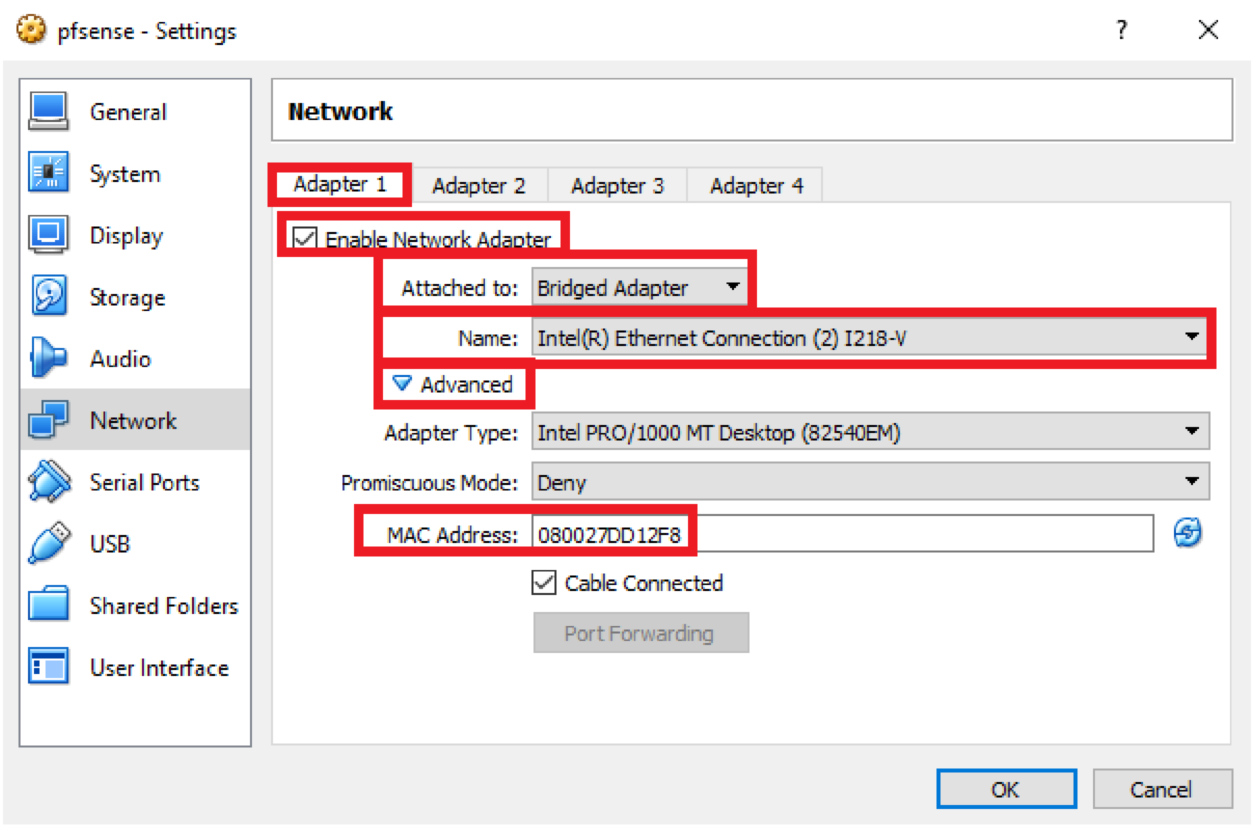
○ USB 2.0 (EHCI) Controller

○ USB 3.0 (xHCI) Controller

USB Device Filters

Note: Virtual machine escapes or "VM escapes" are a specific type of exploit that an attacker will attempt to use in order to gain some sort of execution or access to resources on the hypervisor host itself. While VM escapes that result in code execution on the hypervisor host are rare, there are other means that attackers or malware could potentially abuse to attack the hypervisor host itself, such as through shared folders (file system folders that both the host and the VM can read/write to) or through guest extension software packages (special software packages installed on the VM itself that enable extra functionality, or access to host system resources). Therefore it is important to make your lab VMs as lean as reasonably possible in order to avoid compromising the hypervisor host system.

Finally, click on "Network". You will be greeted by a window pane that has multiple tabs. The first tab is labeled "Adapter 1". First, check that the "Enable Network Adapter" checkbox is checked. On the "Attached to:" drop-down, make sure that "Bridged Adapter" is selected. Under the “Name:” drop-down, choose the physical network card you want to have virtual machine bridge to. Select the network interface you typically use on the host to connect to the internet. Click on the“Advanced” arrow to display more network adapter options. One of the fields that gets displayed is labeled “MAC Address:”. Document the MAC address displayed.



Enable Network Adapter ☑

Attached to: Bridged Adapter

Name: Intel(R) Ethernet Connection (2) I218-V

Advanced

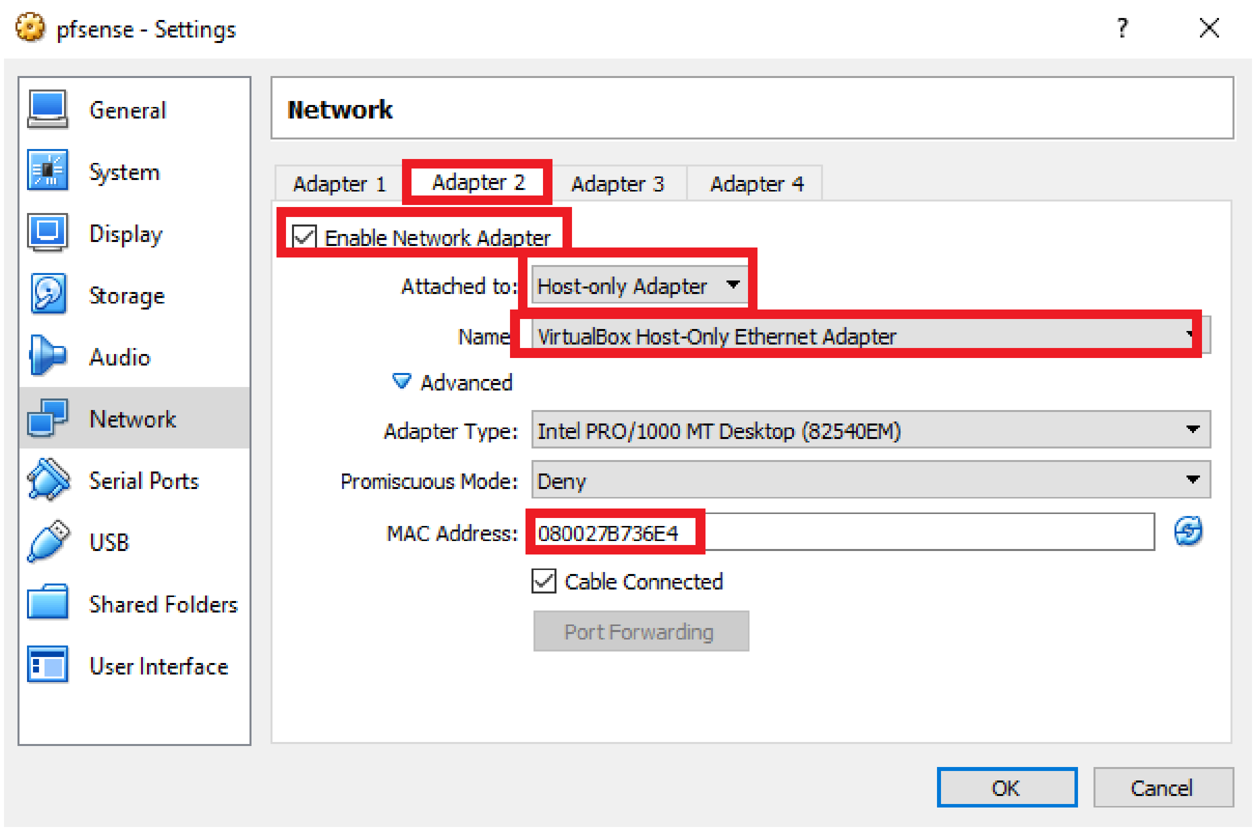
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 080027DD12F8

Cable Connected ☑

Click on the “Adapter 2” tab, and enable the network adapter if it is disabled, just like with “Adapter 1”. Next, set “Attached to:” to “Host-Only Adapter”, and “Name:” should default to “VirtualBox Host-Only Ethernet Adapter” on Windows, or “vboxnet0” on non-Windows hosts. Just like with “Adapter 1”, click on “Advanced”, and record the MAC address of this interface.


pfsense - Settings

General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

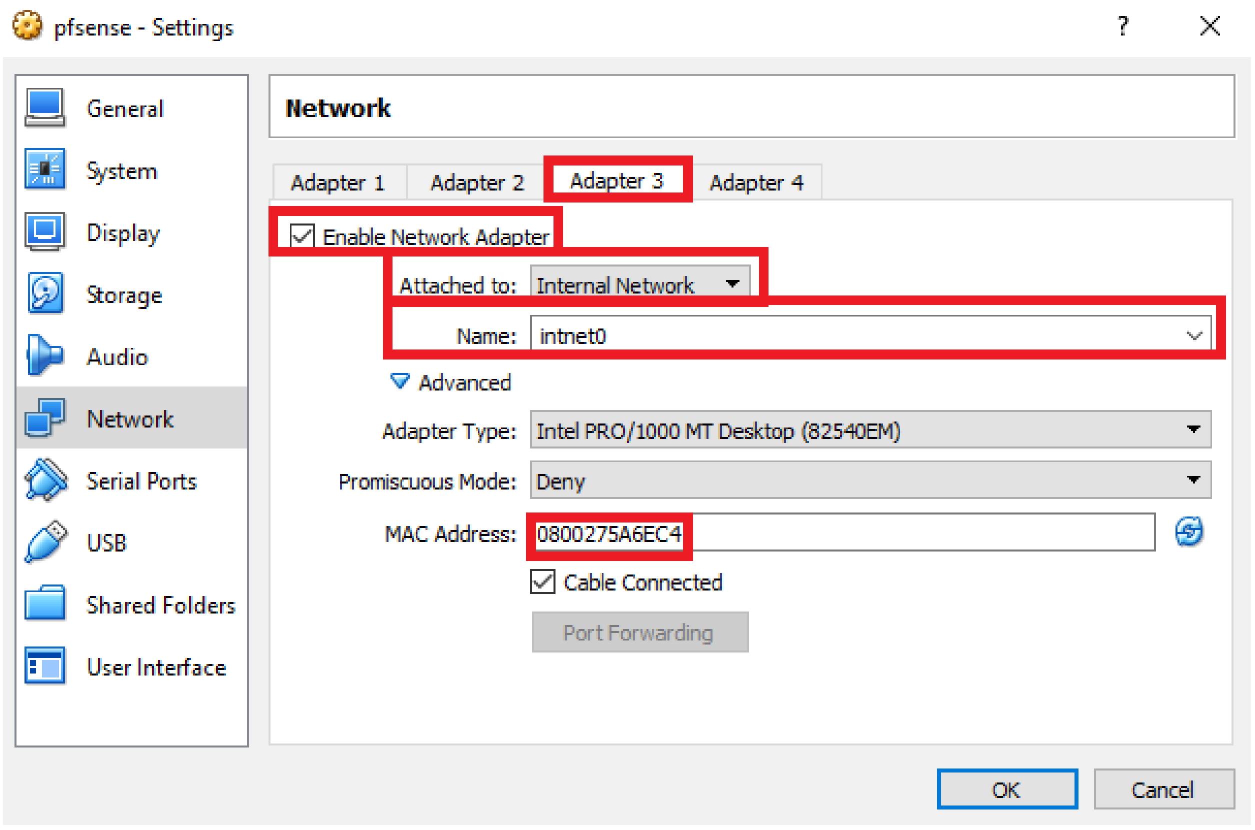
Promiscuous Mode: Deny

MAC Address: 080027B736E4

☑ Cable Connected

Finally, click on the “Adapter 3” tab. Enable the adapter, and set “Attached To:” to “internal network”. The default internal network name will be “intnet” on Windows, or “intnet0” on non-Windows hosts. The default internal network name (in either case) is perfectly fine for this network adapter, just make sure you document the internal network name somewhere. Make sure to record the internal network name for this adapter for later reference. Be sure to click

"Advanced" and document the MAC address of this network adapter as well. Click OK to exit the settings menu.



Network

☑ Enable Network Adapter

Attached to: Internal Network

Name: intnet0

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

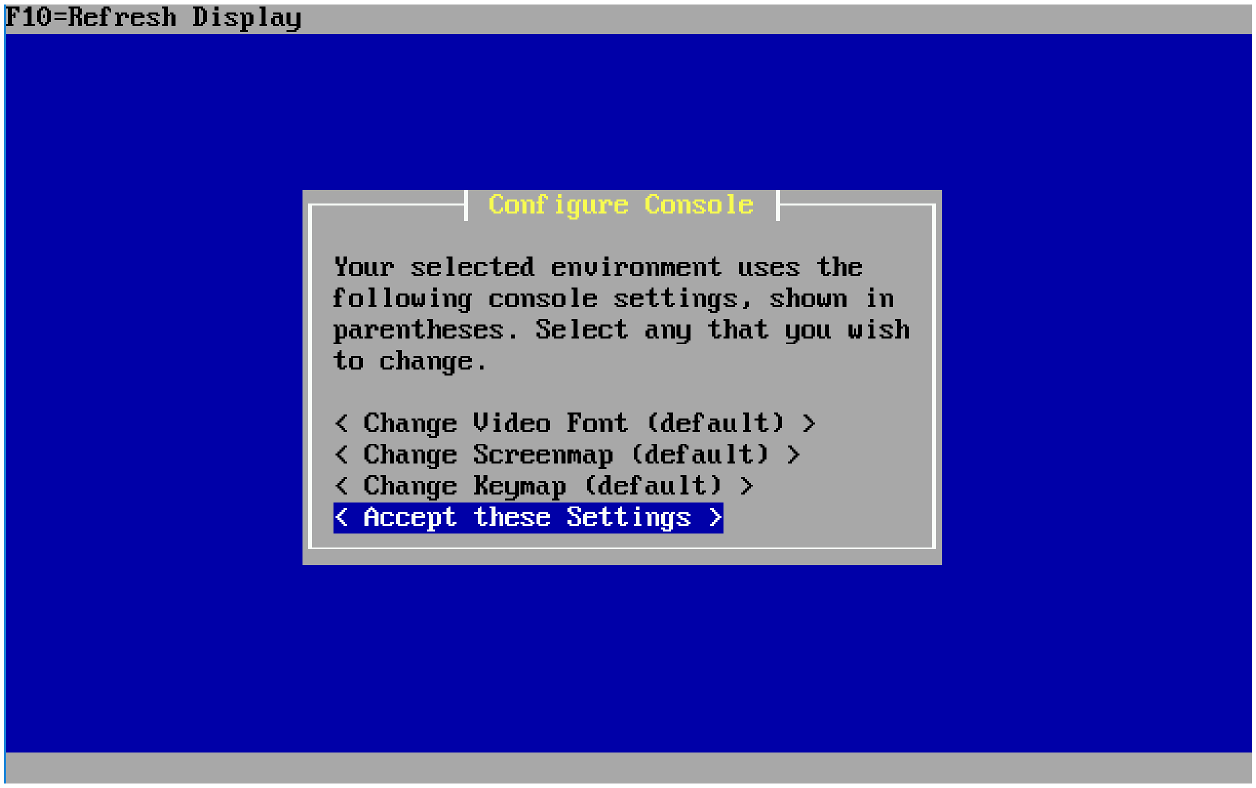
MAC Address: 0800275A6EC4

☑ Cable Connected

Installing pfSense

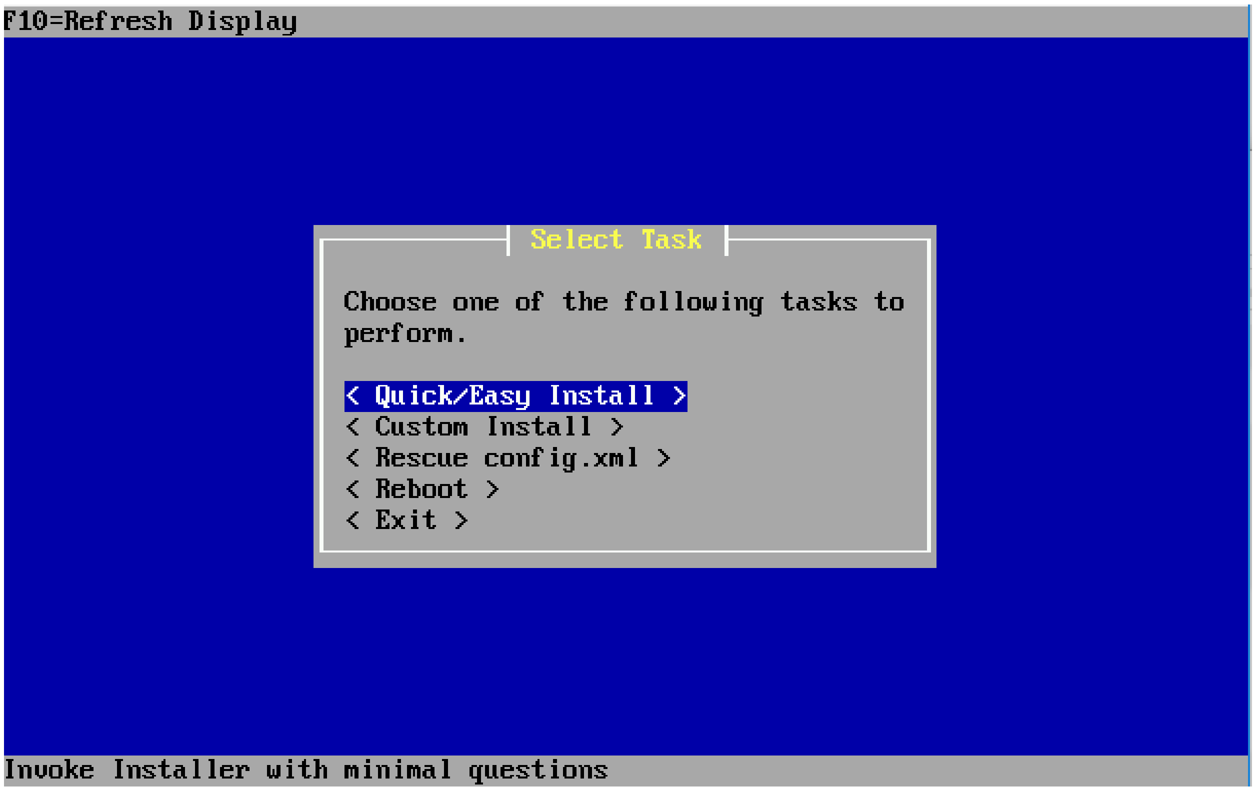
Back at the main menu, click on the “pfSense” entry to highlight it, then click the big green start arrow to boot up the VM. A console window should appear that will allow you to interact with the VM as though you had a keyboard, mouse, and monitor directly attached to it. By default, if you click on the window, the virtual machine will take control of the mouse and keyboard. To detach the mouse and keyboard from the VM console, you have to enter a special key or keyboard combination called a hotkey. In Windows and Linux, this is the right CTRL key, while on OS X, this is the left Command key.

Installing pfSense is very easy. Let the VM boot up, and the system should automatically run the installer. Adjust your video, screenmap, and keymap settings as necessary, then select “< Accept these Settings >”.



```
F10=Refresh Display

Configure Console

Your selected environment uses the
following console settings, shown in
parentheses. Select any that you wish
to change.

< Change Video Font (default) >
< Change Screenmap (default) >
< Change Keymap (default) >
< Accept these Settings >
```

On the next screen, select “< Quick/Easy Install >” and let pfSense do all the heavy lifting. The next screen will inform you that the install will erase the contents of the hard disk. Since our virtual disk is already empty, this doesn’t matter in the least. Select OK, and let the installer run.



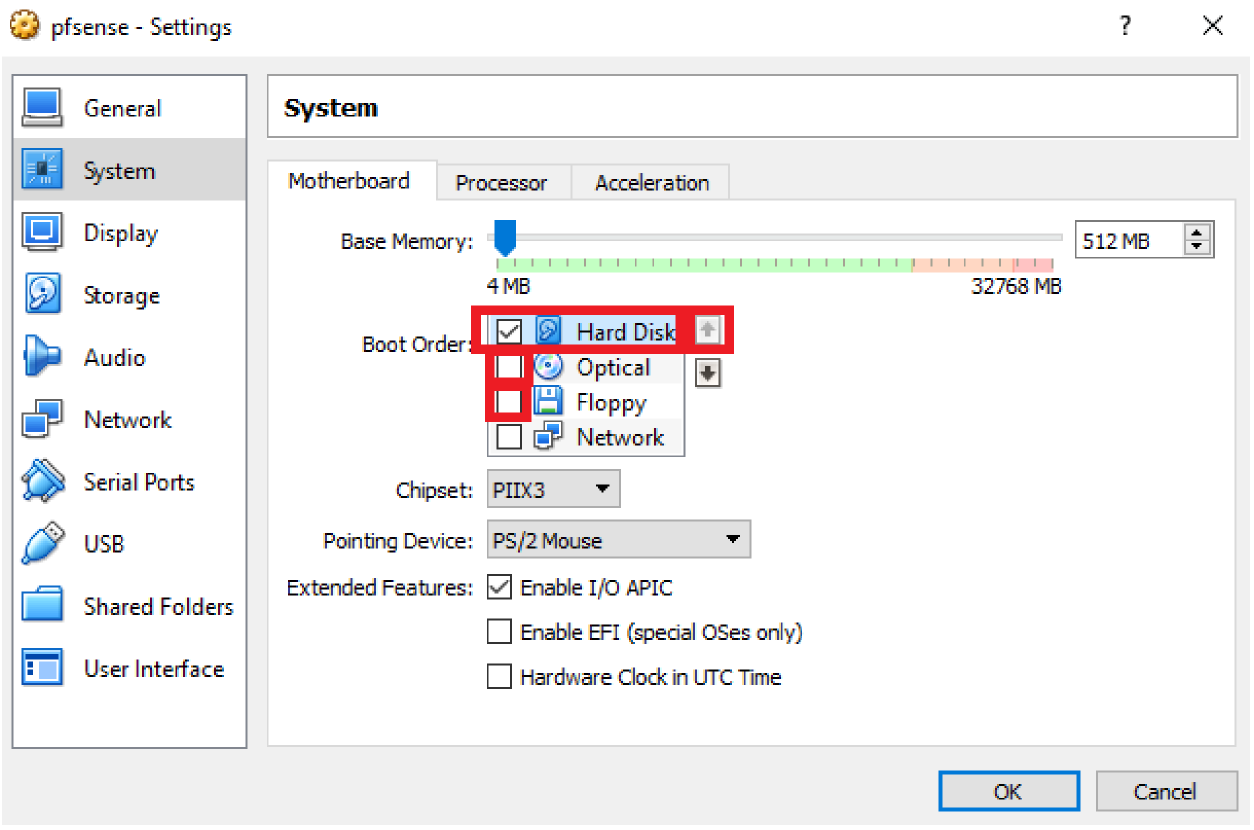
```
F10=Refresh Display

        Select Task

Choose one of the following tasks to
perform.

< Quick/Easy Install >
< Custom Install >
< Rescue config.xml >
< Reboot >
< Exit >

Invoke Installer with minimal questions
```

The installer will go through and install pfSense to the virtual hard disk. At a certain point you will be asked whether the standard kernel or the embedded one should be used. Make sure to select `< Standard Kernel >`. Finally, the installer informs you to reboot the machine to boot from the hard drive. The installation is done, however, we’re not going to reboot. Instead, shut the VM down. Click *Machine*, in the VM’s console window menu, and clicking *ACPI Shutdown*.

Final VM Settings

We have to modify a few more settings of the VM before we start configuring pfSense properly. On the VirtualBox main menu, right click on pfSense and go back to the “Settings…” menu. Select “System”, then ensure the the “Motherboard” tab is selected. We need to modify the “Boot Order” options.

First, uncheck the checkbox next to “Optical”, then make sure that “Hark Disk” is the only checked option remaining. Finally, highlight “Hard Disk” then click the up arrow to the right of the “Boot Order” list, until “Hard Disk” is at the top. This change ensures that the operating system installed on the hard drive is the first and ONLY bootable device for this VM.



Base Memory: 512 MB

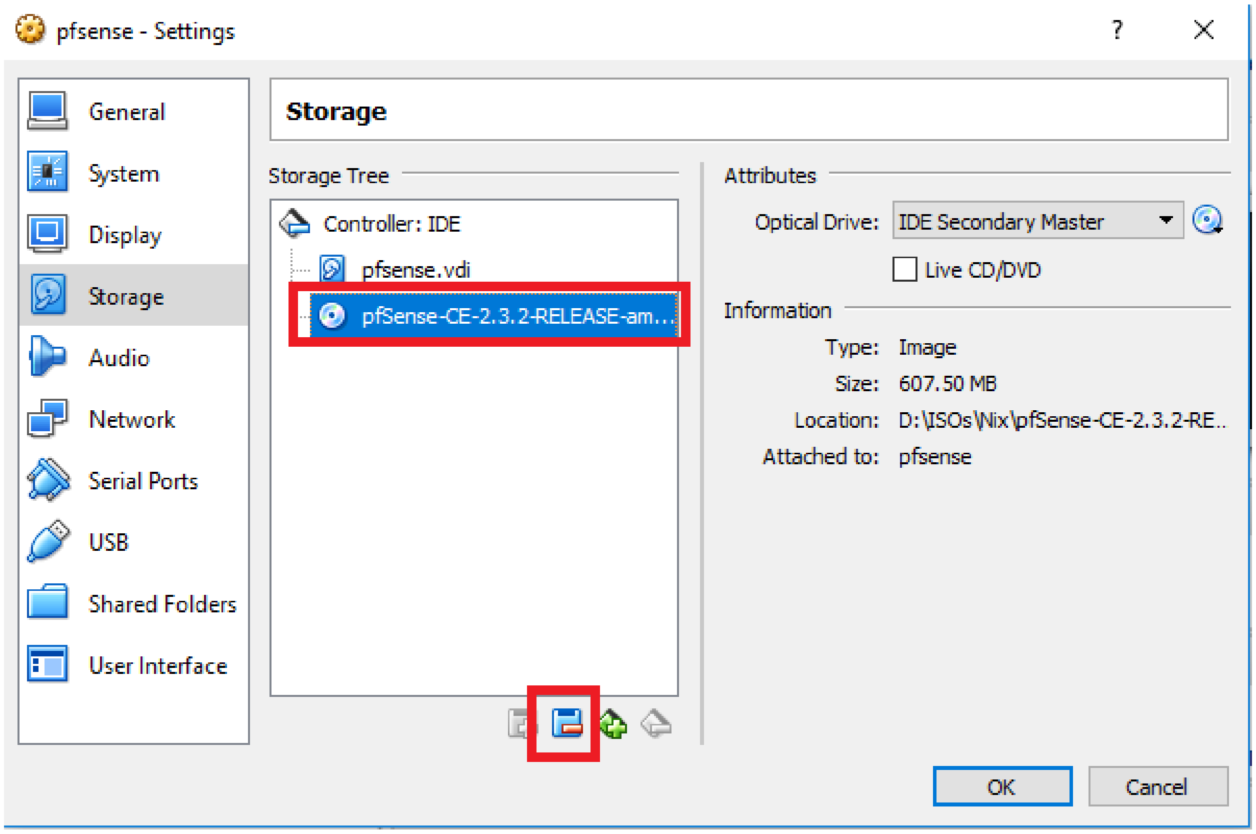
Boot Order:
- [✓] Hard Disk
- [ ] Optical
- [ ] Floppy
- [ ] Network

Chipset: PIIX3

Pointing Device: PS/2 Mouse

Extended Features:
- [✓] Enable I/O APIC
- [ ] Enable EFI (special OSes only)
- [ ] Hardware Clock in UTC Time

Next, click "Storage". Under the "Storage Tree" pane, click on the icon that looks like a CD, then on the bottom of the pane, click on the blue icon with a red minus sign on it. This will uninstall the virtual CD/DVD drive for this VM. This change ensures that the virtual hard drive is the only storage media available for this VM now. Click OK to exit the "Settings" menu.



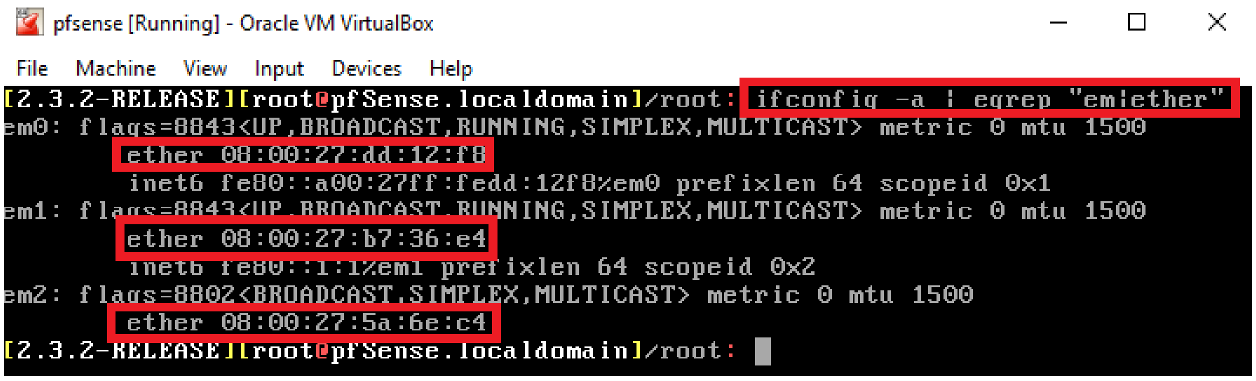
Controller: IDE
pfsense.vdi
pfSense-CE-2.3.2-RELEASE-am...

Optical Drive: IDE Secondary Master
Live CD/DVD

Type: Image
Size: 607.50 MB
Location: D:\ISOs\Nix\pfSense-CE-2.3.2-RE..
Attached to: pfsense

Network Configuration

On the VirtualBox main menu. Click on the “pfSense” VM to highlight it, then click the start arrow to boot the VM. Once the console appears, wait a moment for the VM to finish booting, and you should be greeted by the pfSense main menu. First things first, we have to correlate which interfaces in the VM correspond to adapter 1, adapter 2, and adapter 3 in the VirtualBox network settings. In the pfSense main menu, select option 8 to open up a shell. If you're not familiar with the command line interface in Linux/Unix, don’t worry; we won’t be here very long. Type the command `ifconfig -a | egrep “em|ether”`. What this command does is run `ifconfig -a`, a command that shows all the information for all of the network interfaces installed that the OS recognizes. The pipe symbol “`|`” is taking the output from the `ifconfig` command and “piping” it as input to the `egrep` command. We are telling `egrep` to process the output from `ifconfig` and ONLY show us lines that command “em” OR “ether”. If you did it correctly, the output should look something like this:

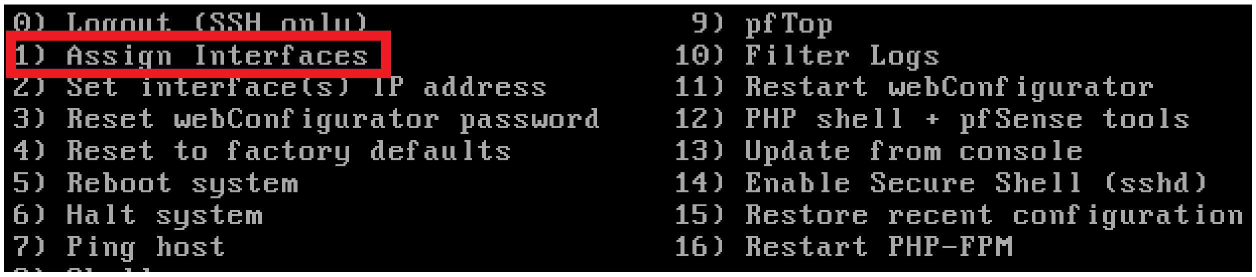

pfsense [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
[2.3.2-RELEASE][root@pfSense.localdomain]/root: ifconfig -a | egrep "em|ether"
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 08:00:27:dd:12:f8
        inet6 fe80::a00:27ff:fedd:12f8%em0 prefixlen 64 scopeid 0x1
em1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 08:00:27:b7:36:e4
        inet6 fe80::1:1%em1 prefixlen 64 scopeid 0x2
em2: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 08:00:27:5a:6e:c4
[2.3.2-RELEASE][root@pfSense.localdomain]/root:
```

In BSD/Linux terms, the MAC address is referred to as “ether”, shorthand for ethernet address. What we need to do is compare the ether addresses displayed here to figure out which interface in the VM (em0, em1, and em2) maps to which adapter (adapter 1, 2, and 3) in VirtualBox. This is why I had you document the MAC addresses for adapter 1, 2, and 3 earlier when we created the VM; We need to know which em interface corresponds to which adapter number in the VirtualBox network settings so we can make sure the interfaces in the VM are assigned and configured correctly.

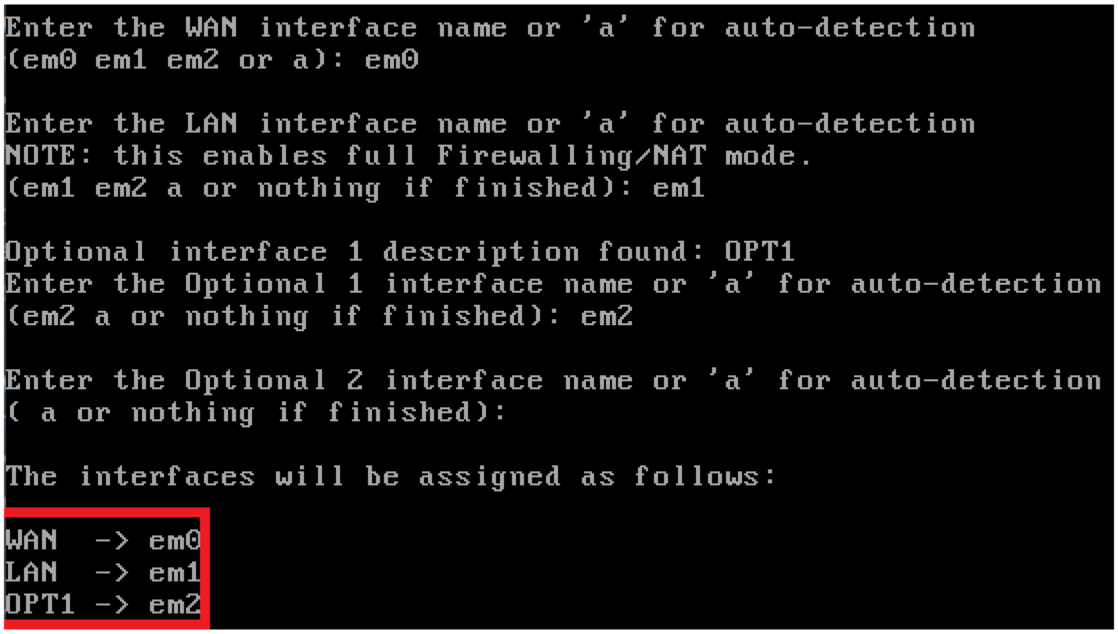
We need to know which interface in pfSense corresponds to which adapter so we can designate the WAN (bridged network), LAN (Host-Only Network) and OPT1 (Internal Network) properly in pfSense. In my case, the ether address for em0 is 00:00:27:dd:12:f8. This corresponds to the MAC address assigned to adapter 1 in VirtualBox (000027dd12f8-- exactly the same, without the colon separators), em1 corresponds to adapter 2, and em2 corresponds to adapter 3. Once you have figured out which interface corresponds to which VirtualBox adapter, type “exit” to leave the shell and go back to the pfSense main menu.

Select option 1, “Assign Interfaces” to run the interface assignment wizard. To manage its network services and firewall policies, pfSense uses internal network interface descriptors (WAN, LAN, OPT1, etc.). Assigning the interfaces defines which of them (em0, em1, em2, etc.) corresponds to which of pfSense's internal descriptors.



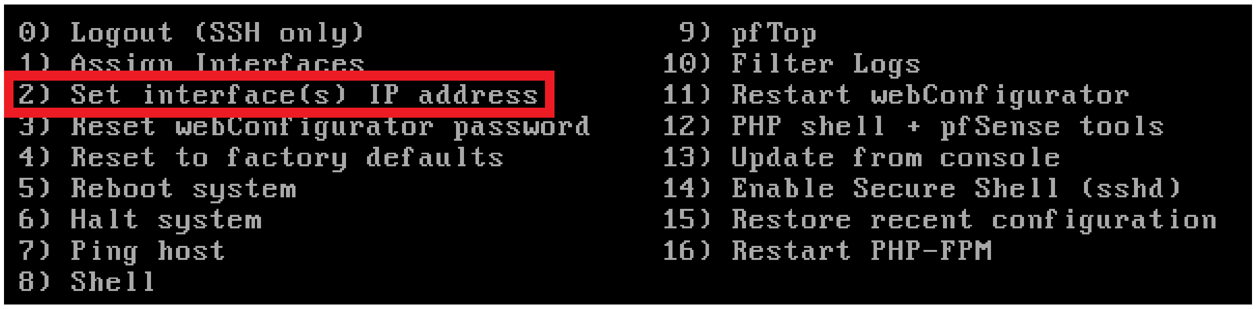
```
 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
```

The WAN interface needs to be assigned to the bridged network adapter (Adapter 1, the Bridged network). The LAN interface needs to be assigned to our Host-Only adapter (Adapter 2, the Management network), and finally, OPT1 needs to be assigned to our internal network adapter (Adapter 3, the IPS 1 network). In my case, em0 became the WAN interface, em1 the LAN interface, and em2 was mapped to the OPT1 interface. After confirming that the settings are correct, you should automatically be returned to the pfSense main menu.



```
Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): em0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): em1

Optional interface 1 description found: OPT1
Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished): em2

Enter the Optional 2 interface name or 'a' for auto-detection
( a or nothing if finished):

The interfaces will be assigned as follows:

WAN  -> em0
LAN  -> em1
OPT1 -> em2
```

Now that we have assigned the network interfaces, we have to configure IP addresses for these interfaces. In most cases, the WAN interface will automatically get an IP address from the device on your physical network that provides DHCP services. If this is not happening, you may have to troubleshoot your physical network. This is beyond the scope of our guide. As for the LAN and OPT1 networks, we have to manually set the IP address, subnet mask, and DHCP scopes for these networks. Select option 2 in the pfSense main menu to get started.



```
 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell
```

The configuration wizard starts by asking for the interface the settings should be changed for. We have to go through the process two times, for the LAN and the OPT1 interface individually. Here are the settings I recommend:

LAN IP: **172.16.1.1**
LAN Subnet bit count: **24**
LAN upstream gateway address: **<empty>**
LAN IPv6 address: **<empty>**
Do you want to enable the DHCP server on LAN? **y**
LAN DHCP start address: **172.16.1.10**
LAN DHCP end address: **172.16.1.254**
Do you want to revert to HTTP as the webConfigurator protocol? **N**

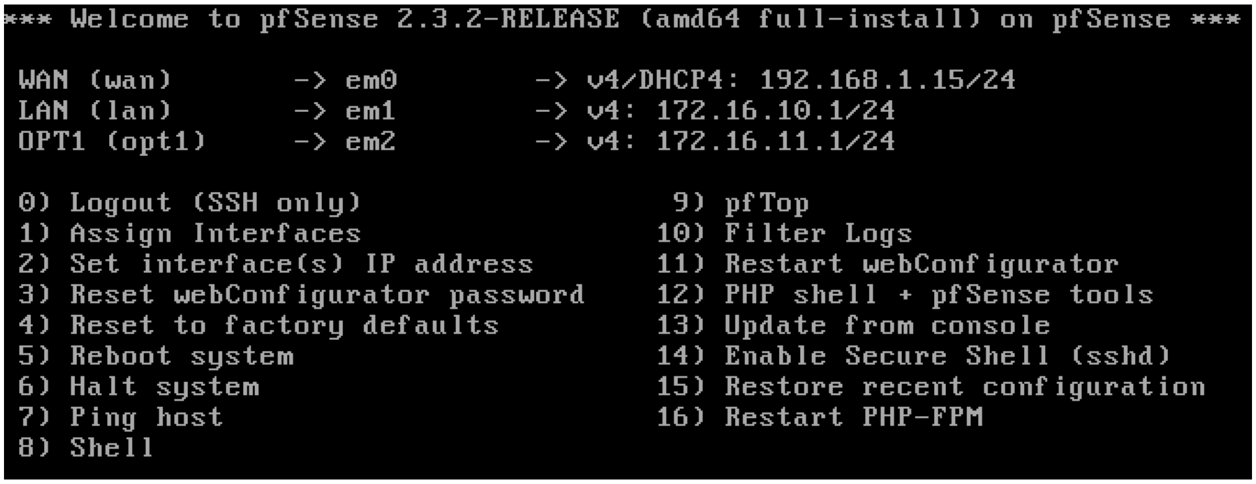
OPT1 IP: **172.16.2.1**
OPT1 Subnet bit count: **24**
OPT1 upstream gateway address: **<empty>**
OPT1 IPv6 address: **<empty>**
Do you want to enable the DHCP server on OPT1? **y**
OPT1 DHCP start address: **172.16.2.10**
OPT1 DHCP end address: **172.16.2.254**
Do you want to revert to HTTP as the webConfigurator protocol? **n**

The reason we choose 1.10 and 2.10 as the DHCP start addresses is to reserve a few IP addresses for static DHCP allocations. Static DHCP allocations allow you to configure a DHCP server to always serve the same IP address when requested from a particular MAC address. We will talk about this a little bit more later. For now, we are done mucking around in the pfSense CLI. From here on out, we will be using the webConfigurator to manage and configure our pfSense VM.

Note: If you are using 172.16.1.0/24 or the 172.16.2.0/24 network ranges in your physical network, choose another network range to assign to the LAN and OPT1 interfaces to avoid network conflicts.

Note: When you are configuring the LAN and OPT1 interface IP addresses, the configuration script will ask "Do you want to revert to HTTP as the webConfigurator protocol?" **Always say no to this.** This allows the web UI to accept HTTP logins, and makes your firewall credentials vulnerable to sniffing over the network.

Your pfSense main menu should look something like this when you're all done:



```
*** Welcome to pfSense 2.3.2-RELEASE (amd64 full-install) on pfSense ***

 WAN (wan)       -> em0        -> v4/DHCP4: 192.168.1.15/24
 LAN (lan)       -> em1        -> v4: 172.16.10.1/24
 OPT1 (opt1)     -> em2        -> v4: 172.16.11.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell
```

Before moving on to setting up pfSense from the webConfigurator, you must configure the Host-Only network card attached to the hypervisor host. If you are using Windows as your host, and If you haven't already, you'll want to visit the "Unbinding Network Protocols on Windows Virtual Adapters" at a minimum. This will guide you through configuring the VirtualBox Host-Only Ethernet Adapter with an IP address as well as unbinding network protocols on this adapter to increase the security of your Windows hypervisor host. You may also want to consider "Using Windows Firewall to Limit Exposure of Windows Hypervisor Hosts" to further enhance the security of the Windows host when interacting with your lab VMs.

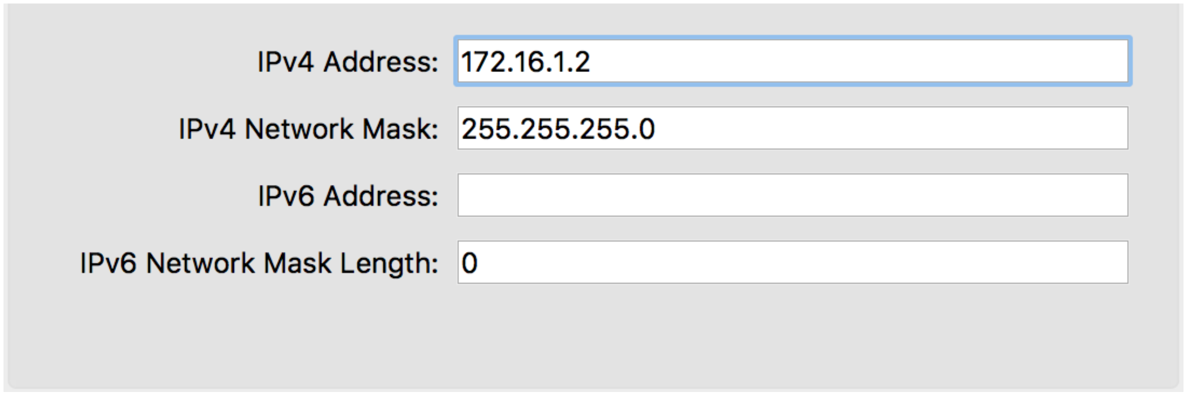
Linux and OS X users can configure the IP address of their host-only network card by using the command utilities `ifconfig` or `ip addr add`. Try one of the following commands to set the IP address of the vboxnet0 interface:

```
ifconfig vboxnet0 172.16.1.2 netmask 255.255.255.0
ip addr add 172.16.1.2/24 dev vboxnet0
```

Please note that you will need root permissions to run these commands, and that these settings will NOT persist between reboots of your hypervisor host system. Configuring IP address persistence for your host OS is beyond the scope of this guide.

If you don't want to mess around with `ifconfig` or the `ip` command you can use the VirtualBox "Preferences" menu to set the IP address of vboxnet0 as well. Navigate to "Network", select the "Host-Only Networks" tab, click vboxnet0 to highlight it, then click the screwdriver icon. A small window will pop up. Select the "Adapter" tab if it isn't already selected, then fill out the following:

IPv4 Address: **172.16.1.2**
IPv4 Network Mask: **255.255.255.0**



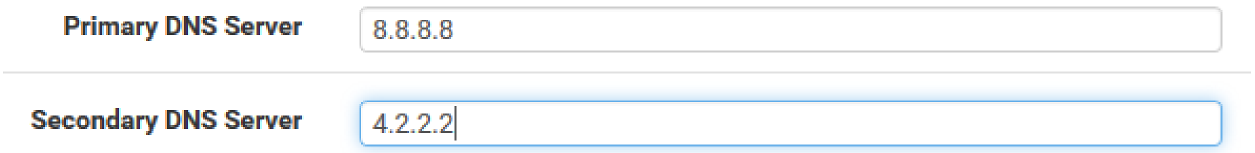
IPv4 Address:	172.16.1.2
IPv4 Network Mask:	255.255.255.0
IPv6 Address:	
IPv6 Network Mask Length:	0

Leave the rest of the fields alone, then click OK to exit the configuration menu, then click OK to exit preferences menu.

webConfigurator - Initial Setup

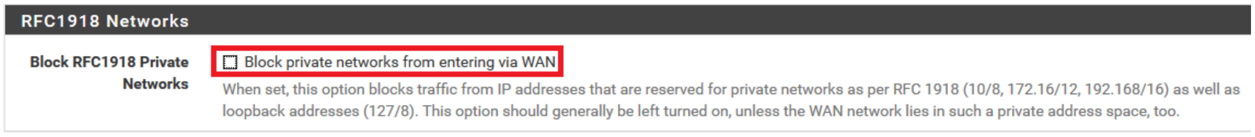
On your VirtualBox host OS, open your favorite web browser (I prefer Firefox - https://www.mozilla.org/en-US/firefox/new/) and navigate to https://172.16.1.1 (or the IP address you assigned to the LAN interface of your pfSense system). **The default credentials for access to the web interface are admin/pfsense.** If this is your first time logging in, pfSense takes you through a nice little setup wizard. I'll highlight some of the important things to take note of, and/or change as necessary:

Set the primary and secondary DNS servers you plan on using. I typically use 8.8.8.8 (Google public DNS) and 4.2.2.2 (Level 3 public DNS). If you're using this lab at work, your workplace may have restrictions and you may have to use their DNS servers instead.



Primary DNS Server	8.8.8.8
Secondary DNS Server	4.2.2.2

Uncheck the checkbox next to "Block private networks from entering via WAN" rule. Uncheck the checkbox next to "Block Bogon Networks" on this page as well.



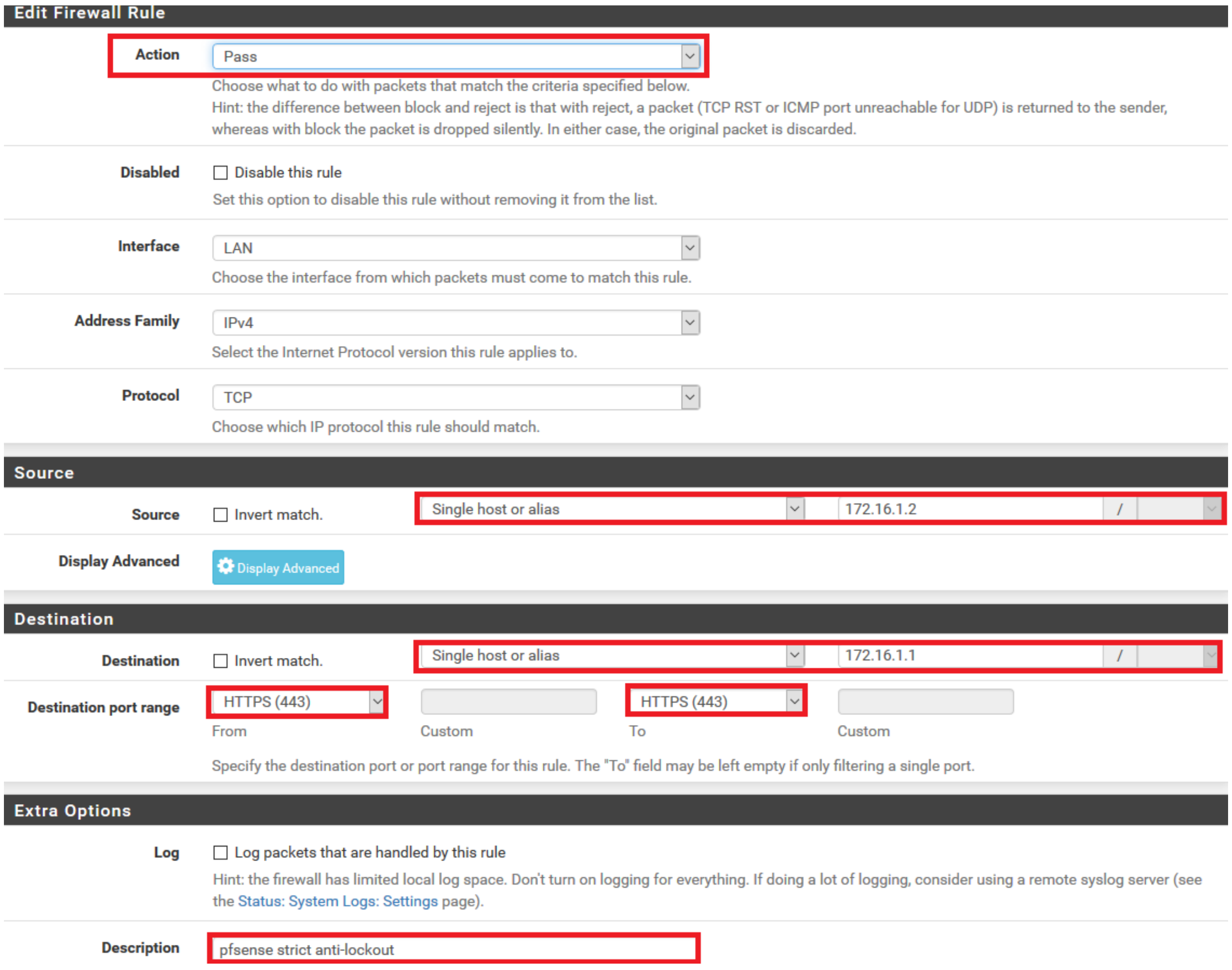
RFC1918 Networks

Block RFC1918 Private Networks: ☐ Block private networks from entering via WAN

When set, this option blocks traffic from IP addresses that are reserved for private networks as per RFC 1918 (10/8, 172.16/12, 192.168/16) as well as loopback addresses (127/8). This option should generally be left turned on, unless the WAN network lies in such a private address space, too.

The other default settings on the first time setup wizard should be fine. Please note that you will be asked to change the password for the admin account as a part of the setup wizard. Document the password, and keep it somewhere safe.

Our next order of business is to restrict access to the web interface of the firewall to the machine with the IP address 172.16.1.2. On the menu bar at the top of the page, select *Firewall > Rules*. Click on *LAN* to modify the firewall policy for the *LAN* interface. Click the *Add* button with the arrow facing up. This will add the firewall rule to the top of the firewall policy to where it will evaluated first. The *Edit Firewall Rule* page is fairly straightforward. I've highlighted the options to be aware of.



Edit Firewall Rule

Action	Pass Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.
Disabled	☐ Disable this rule Set this option to disable this rule without removing it from the list.
Interface	LAN Choose the interface from which packets must come to match this rule.
Address Family	IPv4 Select the Internet Protocol version this rule applies to.
Protocol	TCP Choose which IP protocol this rule should match.

Source

Source	☐ Invert match.	Single host or alias	172.16.1.2 /
Display Advanced	Display Advanced		

Destination

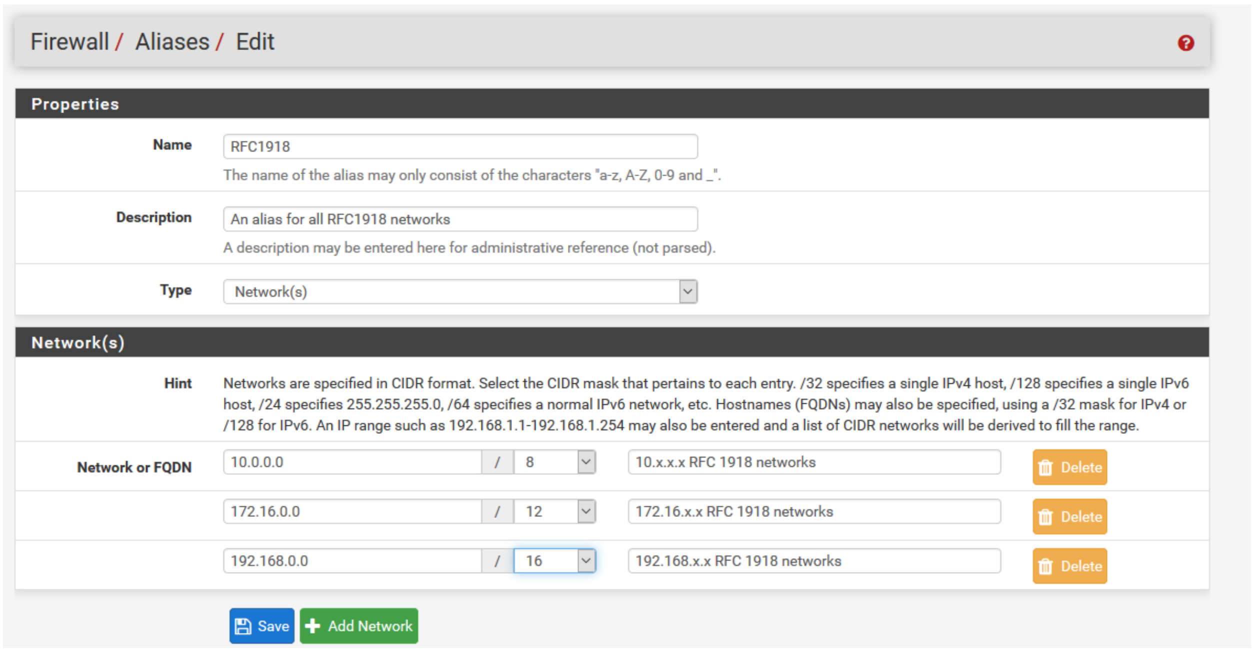
Destination	☐ Invert match.	Single host or alias	172.16.1.1 /	
Destination port range	HTTPS (443) From	Custom	HTTPS (443) To	Custom

Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.

Extra Options

Log	☐ Log packets that are handled by this rule Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the Status: System Logs: Settings page).
Description	pfsense strict anti-lockout

When you are done, click the Save icon at the bottom of the page. This will take you back to the previous page. A yellow dialogue box with a green button called “Apply Changes” will appear. Click this button to apply this new firewall rule. Next, we want to an alias. Navigate to *Firewall > Aliases*. On the Firewall Aliases page, click on *IP*, then click *Add*. Create an alias with the following settings:


Firewall / Aliases / Edit

Properties

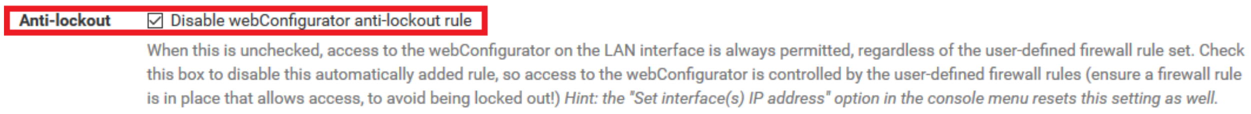
Name	RFC1918
	The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".
Description	An alias for all RFC1918 networks
	A description may be entered here for administrative reference (not parsed).
Type	Network(s)

Network(s)

Hint: Networks are specified in CIDR format. Select the CIDR mask that pertains to each entry. /32 specifies a single IPv4 host, /128 specifies a single IPv6 host, /24 specifies 255.255.255.0, /64 specifies a normal IPv6 network, etc. Hostnames (FQDNs) may also be specified, using a /32 mask for IPv4 or /128 for IPv6. An IP range such as 192.168.1.1-192.168.1.254 may also be entered and a list of CIDR networks will be derived to fill the range.

Network or FQDN	10.0.0.0	/ 8	10.x.x.x RFC 1918 networks	Delete
	172.16.0.0	/ 12	172.16.x.x RFC 1918 networks	Delete
	192.168.0.0	/ 16	192.168.x.x RFC 1918 networks	Delete

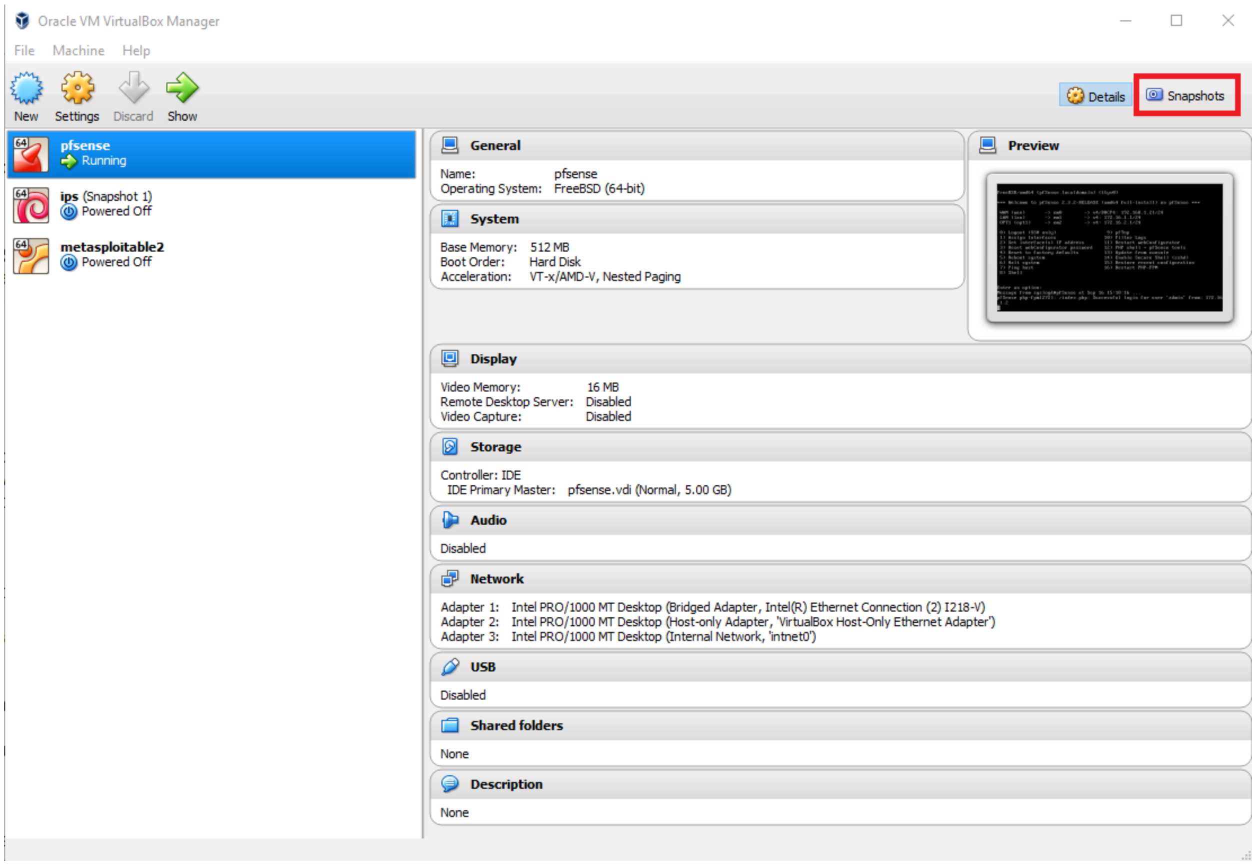
Click Save to be brought back to the previous page, then click *Apply Changes*. We just created an alias for RFC1918 networks (local networks that are not routable through the public internet) This will come in handy later for creating firewall rules around these networks. Our final action for now is to navigate to *System > Advanced*, and click to fill in the checkbox next to the option *Disable webConfigurator anti-lockout rule*, and click *Save* on the bottom of the page. **Be aware that if you did not create the firewall rule above, this <u>will</u> lock you out of the webConfigurator as soon as you click *Save*.**


Anti-lockout ☑ Disable webConfigurator anti-lockout rule

When this is unchecked, access to the webConfigurator on the LAN interface is always permitted, regardless of the user-defined firewall rule set. Check this box to disable this automatically added rule, so access to the webConfigurator is controlled by the user-defined firewall rules (ensure a firewall rule is in place that allows access, to avoid being locked out!) *Hint: the "Set interface(s) IP address" option in the console menu resets this setting as well.*

Take a Snapshot

Snapshotting is a VERY important concept with VMs. It lets you restore your virtual machines to its state in the past when the snapshot was taken.This allows you to undo misconfigurations, solve problems or potential issues that might affect the VM, in relatively easy fashion. We're going to take a snapshot of pfSense in its current state. In the VirtualBox main menu, click on the icon that looks like a little camera in the upper right corner of the menu, labeled "Snapshots".



General
Name: pfsense
Operating System: FreeBSD (64-bit)

System
Base Memory: 512 MB
Boot Order: Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging

Display
Video Memory: 16 MB
Remote Desktop Server: Disabled
Video Capture: Disabled

Storage
Controller: IDE
IDE Primary Master: pfsense.vdi (Normal, 5.00 GB)

Audio
Disabled

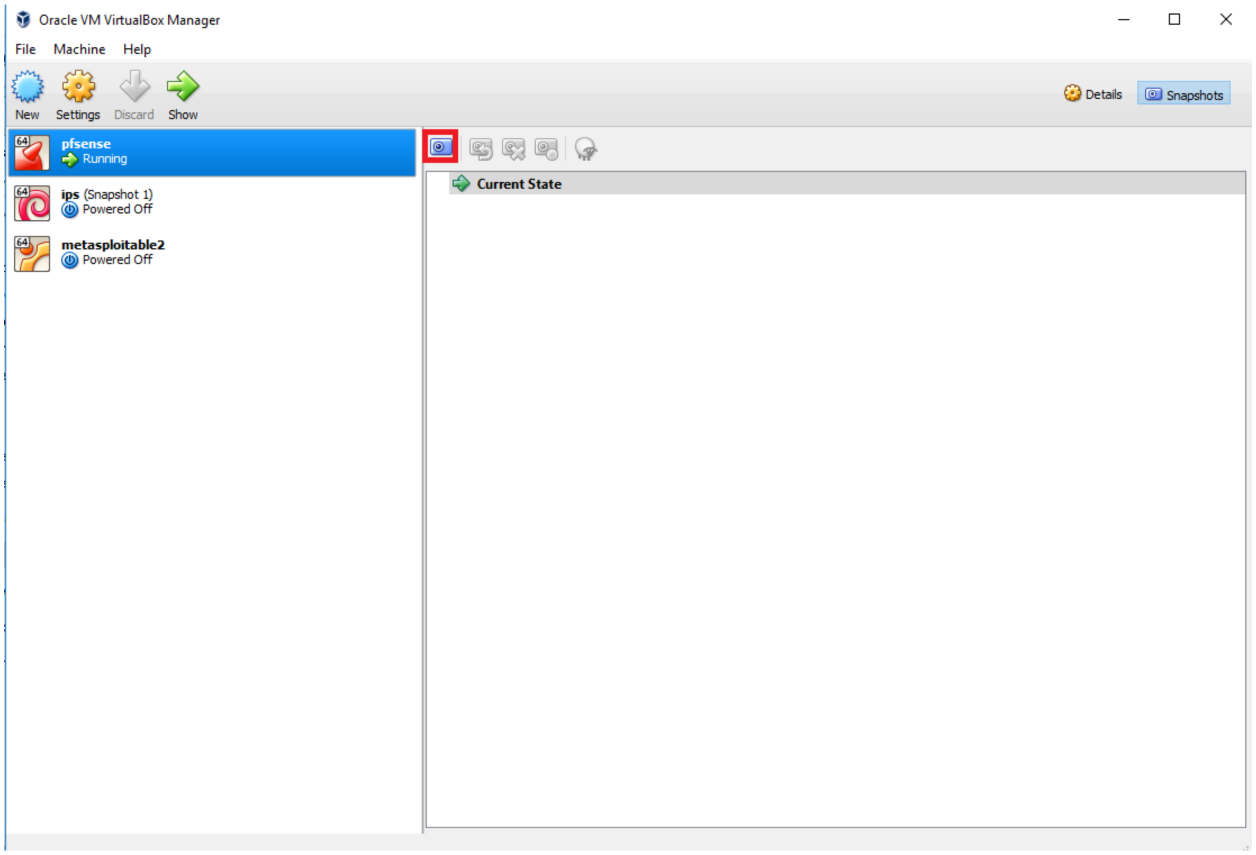
Network
Adapter 1: Intel PRO/1000 MT Desktop (Bridged Adapter, Intel(R) Ethernet Connection (2) I218-V)
Adapter 2: Intel PRO/1000 MT Desktop (Host-only Adapter, 'VirtualBox Host-Only Ethernet Adapter')
Adapter 3: Intel PRO/1000 MT Desktop (Internal Network, 'intnet0')

USB
Disabled

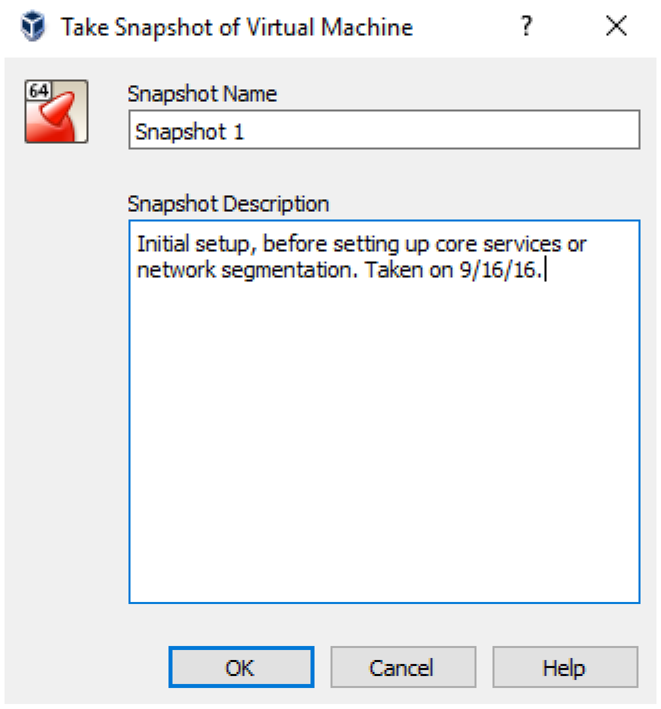
Shared folders
None

Description
None

The view changes to display the snapshot manager. You will find another camera icon located in the upper left corner of right pane. Click on this icon to take a snapshot of the VM as it is right now.



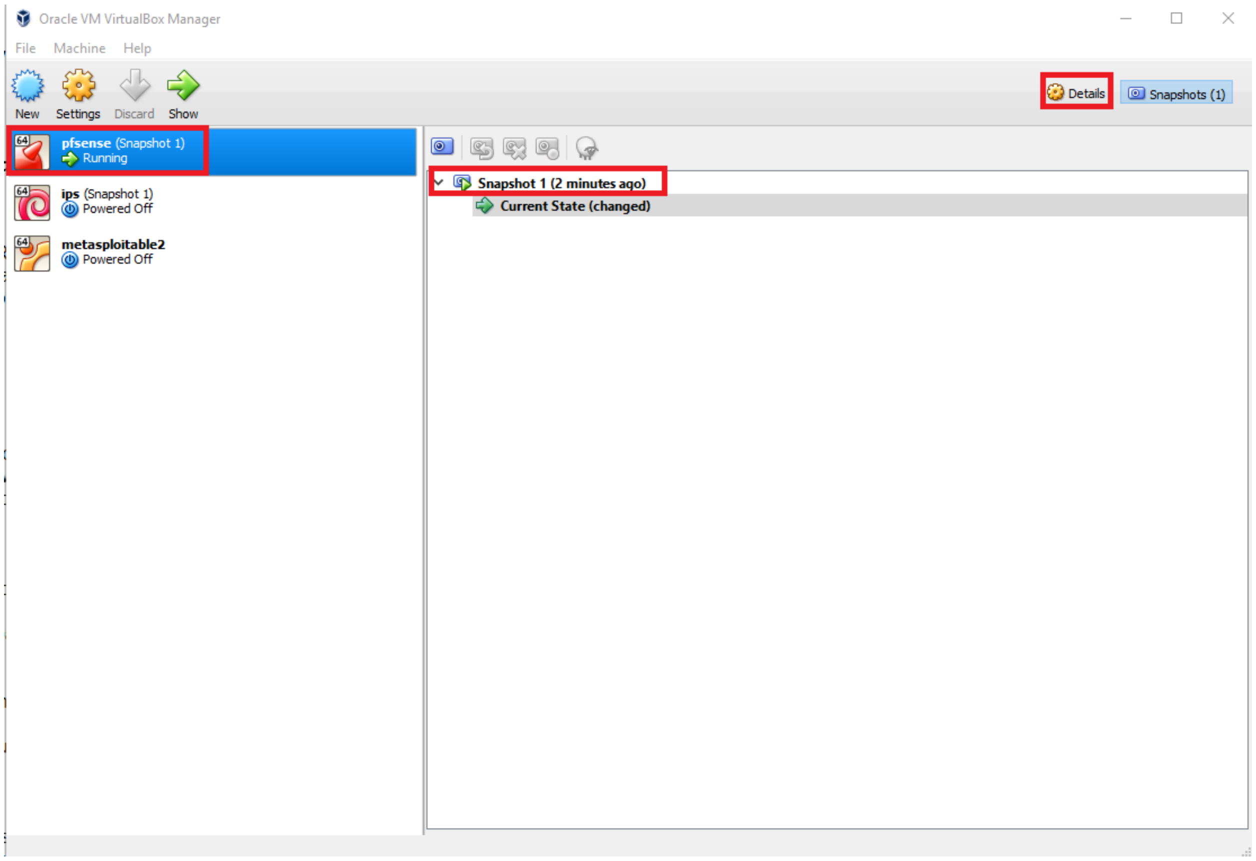
Clicking the camera icon brings up a window entitled “Take Snapshot of Virtual Machine”. It asks you to name the snapshot and give it a description. If you keep multiple snapshots make sure you keep good descriptions that include WHEN the snapshot was taken, and WHAT STATE the VM is in when you took the snapshot. Once you've named it and added a good, valid description, click OK to generate the snapshot.


Take Snapshot of Virtual Machine

Snapshot Name: Snapshot 1

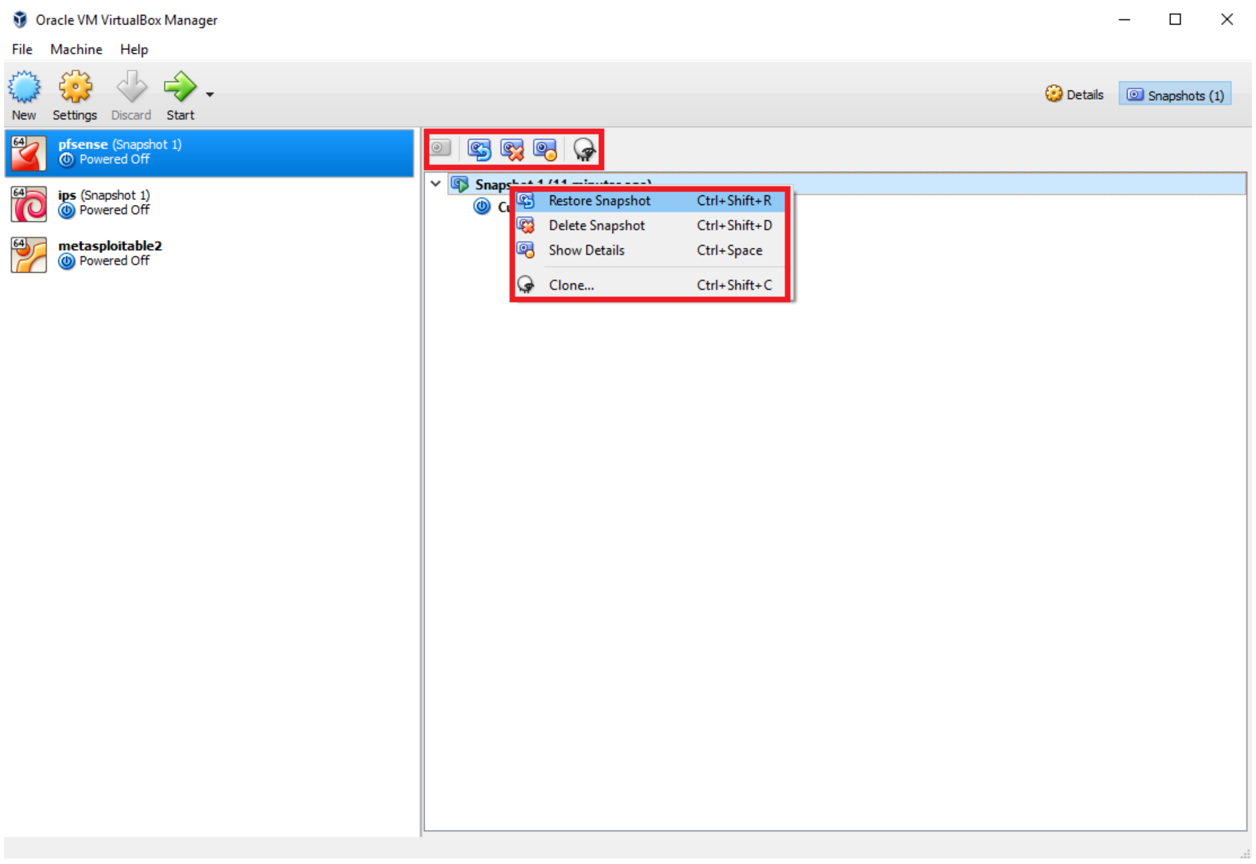
Snapshot Description: Initial setup, before setting up core services or network segmentation. Taken on 9/16/16.

After a few seconds, the snapshot you just created shows up in the snapshot manager for the selected VM, and a little indicator appears next to the name of the VM, displaying the number snapshots available.Remember to delete old snapshots you are NOT planning to use any more, in order to conserve disk space. When you are done, you can click the cog labeled “Details” to return the VirtualBox main menu.



If you ever have to restore a VM from a snapshot, first, shutdown/power off the VM, then navigate to the snapshot manager. Right click on the snapshot you wish to revert to. Choose “Show Details” to view the description you (hopefully) provided for the snapshot, or select “Restore Snapshot” to start the snapshot restore process. This will take a moment or two, and voila, your VM is restored to its previous state.

Note that the icons above the snapshot pane perform the same function as those presented in the menu when right clicking the snapshot: the camera with the U-turn arrow performs a snapshot restore, the camera with the red X deletes the highlighted snapshot, and the camera with the circle shows you details on the highlighted snapshot. The sheep icon allows you to make an exact copy (also known as a clone) of the VM in its current state or at any state it was in, during any of the snapshots.



pfSense Summary

At the end of all this, you should have a pfSense VM with the following settings:

- 512MB of RAM
- 5GB of disk
- 1 Virtual CPU
- DVD drive disabled
- Audio disabled
- USB controller disabled
- 3 Network interfaces:
    - 1 Bridged (Bridge vswitch/WAN interface)
    - 1 Host-Only (Management vswitch/LAN interface)
    - 1 Internal Network (IPS vswitch1/OPT1 interface)

pfSense should be configured as followed:

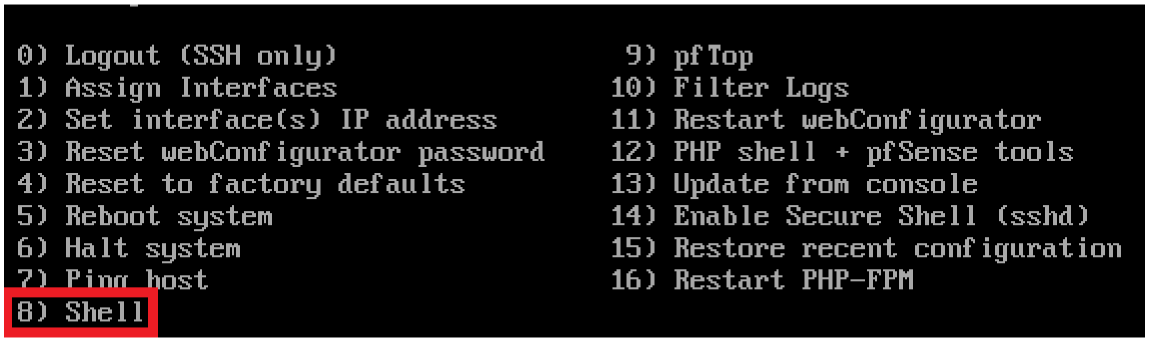
- The WAN interface should be the VirtualBox network adapter connected to the Bridged Adapter. This interface should automatically get an IP address via DHCP. If this isn't happening, get some help with whoever administers your local network, or start troubleshooting. **Alternatively, configure a static IP address**
- The LAN interface should have an IP address of 172.16.1.1.
- The LAN network should be 172.16.1.0 with a /24 (255.255.255.0) subnet mask
- DHCP should be configured for this network with a scope of 172.16.1.10-172.16.1.254
    - 172.16.1.3-172.16.1.9 are available for setting static DHCP mappings
- The OPT1 interface should have an IP address of 172.16.2.1
- The OPT1 network should be 172.16.2.0 with a /24 (255.255.255.0) subnet mask
- DHCP should be configured for this network with a scope of 172.16.2.10-172.16.2.254
    - 172.16.2.2-172.16.2.9 are available for static DHCP mappings.
- 172.16.1.2 should have a firewall rule configured to allow it access to the firewall over HTTPS; this is the anti-lockout firewall rule.
- You should have an alias for RFC1918 networks configured for use with the firewall policy.
- You should have at least one good snapshot you can revert to if you need to redo a step or somehow got lost/confused.

What's Next?

Now that you have an operational pfSense VM, your next steps are to read the pfSense Firewall Rules and Network Services Guide. Make sure the firewall policies match those illustrated in the segmentation guide, and make sure that the pfSense VM is hosting all the services mentioned in the Core Network Services guide to make your firewall fully functional and ready to handle your lab network.

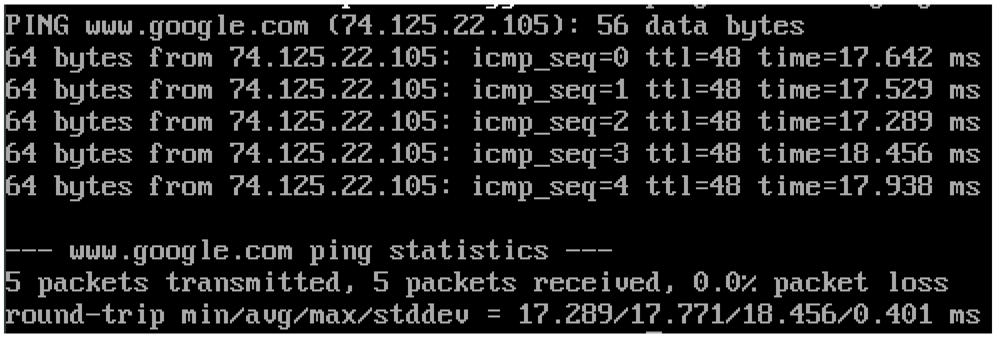
Final Connectivity Checks and Troubleshooting

Once you have finished setting up your pfSense VM according to the instructions laid out, Open a console session to the pfSense VM, and attach your mouse and keyboard by clicking on the console. If the screen is blank, hit the *Enter* key to bring up the pfSense main menu. Select option *8) Shell* to start a shell session on pfSense.



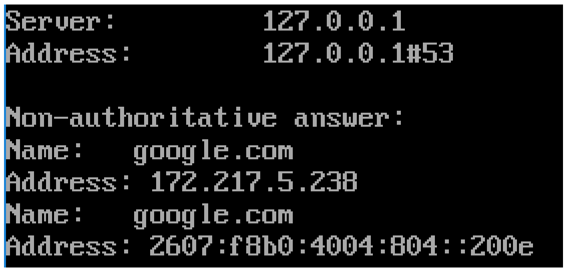
```
0) Logout (SSH only)                  9) pfTop
1) Assign Interfaces                 10) Filter Logs
2) Set interface(s) IP address       11) Restart webConfigurator
3) Reset webConfigurator password    12) PHP shell + pfSense tools
4) Reset to factory defaults         13) Update from console
5) Reboot system                     14) Enable Secure Shell (sshd)
6) Halt system                       15) Restore recent configuration
7) Ping host                         16) Restart PHP-FPM
8) Shell
```

The first, run the command `ping -c 5 www.google.com`



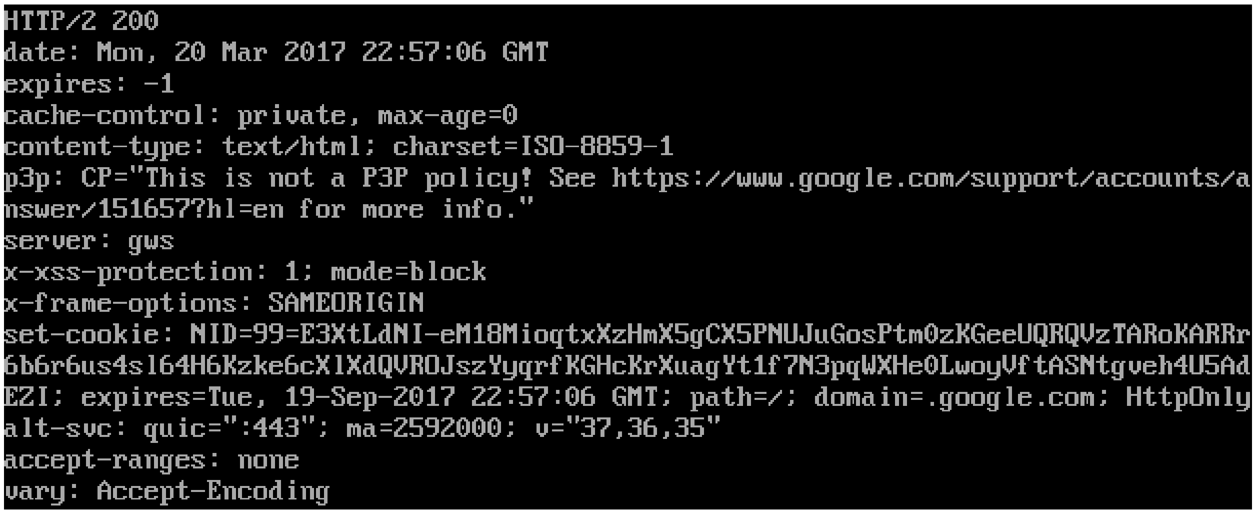
```
PING www.google.com (74.125.22.105): 56 data bytes
64 bytes from 74.125.22.105: icmp_seq=0 ttl=48 time=17.642 ms
64 bytes from 74.125.22.105: icmp_seq=1 ttl=48 time=17.529 ms
64 bytes from 74.125.22.105: icmp_seq=2 ttl=48 time=17.289 ms
64 bytes from 74.125.22.105: icmp_seq=3 ttl=48 time=18.456 ms
64 bytes from 74.125.22.105: icmp_seq=4 ttl=48 time=17.938 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 17.289/17.771/18.456/0.401 ms
```

The output from the `ping` command should look similar to the output above. Next, run the command `nslookup google.com`



```
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.5.238
Name:   google.com
Address: 2607:f8b0:4004:804::200e
```

Again, you should have output similar to the illustration above. Finally, run `curl -I https://www.google.com`



```
HTTP/2 200
date: Mon, 20 Mar 2017 22:57:06 GMT
expires: -1
cache-control: private, max-age=0
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See https://www.google.com/support/accounts/a
nswer/151657?hl=en for more info."
server: gws
x-xss-protection: 1; mode=block
x-frame-options: SAMEORIGIN
set-cookie: NID=99=E3XtLdNI-eM18MioqtxXzHmX5gCX5PNUJuGosPtm0zKGeeUQRQVzTARoKARRr
6b6r6us4sl64H6Kzke6cXlXdQVROJszYyqrfKGHcKrXuagYt1f7N3pqWXHe0LwoyVftASNtgveh4U5Ad
EZI; expires=Tue, 19-Sep-2017 22:57:06 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: quic=":443"; ma=2592000; v="37,36,35"
accept-ranges: none
vary: Accept-Encoding
```

The output of your `curl` command should be similar to the screen capture above. If so, you may type `exit` to leave the shell, and close your connection to the pfSense console. If you had any problems with any of the commands above, you've got troubleshooting to do. When it comes to troubleshooting, start at layer 1 of the OSI model (Physical) and work your way up. Here are some questions to help the troubleshooting process:

- Which of the commands above failed? Did they all fail?
- Have you checked your physical cabling or wireless network connectivity?
- Does your company/LAN have MAC filtering/port security on? You might need to modify switch settings for your drop, or talk to your network administrator.
- Have you checked your upstream gateway/SOHO router? Trying pinging the gateway's IP address to see if your pfSense VM can reach the gateway. Try running the `traceroute` command to google.com to see if and where your ping command is failing.

- Do you have an upstream firewall in place? Is 443/TCP allowed outbound? Is 53/UDP allowed outbound? Is ICMP allowed?
- Did you input the commands correctly? Try entering them again. No, I’m not patronizing you.

Once you have verified that your firewall’s connectivity, policies, and services are properly configured, **SNAPSHOT THE VM <u>BEFORE</u>** moving on and trying your hand at setting up the remaining VMs.

Your turn

The pfSense virtual machine required a lot of custom configuration, both in VirtualBox as well as in the VM itself once the OS had been installed. Now that we have done that together, I'm going to have you create the rest of the virtual machines yourself -- with the exception of the Metasploitable 2 VM since it's a special case for reasons you will see momentarily. Below are a set of spec sheets to help you create the remaining virtual machines for the lab on your own. Think of them as checklists you should be able to run through unaccompanied.

All of our Linux-based virtual machines are built on Debian Linux based distros (both, Ubuntu and Kali Linux are derived from Debian), so setup and configuration of all of our lab VMs should be very similar.

Kali Linux VM

Kali Linux is a popular penetration testing distro. Popular because hackers and script kiddies are lazy, and practically everything you need for performing a penetration test or joining anonymous is installed by default. Kali is going to be our loud, obnoxious attacker that we're going to use as a noise generator for the express purpose of generating events on our IPS VM. I want you to perform the following tasks:

- Download Kali Linux 64-bit here: https://www.kali.org/downloads/
- Create a VM with the following settings:
    - Name the VM "Kali"
    - Choose "Linux" as the type
    - Choose "Debian (64-bit)" as the version
    - 4GB of RAM
    - 80GB fixed size VDI
    - 1-2 virtual CPUs (virtual CPUs can be adjusted under VM Settings > System > Processor tab once the VM has been created)
- Once the VM is created, adjust the following settings:
    - Under the System settings, ensure that the VM is using a PS/2 Mouse
    - Under Storage, point the virtual cd/dvd drive to where you downloaded the Kali Linux ISO, so the VM has a bootable drive, and we have an actual OS to install
    - Disable Audio
    - Disable the USB controller
    - Under Network settings, make sure that Adapter 1 (and only Adapter 1) is enabled and that it is attached to "Internal Network", and that the internal network is named "intnet" (if hosting Virtualbox on Windows) or "intnet0" (if hosting

VirtualBox on a non-Windows OS). The internal network name should be the EXACT same as the internal network name for Adapter 3 on the “pfSense” VM

        - **Document the MAC address for Adapter 1**, under its advanced settings.

- Install Kali Linux
    - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.2.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy
- After the installation is completed, power down the VM and adjust the following settings:
    - Remove the virtual optical drive under the Storage Tree in Storage settings
    - Under system settings, disable the Floppy and Optical drives, and make the Hard Disk the first device the VM boots from
- Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the OPT1 interface DHCP for the MAC address of Adapter 1 for the Kali VM; assign it the IP address 172.16.2.2, making sure to enter a description that tells you which VM this static mapping belongs to
- Power on the virtual machine and do the following:
    - Verify that the static mapping for the VM’s IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter’s name is in the VM) was given the IP address you configured its static mapping for in pfSense
        - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
    - Verify the virtual machine can reach the internet
        - In a terminal/shell run the command `curl -I https://www.google.com`
            - This will confirm DNS can resolve domain names, and that the VM can reach the internet
            - If the command appears to hang or does not return output, use the “Final Connectivity Checks and Troubleshooting” guide we used for testing connectivity with pfSense
                - **Note:** The `ping` command, in this case, will work from this virtual machine, due to the unique firewall policy on the OPT1 network.
    - Update/patch the VM
        - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
            - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
    - Create a snapshot for the VM

Now you should have a functional Kali Linux VM. At this point, the virtual machine should be fully operational, and you should be able to control it from the VM's console. However, if you want to enable and configure SSH access to this host, I have configured a guide for doing so. Jump to "How to Enable SSH on Kali Linux" to enable the ssh service for this VM.

SIEM VM

SIEM is shorthand for Security Intrusion Events Manager. This is fancy security nomenclature for "log aggregator"; we will be having our IPS VM log its events here. We're going to be running Splunk on this VM. Splunk is a commercial program for managing logs on a pretty large scale. By default, and with no licensing, Splunk only allows you to collect or "index" 500MB worth of logs per day, however, there are /ways/ around this that we'll talk about later. Perform the following tasks to set up the "SIEM" VM:

- Download the latest Ubuntu Server 64-bit LTS release. In our case, this is Ubuntu Server 16.04.1, which can be found here: http://www.ubuntu.com/download/server
- Create a VM with the following settings:
    - Name the VM "SIEM"
    - Choose "Linux" as the type
    - Choose "Ubuntu (64-bit) as the version
    - 4GB of RAM
    - 80GB fixed size VDI
    - 1-2 virtual CPUs
- Once the VM is created, adjust the following settings:
    - Under the System settings, ensure the VM is using a PS/2 Mouse
    - Under Storage, point the virtual cd/dvd drive to where you downloaded the Ubuntu Server Linux ISO, so the VM has a bootable drive, and we have an actual OS to install
    - Disable Audio
    - Disable the USB controller
    - Under Network settings, make sure that Adapter 1(and only adapter 1) is enabled and that it is attached to "Host-Only Adapter", and that the name is "VirtualBox Host-Only Ethernet Adapter" -- the same as "Adapter 2" on the pfSense VM
        - Document the MAC address for Adapter 1, under its advanced settings
- Install Ubuntu Server
    - Be sure to install the Standard System Utilities and OpenSSH Server role when asked. You should not need to install any other roles or features for this server
    - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.1.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy
- After the installation is completed, power down the VM and adjust the following settings:

    - Remove the virtual optical drive under the Storage Tree in Storage settings
    - Under system settings, disable the Floppy and Optical drives, and make the Hard Disk the first device the VM boots from
- Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the LAN interface DHCP, for the MAC address of adapter 1 for the SIEM VM; assign it the IP address 172.16.1.3, making sure to enter a description that tells you which VM this static mapping belongs to
- Power on the virtual machine and do the following:
    - Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense
        - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
    - Verify the virtual machine can reach the internet
        - In a terminal/shell run the command `curl -I https://www.google.com`
            - This will confirm DNS can resolve domain names, and that the VM can reach the internet
            - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense
                - **Note:** The `ping` command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail
    - Update/patch the VM
        - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
            - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
    - Create a snapshot for the VM

IPS VM

This VM is going to be responsible for running the AFPACKET bridge between the IPS 1 and IPS 2 networks. In the case of VirtualBox, these are going to be two internal networks. Perform the following actions to set up the IPS VM:

- We will be installing Ubuntu Server 16.04.1 LTS, 64-bit. If you installed the SIEM VM already, you shouldn't need to download the ISO again, but just incase, download it here: http://www.ubuntu.com/download/server
- Create a VM with the following settings:
    - Name the VM "IPS"
    - Choose "Linux" as the type

    - Choose “Ubuntu (64-bit)” as the version
    - 2GB of RAM
    - 80GB fixed size VDI
    - 1-2 virtual CPUs (virtual CPUs can be adjusted under VM Settings > System > Processor tab once the VM has been created)
- Once the VM is created, adjust the following settings:
    - Under the System settings, ensure that the VM is using a PS/2 Mouse
    - Under Storage, point the virtual cd/dvd drive to where you downloaded the Ubuntu Server Linux ISO, so the VM has a bootable drive, and we have an actual OS to install
    - Disable Audio
    - Disable the USB controller
    - Under Network settings, make sure that Adapter 1, Adapter 2, and Adapter 3 are all enabled, and are using the following settings:
        - Adapter 1 should be attached to “Host-Only Adapter” named “VirtualBox Host-Only Ethernet Adapter”
            - Under Adapter 1, be sure to click “Advanced” and document the MAC address of this interface; This is going to be the primary network interface for this host
        - Adapter 2 should be attached to “Not attached”
            - Click on Advanced settings. Document the MAC address of this interface
        - Adapter 3 should be attached to “Not attached”
            - Click on Advanced settings. Document the MAC address of this interface
- Install Ubuntu Server
    - The installer will reach a section titled “Configure the network” and ask you which network card is the primary network interface for this system
        - Usually, these interfaces are labeled `eth0`, `eth1`, and `eth2`, but may be labeled differently in your case
        - In almost all cases, the first network adapter added to a VM in VirtualBox will line up with the first network interface listed on the “Configure the network screen (which is almost always `eth0`). Hit Enter, and wait for it to try and configure the interface via DHCP.
        - The installer will try to get an IP address for the network interface via DHCP. If this is not successful, select the `<Go Back>` option, and select one of the other two network interfaces listed, and try again.
        - If none of the interfaces work, verify the pfSense VM is running, connected to the “Management Network”, and that the DHCP service is enabled, then try again until DHCP configuration is successful.
    - Be sure to install the Standard System Utilities and OpenSSH Server role when asked. You should not need to install any other roles or features for this server.
    - **Note:** If you configured the firewall on the pfSense VM, and decided to install and

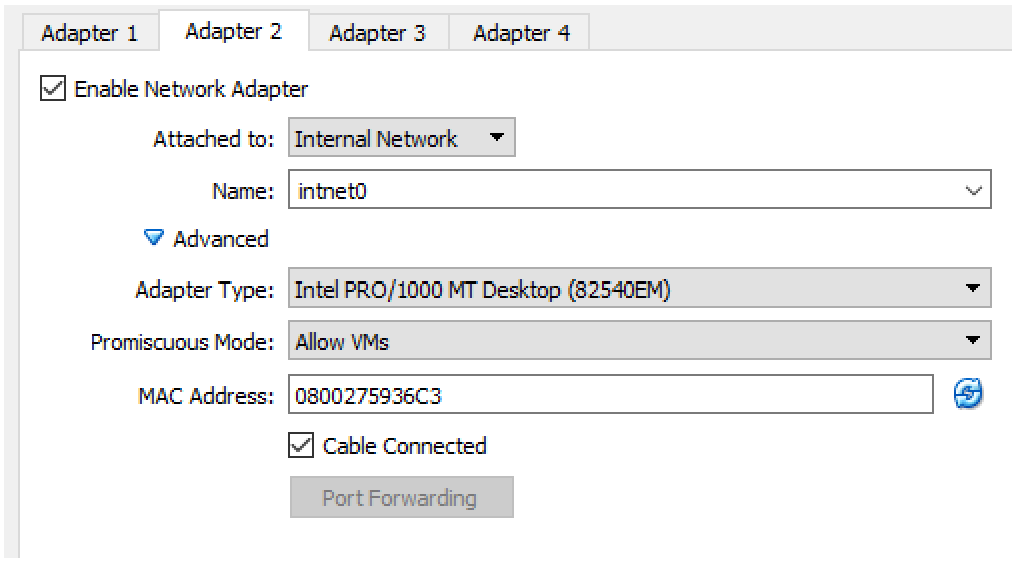
enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.1.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy.

- After the install is completed, power down the VM and adjust the following settings:
    - Remove the virtual CD/DVD under the Storage Tree in Storage settings
    - Under system settings, disable the Floppy and Optical drives, and make the Hard Disk the first device the VM boots from
    - Adapter 2 should be attached to “Internal Network” named “intnet” or “intnet0” -- the same as the kali VM
        - Under advanced settings, there is a dropdown labeled “Promiscuous Mode:”. Set this to “Allow VMs” ***(This is VERY important!)***
    - Adapter 3 should be attached to “Internal Network” named “intnet1”. You will have to name this new internal network. This network is the “IPS 2” network
        - Under advanced settings, there is a dropdown labeled “Promiscuous Mode:”. Set this to “Allow VMs” ***(This is VERY important!)***
- Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the LAN interface DHCP, for the MAC address of adapter 1 for the IPS VM; assign the IP address 172.16.1.4, making sure to enter a description that tells you which VM this static mapping belongs to
- Power on the virtual machine and do the following:
    - Determine which interface in the ips VM maps to which interface in the VM’s network settings
        - In the terminal, run `ifconfig -a | egrep HWaddr`
            - This command will show you the name of the of the interface and it’s HWaddr (Hardware Address - another name for MAC address). Compare this to the MAC addresses you documented earlier
            - You need to determine the name of the interface that has the MAC address associated with “Adapter1” in the VirtualBox network settings. This is going to be the only network interface this VM can use to talk to other hosts over the network
    - Verify that the static mapping for the VM’s IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter’s name is in the VM) was given the IP address you configured its static mapping for in pfSense
        - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
    - Verify the virtual machine can reach the internet
        - In a terminal/shell run the command `curl -I https://www.google.com`
            - This will confirm DNS can resolve domain names, and that the VM can reach the internet

        - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense
            - **Note:** The `ping` command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail
- Update/patch the VM
    - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
        - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
- Create a snapshot for the VM

Promiscuous Mode

In the configuration checklist above for the “IPS” VM, I very briefly mentioned configuring promiscuous mode for the network adapters on the. This setting MUST be set to “Allow VMs”. Promiscuous mode allows a network card that has this setting enabled to collect or “sniff” traffic observed on a given virtual network from other hosts. This configuration is extremely important and MUST be set on the network adapters connected to “intnet/intnet0” and “intnet1” for the “IPS” VM to perform its function correctly.



☑ Enable Network Adapter

Attached to: Internal Network

Name: intnet0

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow VMs

MAC Address: 0800275936C3

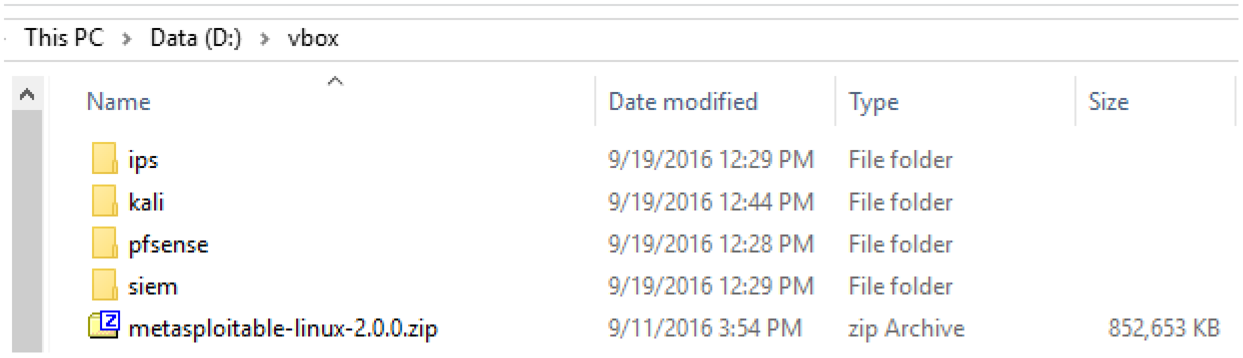
☑ Cable Connected

Metasploitable 2

Setting up Metasploitable 2 is a little bit different compared to the other virtual machines, because the VM comes in a prepackaged for use with VMware products. However, VirtualBox is able to utilize these prepackaged files as well. Download Metasploitable 2 from https://sourceforge.net/projects/metasploitable/files/Metasploitable2/metasploitable-linux-2.0.0.zip/download. Metasploitable 2 is distributed as a zip archive, so you need a compression utility to extract it.

Linux/OS X/BSD users usually have the unzip command available. Open a console/shell, navigate to where the file was downloaded (typically `~/Downloads`), then run `unzip metasploitable-linux-2.0.0.zip`. You may want to move the `metasploitable-linux-2.0.0` directory to the same directory you are storing the VDI files for your other lab VMs.

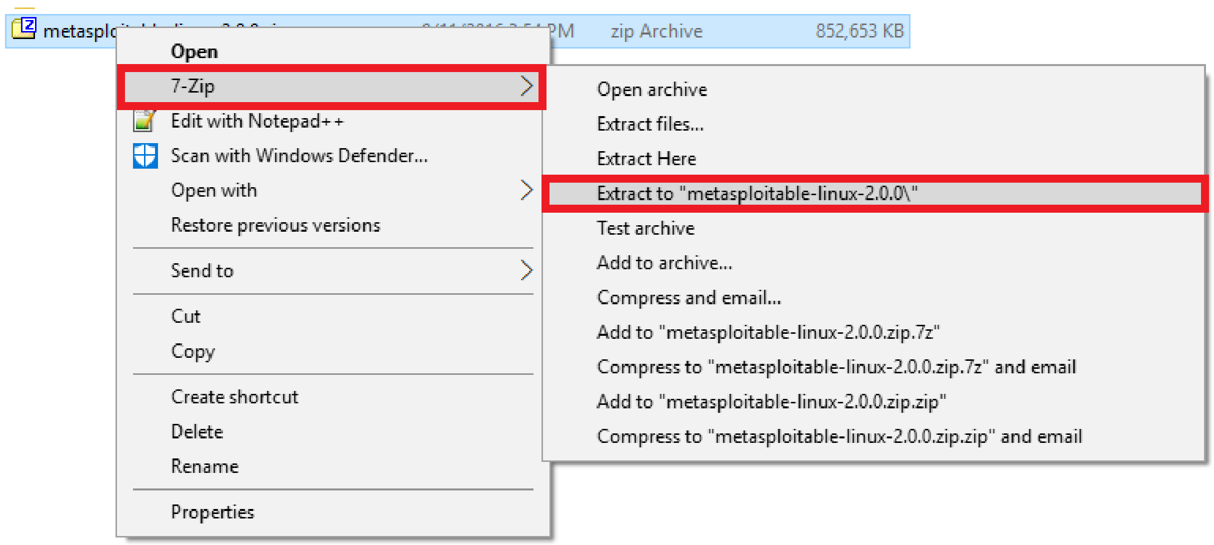
On Windows, I prefer 7-Zip (http://www.7-zip.org/download.html). Move the .zip file to the directory you are storing your VirtualBox VMs. In my case, I moved the .zip file to `D:\vbox`



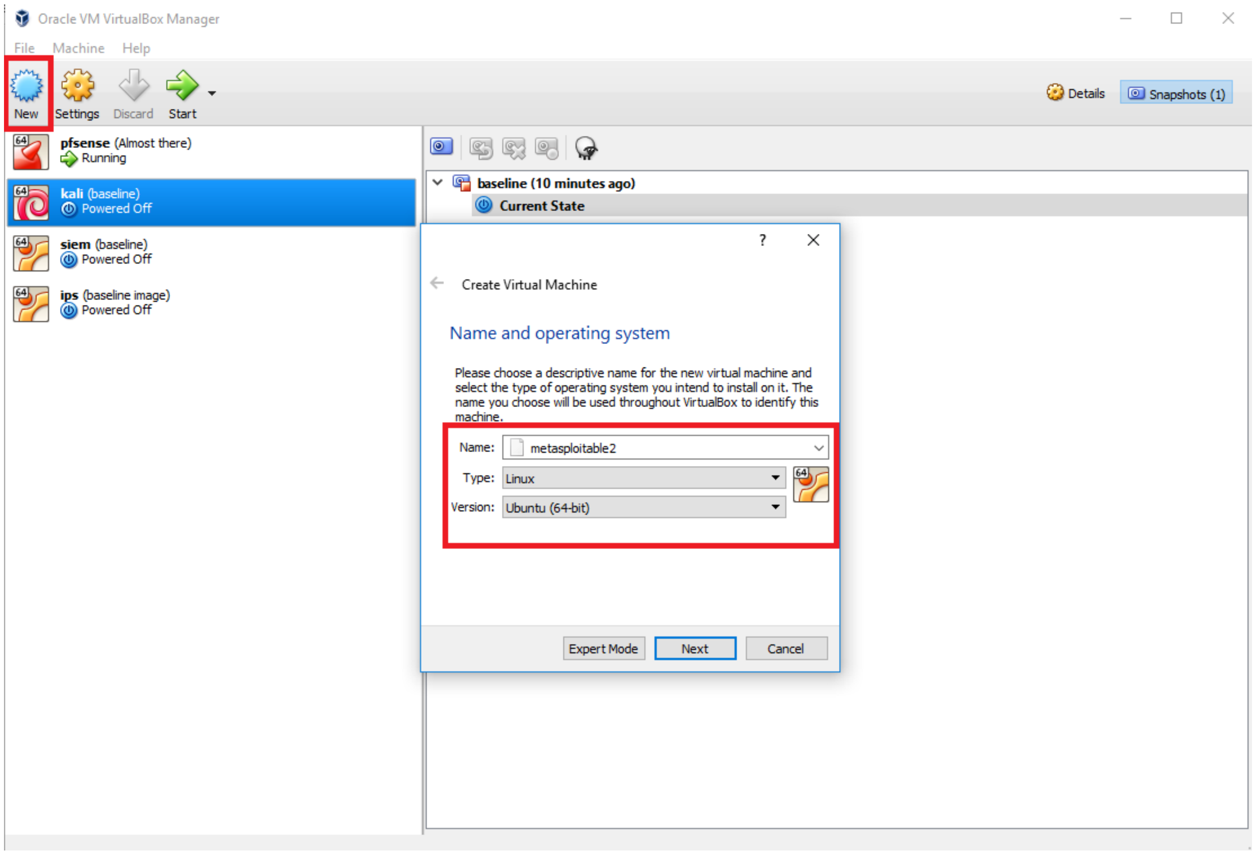
This PC › Data (D:) › vbox

Name	Date modified	Type	Size
ips	9/19/2016 12:29 PM	File folder	
kali	9/19/2016 12:44 PM	File folder	
pfsense	9/19/2016 12:28 PM	File folder	
siem	9/19/2016 12:29 PM	File folder	
metasploitable-linux-2.0.0.zip	9/11/2016 3:54 PM	zip Archive	852,653 KB

Right click on the zip file, and use 7-Zip to extract the files in the zip to `metasploitable-linux-2.0.0`


metasplo
M
zip Archive
852,653 KB
Open
7-Zip
Edit with Notepad++
Scan with Windows Defender...
Open with
Restore previous versions
Send to
Cut
Copy
Create shortcut
Delete
Rename
Properties
Open archive
Extract files...
Extract Here
Extract to "metasploitable-linux-2.0.0\"
Test archive
Add to archive...
Compress and email...
Add to "metasploitable-linux-2.0.0.zip.7z"
Compress to "metasploitable-linux-2.0.0.zip.7z" and email
Add to "metasploitable-linux-2.0.0.zip.zip"
Compress to "metasploitable-linux-2.0.0.zip.zip" and email

Fire up VirtualBox and select “New” to create a new VM. Name the VM “Metasploitable 2”. For the OS type, select “Linux”. For the version, select “Ubuntu (64-bit)”. Then click next.


Oracle VM VirtualBox Manager
File Machine Help
New Settings Discard Start
Details Snapshots (1)
pfsense (Almost there) Running
kali (baseline) Powered Off
siem (baseline) Powered Off
ips (baseline image) Powered Off
baseline (10 minutes ago)
Current State
Create Virtual Machine
Expert Mode
Next
Cancel

Name and operating system

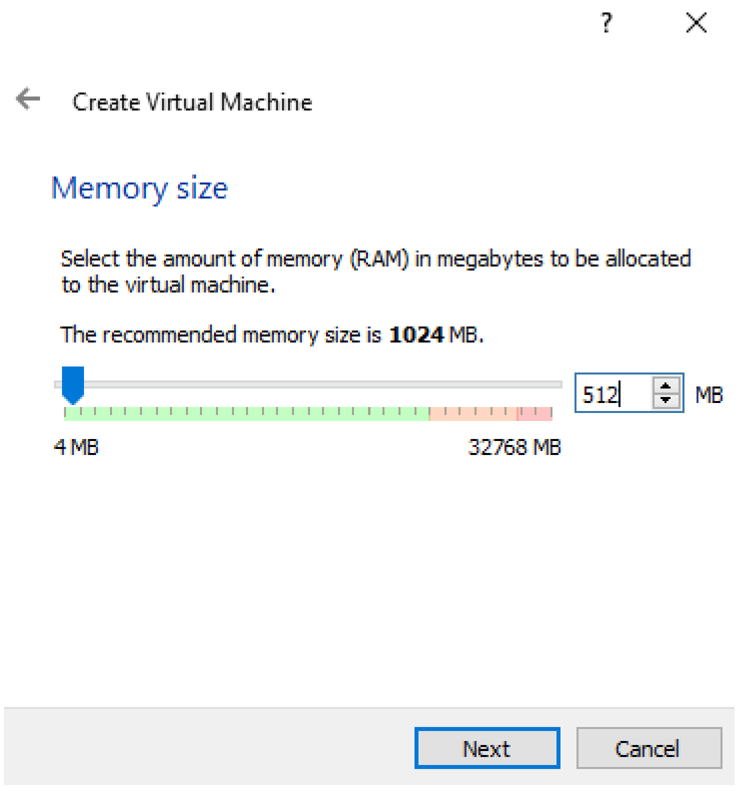
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: metasploitable2

Type: Linux

Version: Ubuntu (64-bit)

Allocate 512MB of RAM to the VM, and click next.



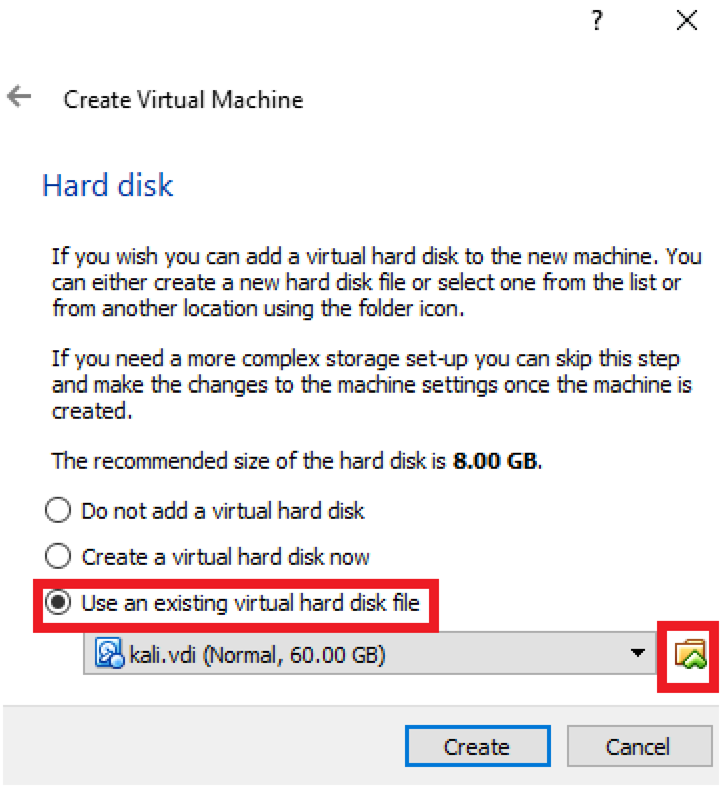
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

512 MB

On the next screen, click the “Use an existing virtual hard disk file” radio button, and click the folder with the green icon. This will open an explorer window.



Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

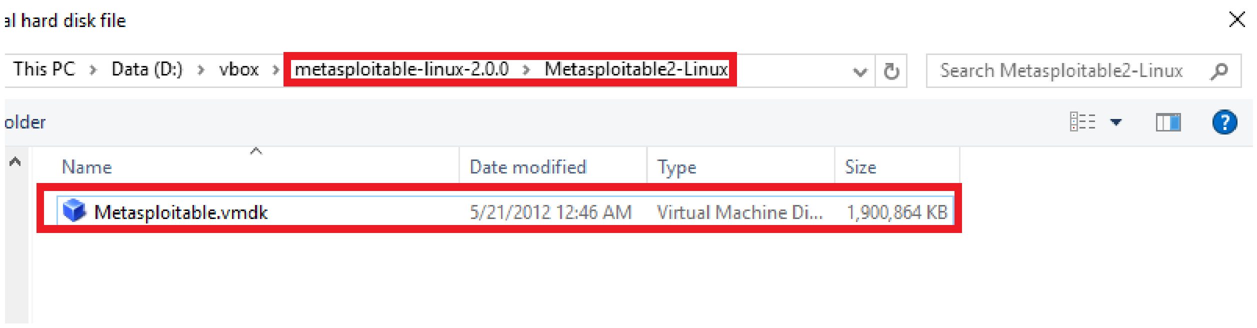
If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **8.00 GB**.

○ Do not add a virtual hard disk
○ Create a virtual hard disk now
● Use an existing virtual hard disk file

kali.vdi (Normal, 60.00 GB)

Navigate to the `metasploitable-linux-2.0.0` directory we extracted from the Metasploitable 2 zip file moments ago. Inside, is a subdirectory named `Metasploitable2-Linux`. Double click to enter that directory. In my case, the full directory path is `D:\vbox\metasploitable-linux-2.0.0\Metasploitable2-Linux`. Double click on the `Metasploitable.vmdk` file.

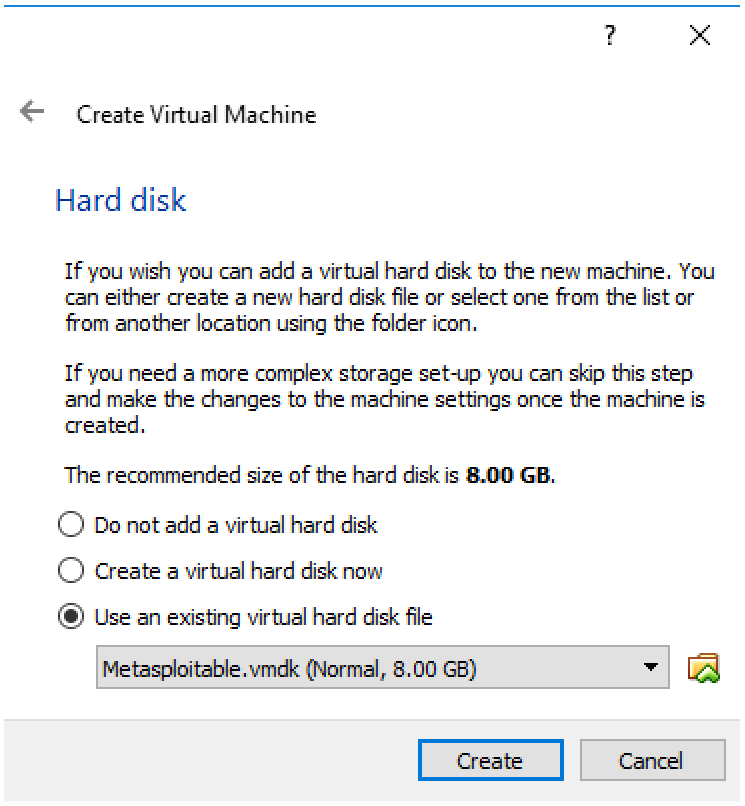


al hard disk file

This PC › Data (D:) › vbox › metasploitable-linux-2.0.0 › Metasploitable2-Linux

Name	Date modified	Type	Size
Metasploitable.vmdk	5/21/2012 12:46 AM	Virtual Machine Di...	1,900,864 KB

Selecting the vmdk file should have brought you back to the “Hard disk” screen. Click the create button. The virtual machine is added to the main VirtualBox menu.


Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **8.00 GB**.

Do not add a virtual hard disk

Create a virtual hard disk now

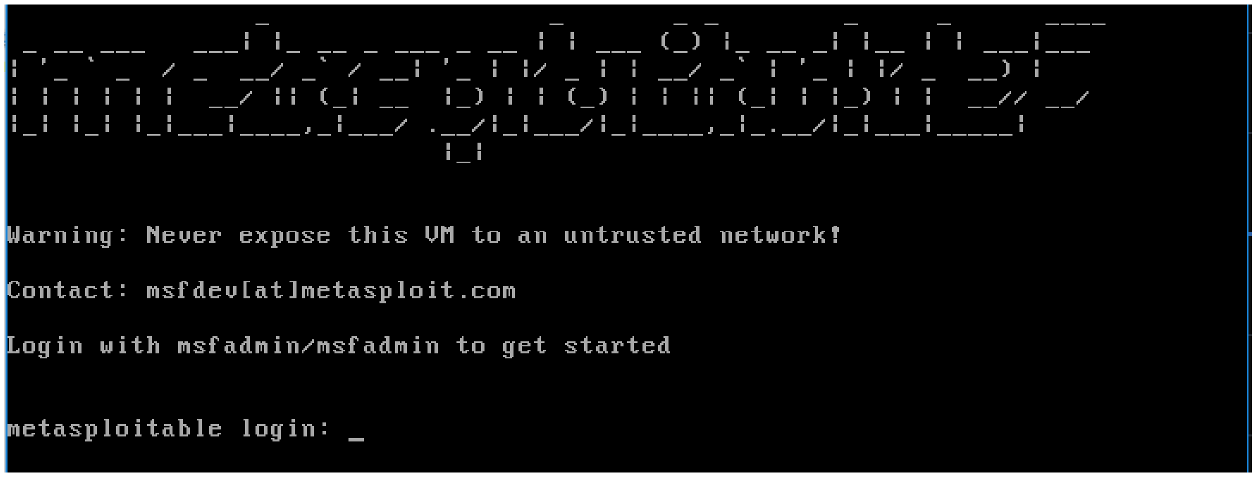
[✓ Use an existing virtual hard disk file]

Metasploitable.vmdk (Normal, 8.00 GB)

The “Metasploitable 2” VM should be added to our list on the VirtualBox main menu. Next up is something you should be familiar with by now: we are going to adjust the settings of the virtual machine.

- Go to the “System” screen. Under “Boot Order”, uncheck the Floppy and Optical options, and make the Hard Drive the first boot option. Change the “Pointing Device:” to “PS/2 Mouse”
- Remove the virtual CD/DVD under the Storage Tree in Storage settings
- Disable Audio
- Disable the USB controller
- Navigate to “Network” and under the “Adapter 1” tab, ensure the adapter is enabled. Attach the adapter to “Internal Network” named “intnet1” (the EXACT same as Adapter 3 for the “IPS” VM). Click on “Advanced” and document the MAC address of the interface.
- In pfSense, under Services > DHCP Server, add a static DHCP mapping under the OPT1 interface for this virtual machine. Give it the IP address 172.16.2.3 and make sure to enter a description that states which VM the mapping belongs to!

At this point, the VM should be bootable. Feel free to power it on to make sure there aren’t any problems. If you’re greeted with a login prompt, then this indicates everything is working fine. Power down the virtual machine and take a snapshot.



```
                _                  _       _ _        _     _      ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: _
```

Note: This VM is located on “intnet1”, which we haven’t bridged to “intnet/intnet0” with the “IPS” VM yet. So in spite of having a static DHCP mapping, it has no way of reach out to the “pfSense” virtual machine for an IP address yet. We are going to fix this soon.

Next Steps

All of the VirtualBox setup steps have been completed. However, you’re not quite done yet. Here is a checklist of tasks to complete:

- If you are running Virtualbox on a Windows host, and haven’t completed the chapter “Defense in Depth for Windows Hosted Hypervisors”, I would very highly suggest doing so in order to harden your hypervisor host.
- While not strictly necessary, you may also want to complete the “Remote Lab Management” guide. Specifically, the sections related to Windows Remote Access, Enabling SSH on Kali Linux, and if you’re lazy like me, the section on Securing root SSH access. This will allow you to remotely manage all of your lab VMs, as well as finish the IPS and Splunk setup guides much more easily than through the VirtualBox console.
- You still need to install IDS/IPS software on the IPS VM to get a functioning AFPACKET bridge between the “IPS 1” and “IPS 2” networks. Check out the “IPS Installation Guide” to learn how to do this with either Snort or Suricata as your IPS software of choice.
- The SIEM VM needs to have Splunk installed and configured. You’ll need to complete the “Splunk Installation Guide”.
- You may want to consider reading the Automated Patching for Linux Lab VMs chapter, and implementing the `updater.sh` script for your Linux VMs.
- Do you want some ideas on where to take your lab? Check out the chapter “In Your Own Image”, for some tips on how to mold your VM lab to better suit your needs.

Setup - VMware Fusion Pro

Note: This guide is written for VMware Fusion Pro, specifically, the latest version as of writing, version 8.5. Fortunately for us, OS X is the only OS VMware Fusion Pro was designed for is OS X. This guide was written using OS X 10.12 *Sierra*, the latest version as of writing, but should work just fine with 10.11 *El Capitan*.

Be aware that this guide specifically requires VMware Fusion Pro. Pro edition allows us to modify virtual networking settings easily and create additional virtual networks through its virtual network editor tool. We need the ability to modify and create virtual networks easily, and only Fusion Pro has a network editor application.

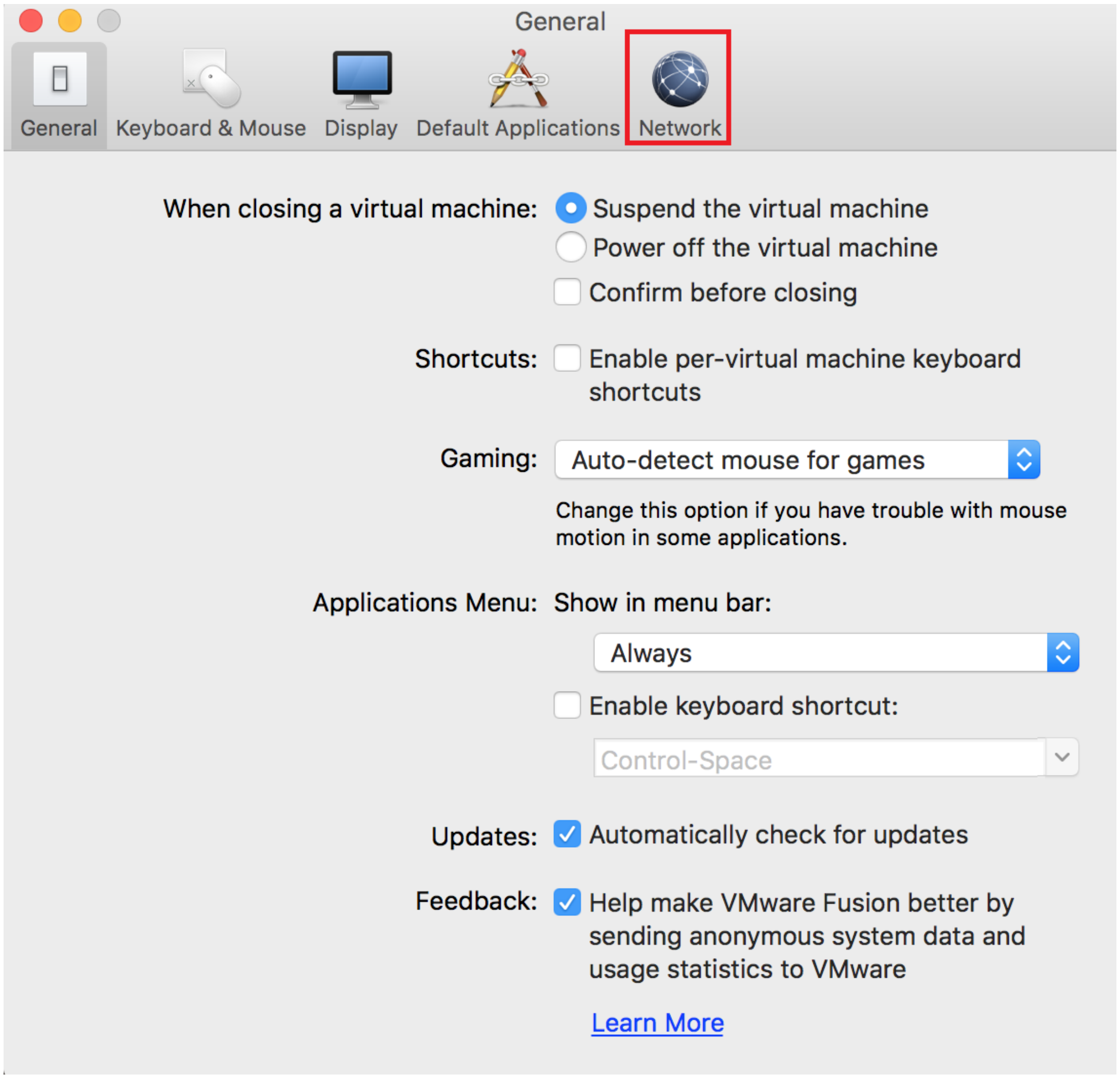
Installation

First, you need to download VMware Fusion Pro from VMware website. You can a trial copy here: http://www.vmware.com/products/fusion/fusion-evaluation.html. Run the installer; the default settings should be fine. If you have a license, enter it when prompted to do so during the installation. If you do not have a license, continue on and use the trial time you have for now.

Hypervisor Preferences

Start up VMware Fusion Pro and you'll be greeted by a screen that automatically assumes you want to create or add a new VM.

Before we go down that route however, we have to make some changes to the hypervisor. In the main menu bar, click on *VMware Fusion*, then click on *Preferences…* . This will open the hypervisor preferences menu. We're interested in the Network options, so click on the *Network* icon.



When closing a virtual machine: [✓ Suspend the virtual machine] Power off the virtual machine
☐ Confirm before closing

Shortcuts: ☐ Enable per-virtual machine keyboard shortcuts

Gaming: Auto-detect mouse for games

Change this option if you have trouble with mouse motion in some applications.

Applications Menu: Show in menu bar: Always

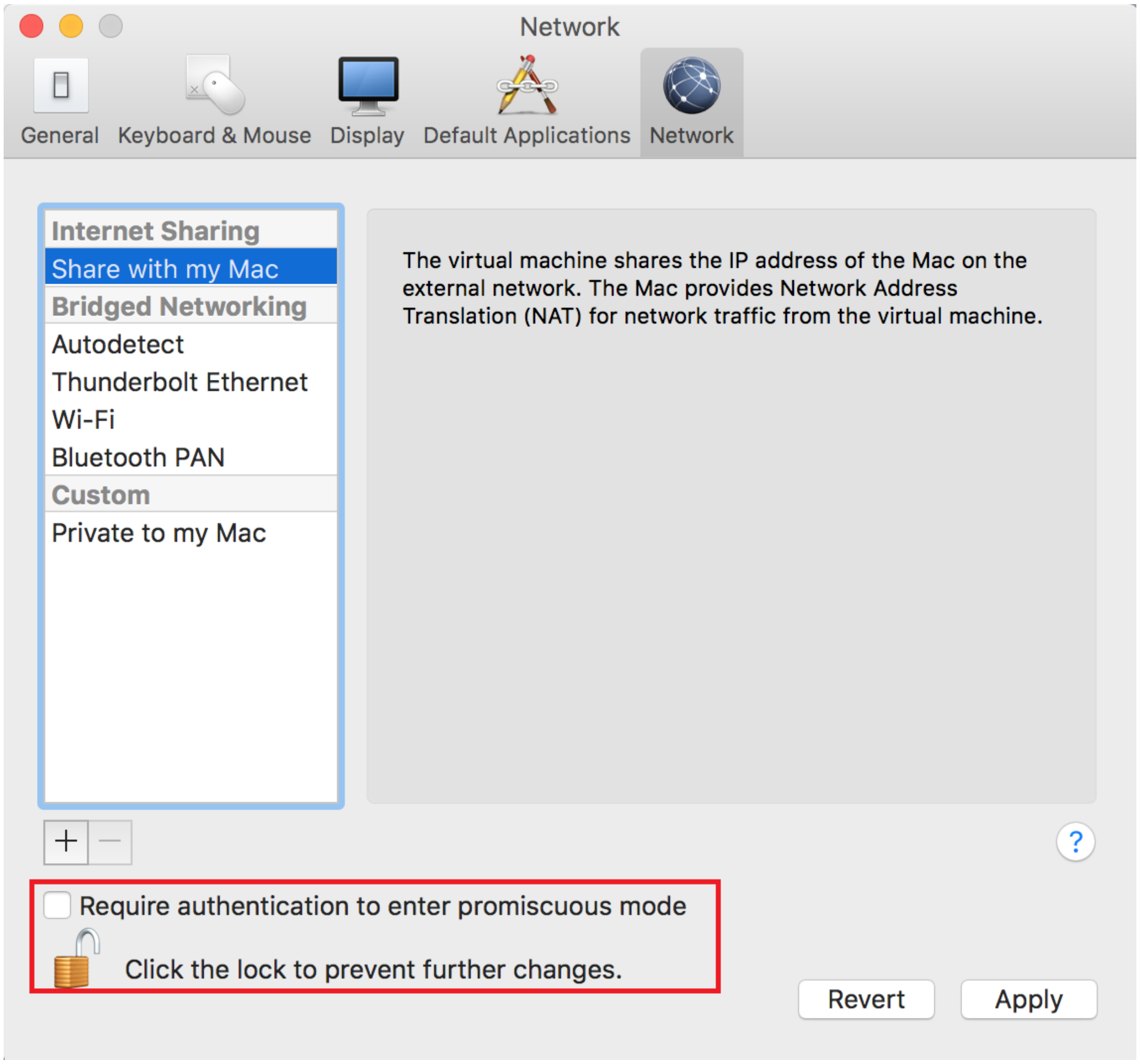
☐ Enable keyboard shortcut: Control-Space

Updates: ✓ Automatically check for updates

Feedback: ✓ Help make VMware Fusion better by sending anonymous system data and usage statistics to VMware

Learn More

The *Network* menu allows you to define network bridging settings, as well as define custom network settings. Before you can change anything, however, you need to click the lock on the bottom of the menu, and input your password, when prompted. After doing so, uncheck the *Require authentication to enter promiscuous mode* checkbox. By doing so, you will no longer be prompted to enter your Mac user's password every time a VM's network interface switches to promiscuous mode to perform certain tasks. As an example, the IPS VM we are going to set up needs promiscuous mode to be enabled. If you're paranoid or want that extra layer of security, you may choose to leave this option enabled. Just keep in mind that every time you restart the IPS VM, you are going to be asked to enter your password twice, as you have authenticate individually for each interface that wants to go into promiscuous mode, with said virtual machine having two of them.



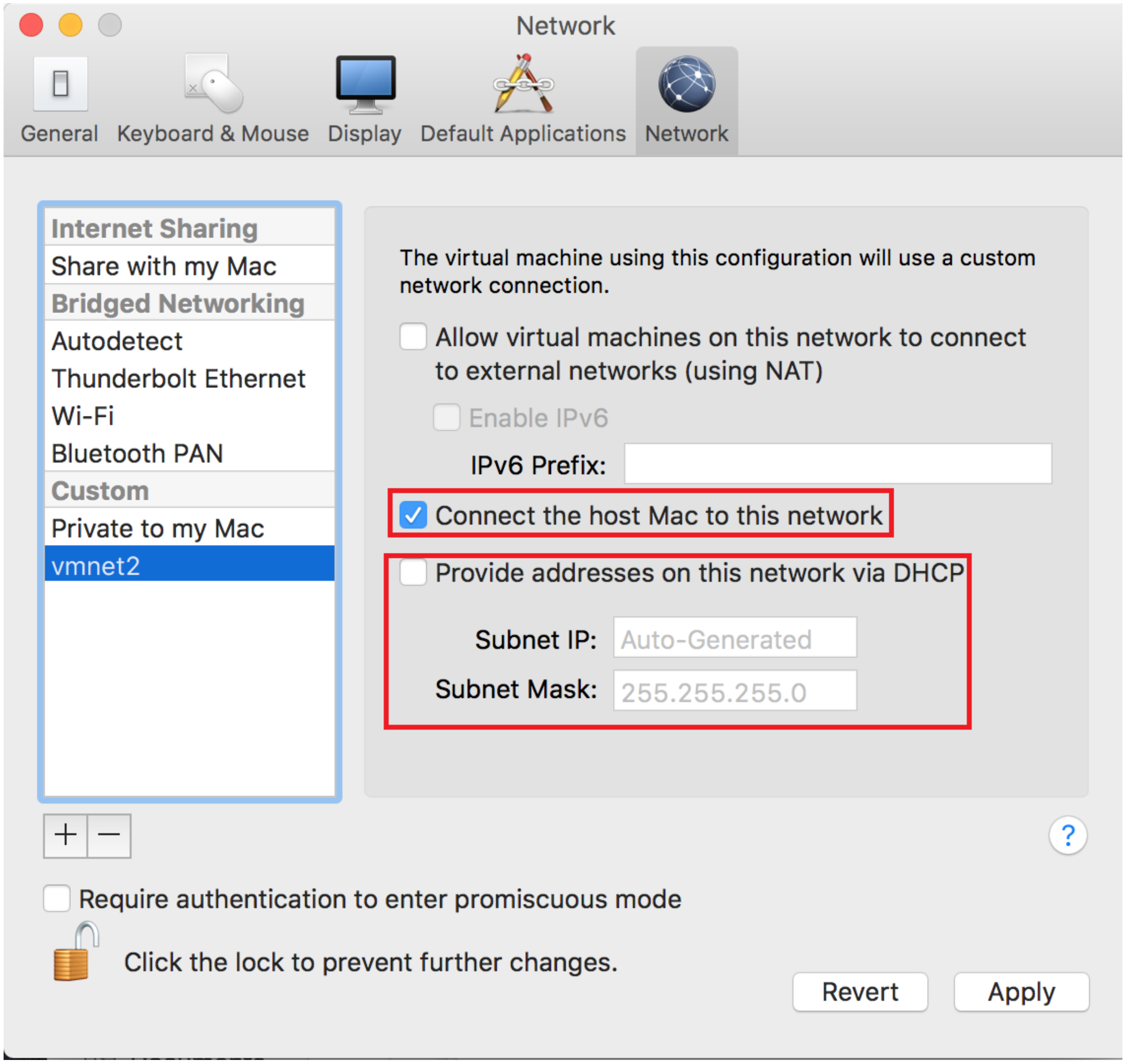
The virtual machine shares the IP address of the Mac on the external network. The Mac provides Network Address Translation (NAT) for network traffic from the virtual machine.

Require authentication to enter promiscuous mode

Click the lock to prevent further changes.

Next, we are going to create three virtual networks. Click the + sign underneath the pane that lists the networks and connections available for VMware Fusion. A new virtual network should appear under *Custom*, labeled *vmnet2*. By default, Fusion Pro will highlight this new network and show the corresponding settings in the pane to the right. Uncheck the *Provide addresses* *on this network via DHCP* checkbox.

For some reason that is completely beyond me, you have no capability to modify DHCP settings for the default host-only network in VMware Fusion (8.5 pro, at least as of this writing), without having to modify installation files directly. As we do not want VMware to handle the DHCP settings on our lab's *Management* network, using Fusion's preconfigured host-only network isn't an option. Instead, we've had to create our own network (vmnet2) to serve this purpose.



The virtual machine using this configuration will use a custom network connection.

☐ Allow virtual machines on this network to connect to external networks (using NAT)

☐ Enable IPv6

IPv6 Prefix:

☑ Connect the host Mac to this network

☐ Provide addresses on this network via DHCP

Subnet IP: Auto-Generated

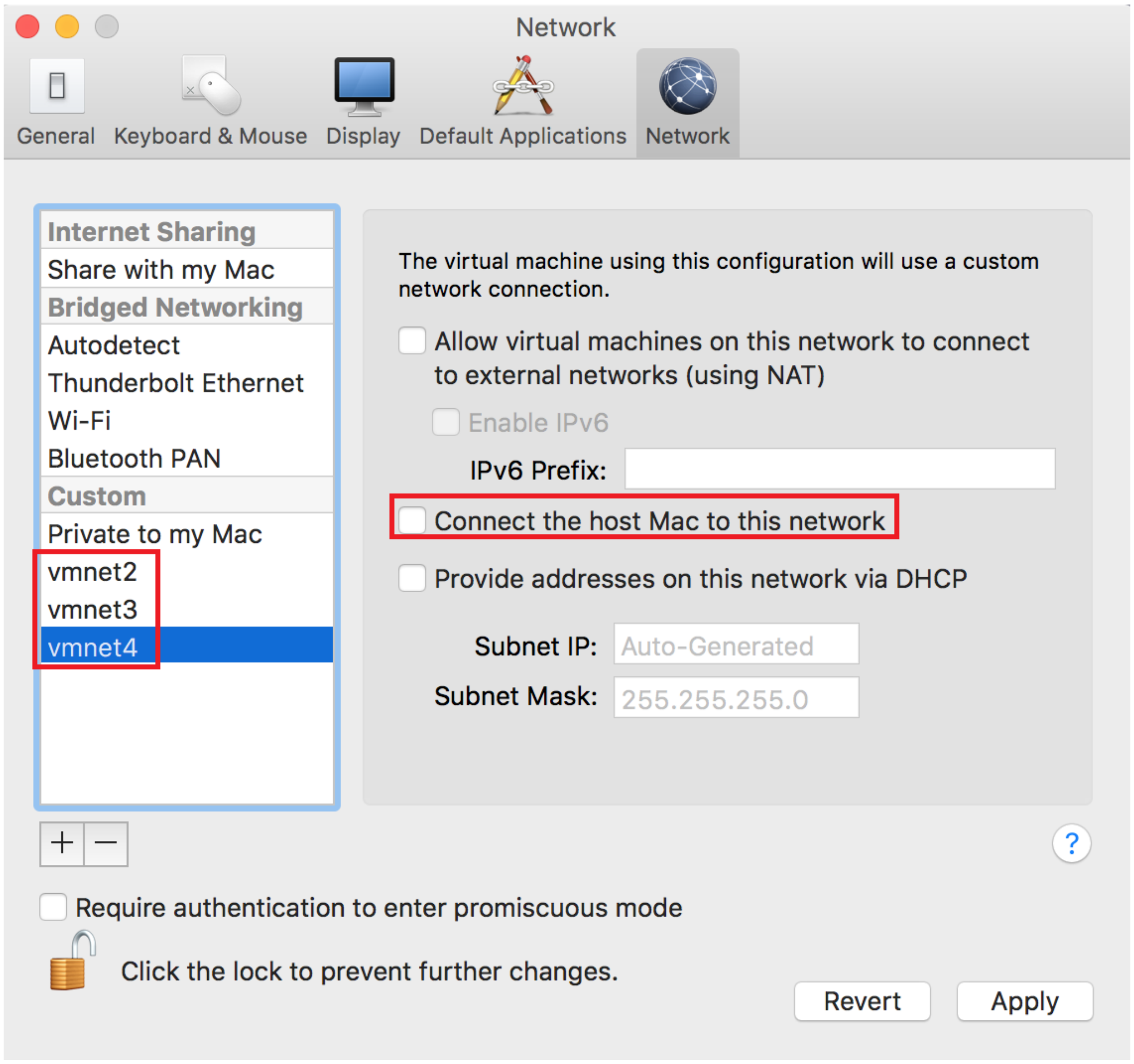
Subnet Mask: 255.255.255.0

☐ Require authentication to enter promiscuous mode

Click the lock to prevent further changes.

After creating *vmnet2*, we're going to create two more virtual networks. They should be labeled *vmnet3* and *vmnet4,* respectively. Just like with *vmnet2*, uncheck the *Provide addresses on this network via DHCP* checkbox. Additionally, **uncheck the *Connect the host Mac to this network* checkbox for both *vmnet3* and *vmnet4*. It is very important that you do this.** These networks will serve as the *IPS 1* and *IPS 2* networks, and we do not want the Mac (our host system) to have direct exposure to these networks.

When finished, you should have three new virtual networks. We are going to use *vmnet2* as our *Management* network, *vmnet3* is going to serve as the *IPS 1* network, and *vmnet4* as the *IPS 2* network.



The virtual machine using this configuration will use a custom network connection.

☐ Allow virtual machines on this network to connect to external networks (using NAT)

☐ Enable IPv6

IPv6 Prefix:

☐ Connect the host Mac to this network

☐ Provide addresses on this network via DHCP

Subnet IP: Auto-Generated

Subnet Mask: 255.255.255.0

☐ Require authentication to enter promiscuous mode

Click the lock to prevent further changes.

When you’re done, click the lock icon to prevent further changes. When prompted by Fusion, click *Apply* to accept the settings, then exit out of the menu by clicking the red circle in the upper left corner of the *Preferences* menu.

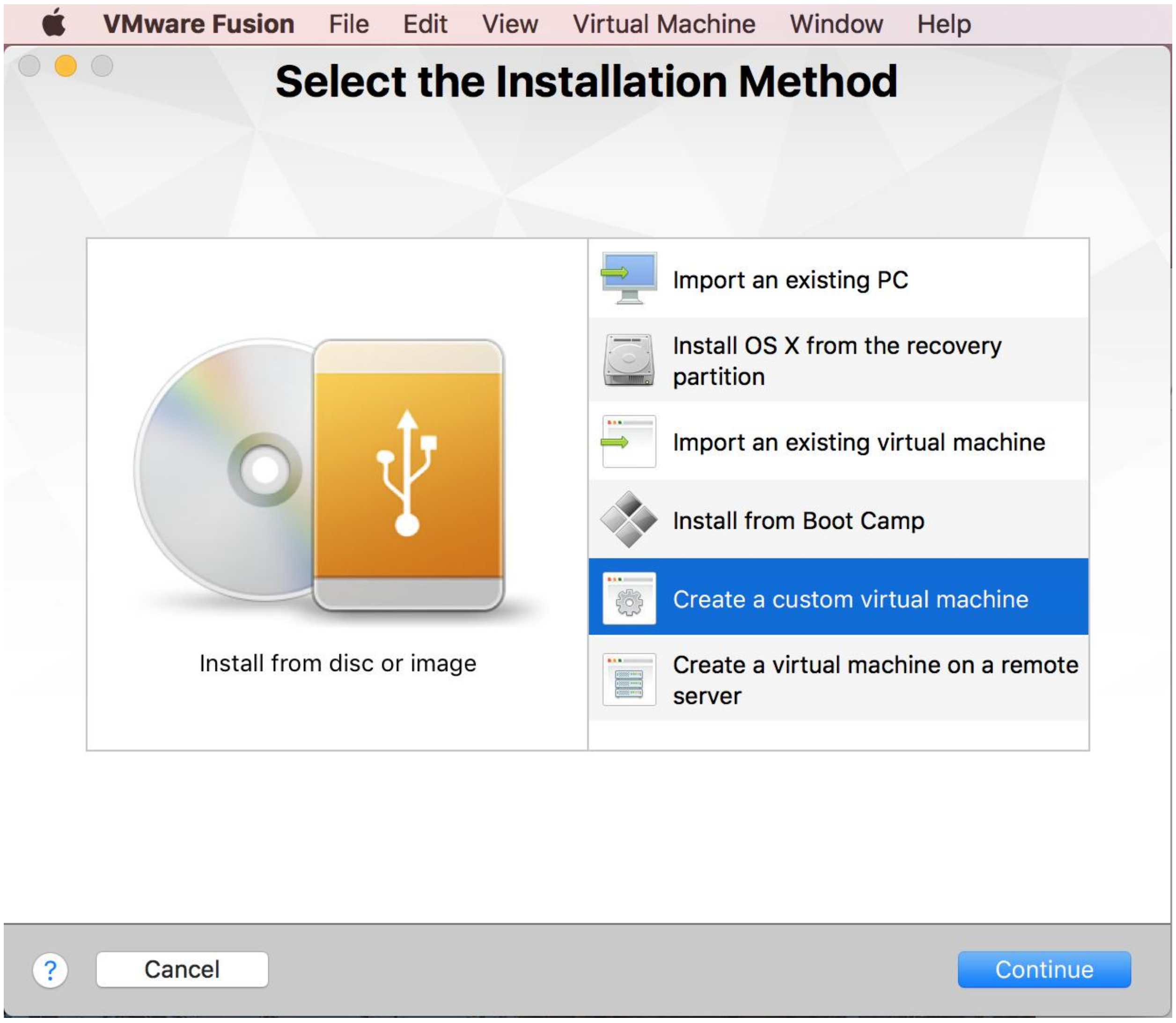
Creating the First VM, pfSense

pfSense is the keystone holding this entire configuration together. Personally, it's my favorite firewall distro due to ease of use, the amount of functionality it includes out of the box, combined with a plugin/add-on system for additional functionality. If you have the CPU, RAM, and disk, pfSense can easily be converted into a so-called “Next-Generation” firewall.

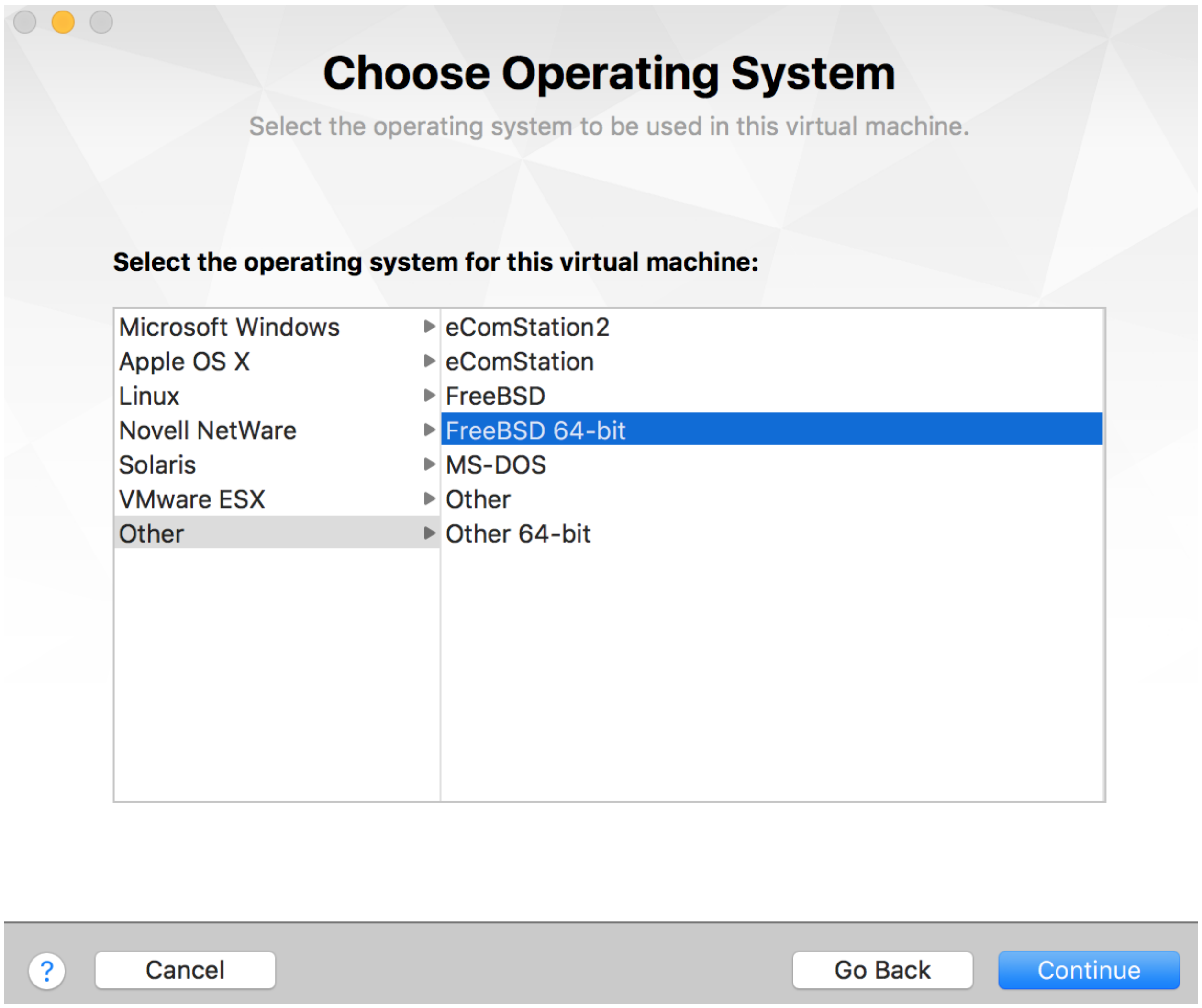
Make your way to https://www.pfsense.org/download/. Download the latest installation ISO for the amd64 architecture. While you’re at it, you may want to download a compression utility. The pfSense maintainers distribute pfSense as an ISO image file compressed with `gzip`. This means that we’ll need to decompress the ISO file at some point. By default, if you double click on a gzipped file in Finder on OS X, it will automatically make a decompressed copy for you. OS X also includes the `gzip` or `gunzip` command line utilities for decompressing gzipped files as well.

Adding a New VM

A window titled *Select the Installation Method* might still be open from starting VMware Fusion earlier to edit the hypervisor preferences. If it is not (e.g. you hit cancel), click *File* on the main menu bar, then choose *New…* to bring up said window and start the new VM wizard. Select *Create a custom virtual machine*, then click *Continue*.


VMware Fusion
File
Edit
View
Virtual Machine
Window
Help
Select the Installation Method
Install from disc or image
Import an existing PC
Install OS X from the recovery partition
Import an existing virtual machine
Install from Boot Camp
Create a custom virtual machine
Create a virtual machine on a remote server
?
Cancel
Continue

The next screen has you choose the the type and the version of the operating system to be installed within the VM. Select *Other* in the left pane, *FreeBSD 64-bit* on the right pane, then click *Continue*.



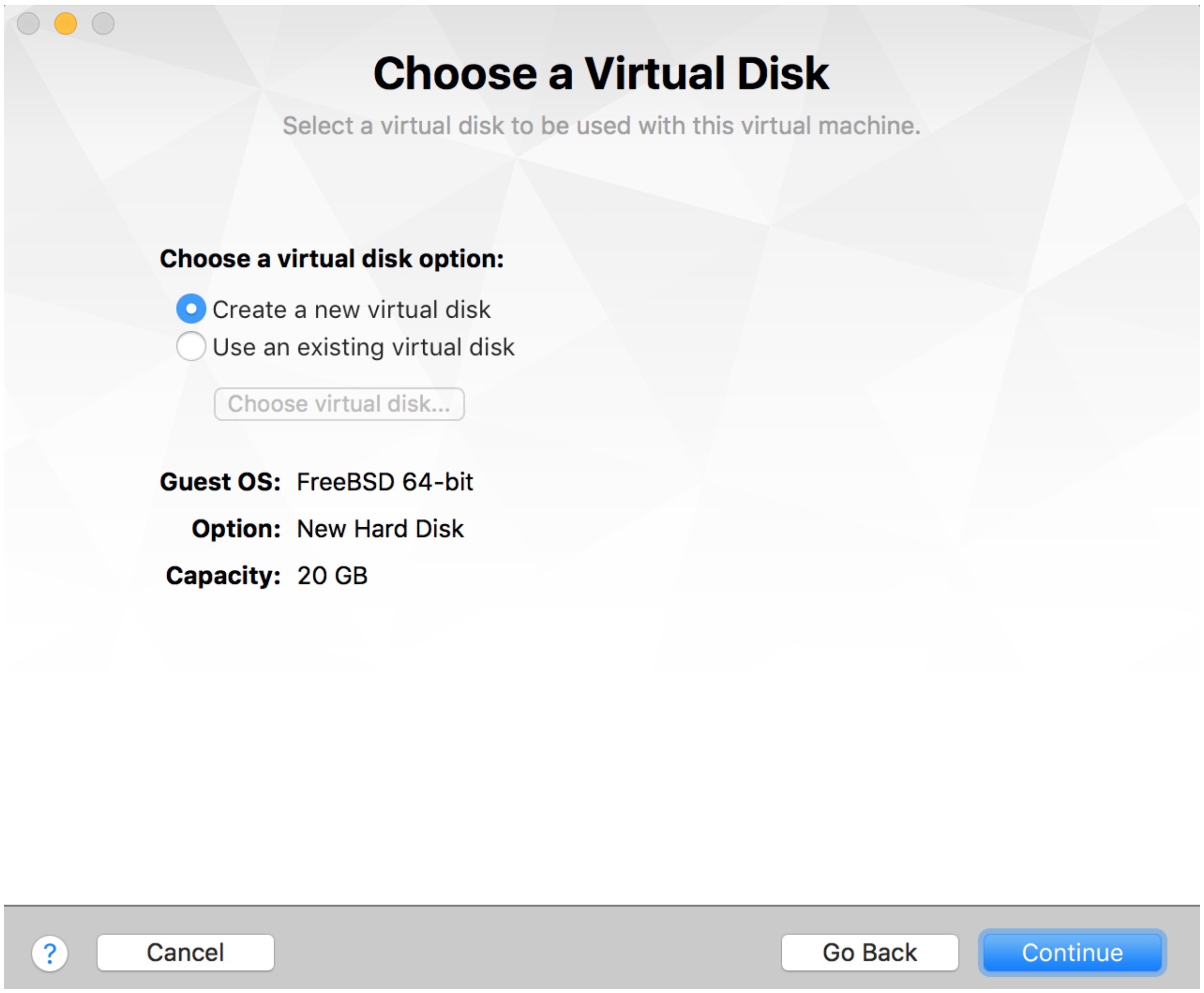
Choose Operating System

Select the operating system to be used in this virtual machine.

Select the operating system for this virtual machine:

Microsoft Windows	eComStation2
Apple OS X	eComStation
Linux	FreeBSD
Novell NetWare	FreeBSD 64-bit
Solaris	MS-DOS
VMware ESX	Other
Other	Other 64-bit

On the next screen you are offered options for the virtual disk to be used with the pfSense VM. Ensure that the *Create a new virtual disk* radio button is selected, then click Continue.


Choose a Virtual Disk

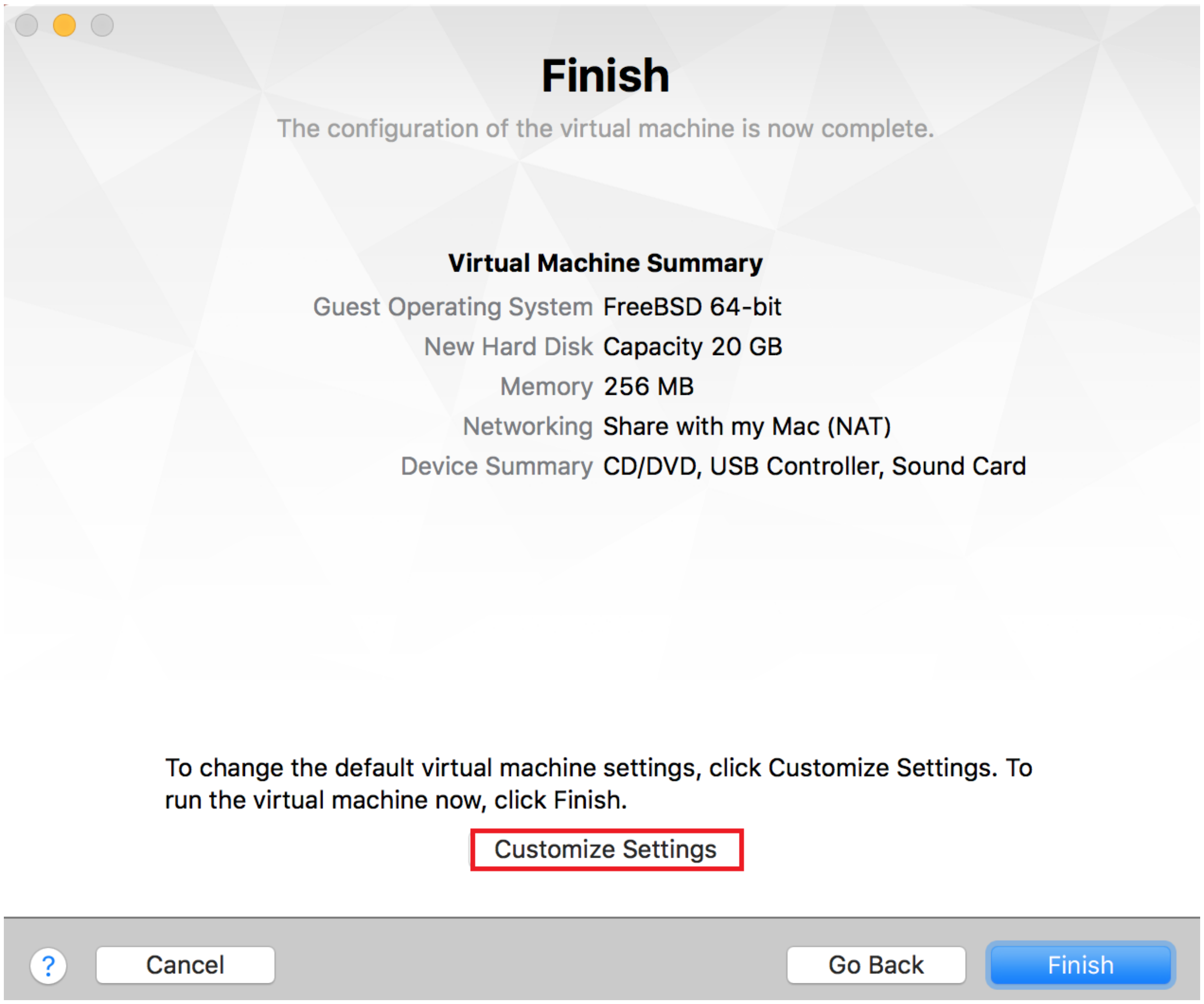
Select a virtual disk to be used with this virtual machine.

Choose a virtual disk option:

[✓ Create a new virtual disk] Use an existing virtual disk

Guest OS:	FreeBSD 64-bit
Option:	New Hard Disk
Capacity:	20 GB

The next screen is entitled *Finish*, but we're not done just yet. Click on the *Customize Settings* button.



Finish

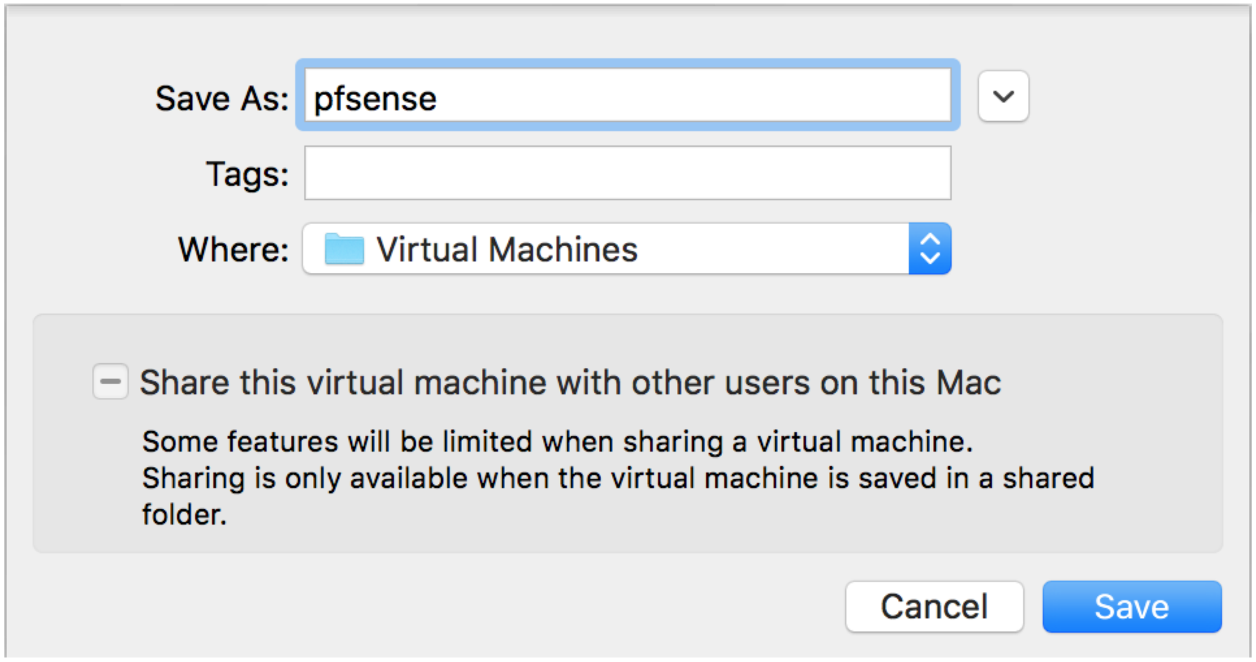
The configuration of the virtual machine is now complete.

Virtual Machine Summary

Guest Operating System	FreeBSD 64-bit
New Hard Disk	Capacity 20 GB
Memory	256 MB
Networking	Share with my Mac (NAT)
Device Summary	CD/DVD, USB Controller, Sound Card

To change the default virtual machine settings, click Customize Settings. To run the virtual machine now, click Finish.

As soon as you click *Customize Settings* (or if you clicked *Finish* by accident), a small window asking to save your VM will pop up. Input “pfSense” in the *Save As:* input field. If you have any custom tags you want associated to this VM, you can enter them in the *Tags:* input field. The *Where:* field allows you to send your VM to a custom location.


Save As: pfsense

Tags:

Where: Virtual Machines

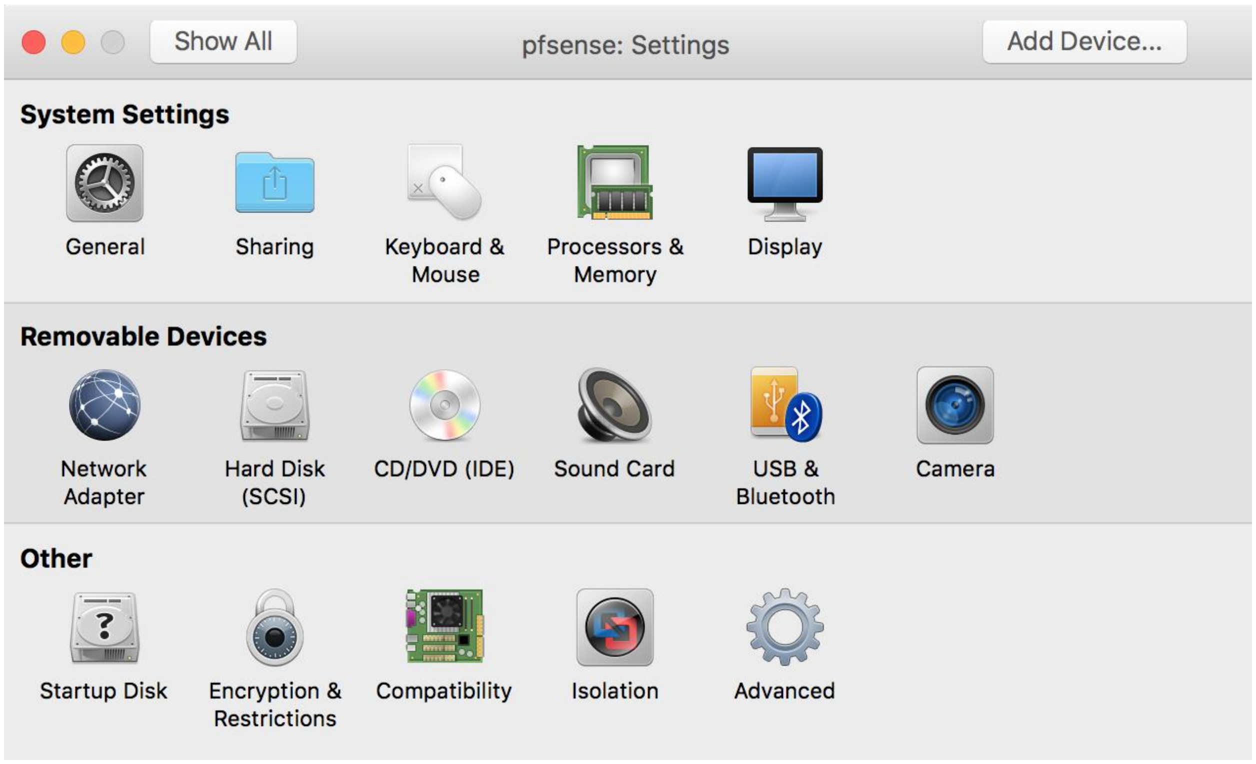
Share this virtual machine with other users on this Mac

Some features will be limited when sharing a virtual machine.
Sharing is only available when the virtual machine is saved in a shared folder.

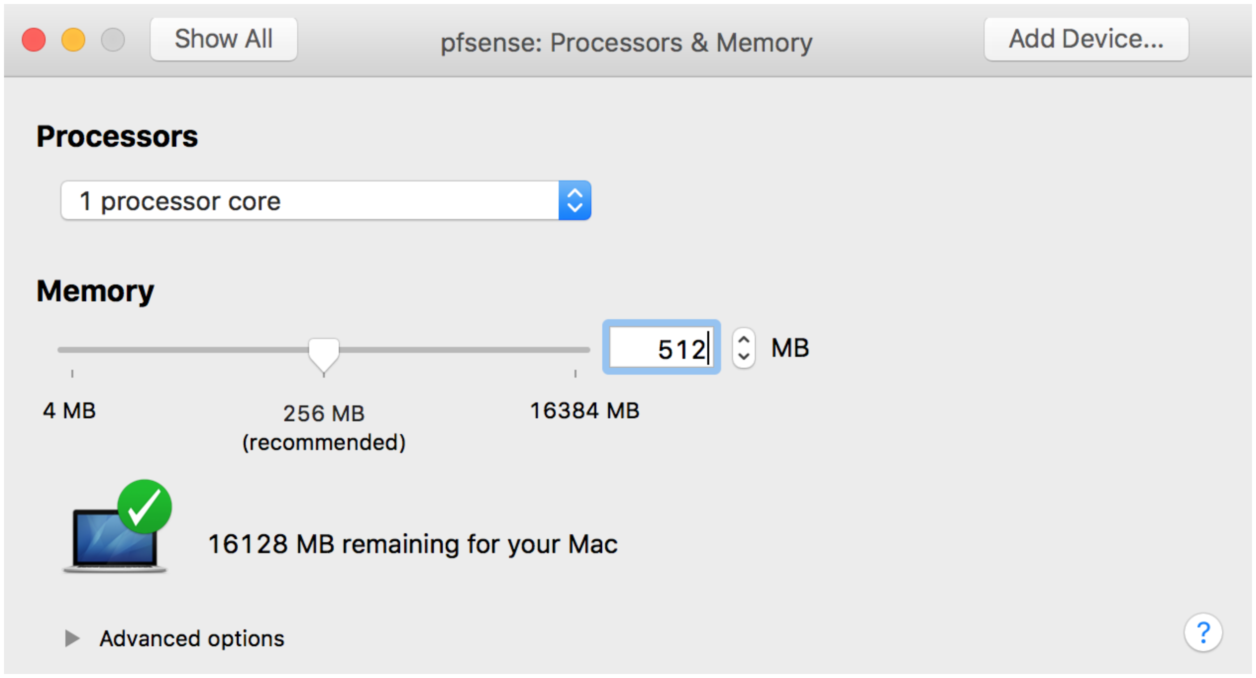
By default, VMware Fusion saves your VMs to `/Users/[username]/Documents/Virtual Machines`. If you had another destination you wanted to store your virtual machines to, you can specify that here. For our purposes, the default location suits us just fine. Unfortunately, **there is no universal ‘save all of my VMs to this custom folder’ option. If you use a custom destination, you have to remember to specify that destination on a per-VM basis**. When you have renamed your VM and are satisfied with tags and storage location, click the *Save* button.

Note: If you accidentally clicked the *Finish* button instead of *Customize Settings*, save your VM as instructed above. VMware Fusion will attempt to start your VM automatically, but the virtual machine will fail to boot, because there are no bootable drives attached to the VM yet. Click the console window that pops up after the New VM wizard is complete to highlight it. On the VMware Fusion menu bar, select *Shut Down* from the *Virtual Machine* menu item. Then, to enter the *pfSense: Settings* menu mentioned below, you may click the wrench icon in the pfSense console window, or while the console window is still highlighted, select the *Settings…* option from the *Virtual Machine* menu. Please note that almost all of the settings we will be modifying below require the VM to be powered off in order to modify them.

A window for pfSense appears on your desktop, and a smaller window entitled *pfSense: Settings* appears. First, click on *Processors & Memory*.



On this screen, you can adjust how many CPU cores and/or how much RAM is available to the VM. Change the input box to “512” MB, then click the *Show All* button to proceed back to the settings menu.



Processors

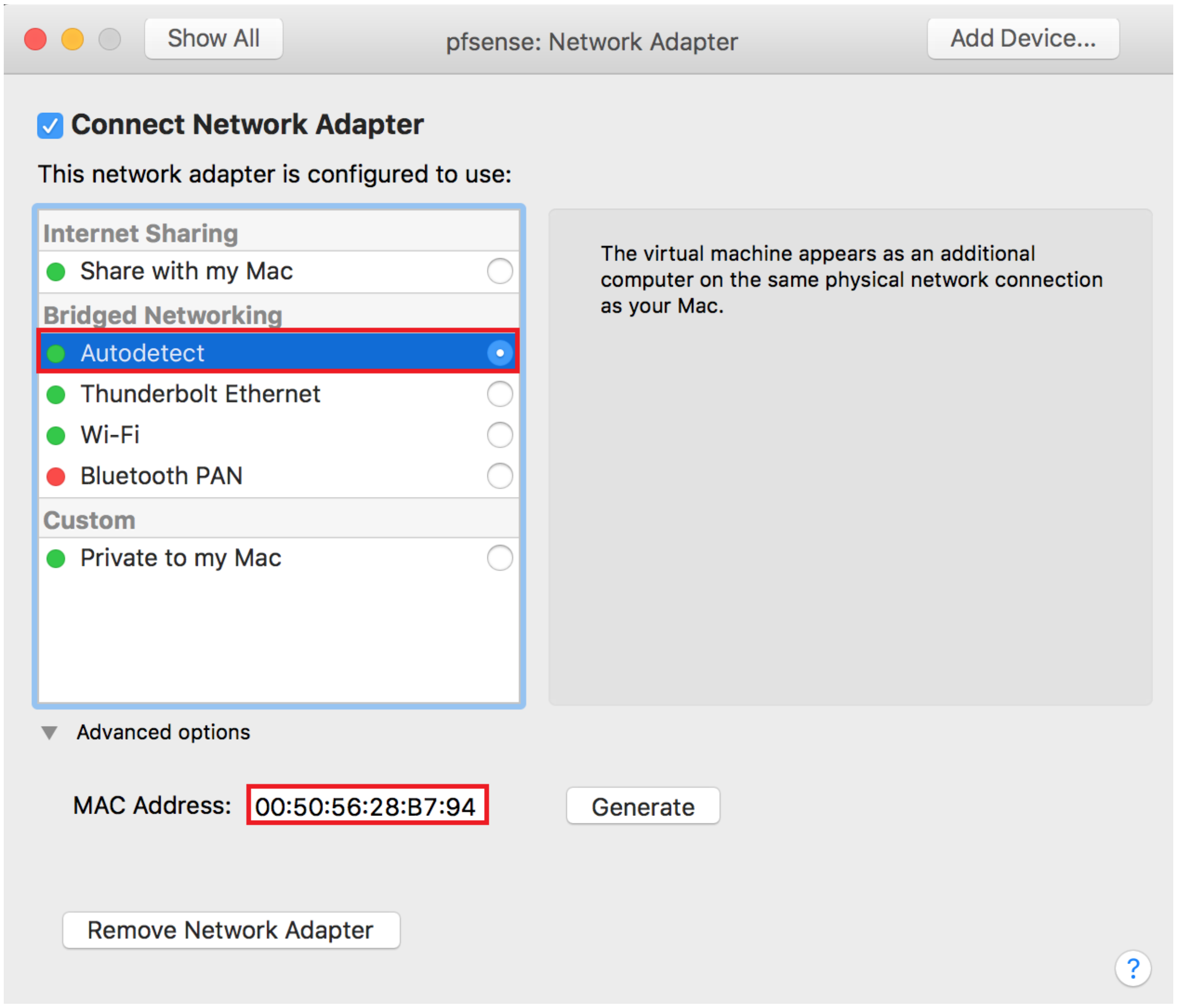
1 processor core

Memory

512 MB

16128 MB remaining for your Mac

Back on the main screen, click on *Network Adapter* under the *Removable Devices* listing. On this screen, you can adjust what virtual network this network adapter will connect to. Ensure that the *Connect Network Adapter* checkbox is checked, then change the radio button to *Autodetect* under *Bridged Networking* in the network list pane. Click on the triangle next to *Advanced options* under the network list pane. Click on the *Generate* button to the right of the *MAC Address:* field to generate a MAC address for this network adapter. **Record this MAC address, and note that it is from the adapter that is connected to the *Bridged* Network. We'll need this information later.** Click *Show All* in the title bar of the window to return to the settings menu.



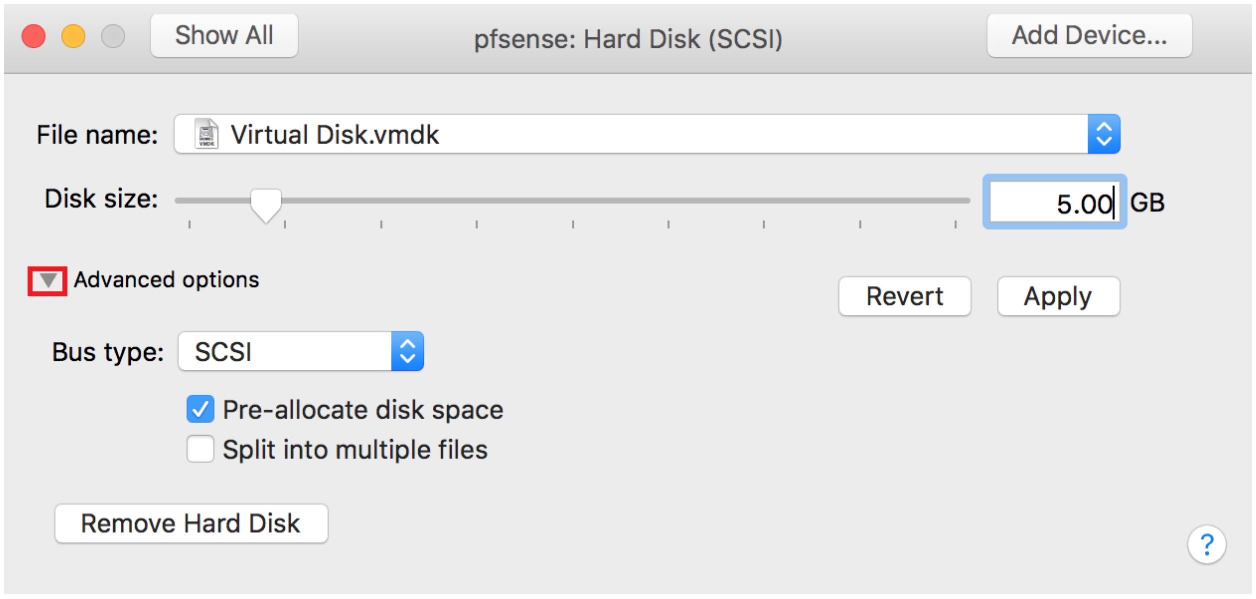
✓ Connect Network Adapter

This network adapter is configured to use: Autodetect

The virtual machine appears as an additional computer on the same physical network connection as your Mac.

MAC Address: 00:50:56:28:B7:94

On the main screen under *Removable Devices*, select *Hard Disk (SCSI)*. This brings you to a page where (among other things) you can modify the size of the VM's virtual disk.. In the input box, change the default setting to “5.00” GB. Click on the triangle next to *Advanced options*. Check the option labeled *Pre-allocate disk space*, and uncheck the option *Split into multiple files*. When you are finished, click *Show All* to go back to the main menu. You will be asked to confirm the changes you have made. Click *Apply* and wait a few moments for VMware Fusion to re-size the virtual disk.



pfsense: Hard Disk (SCSI)

File name: Virtual Disk.vmdk

Disk size: 5.00 GB

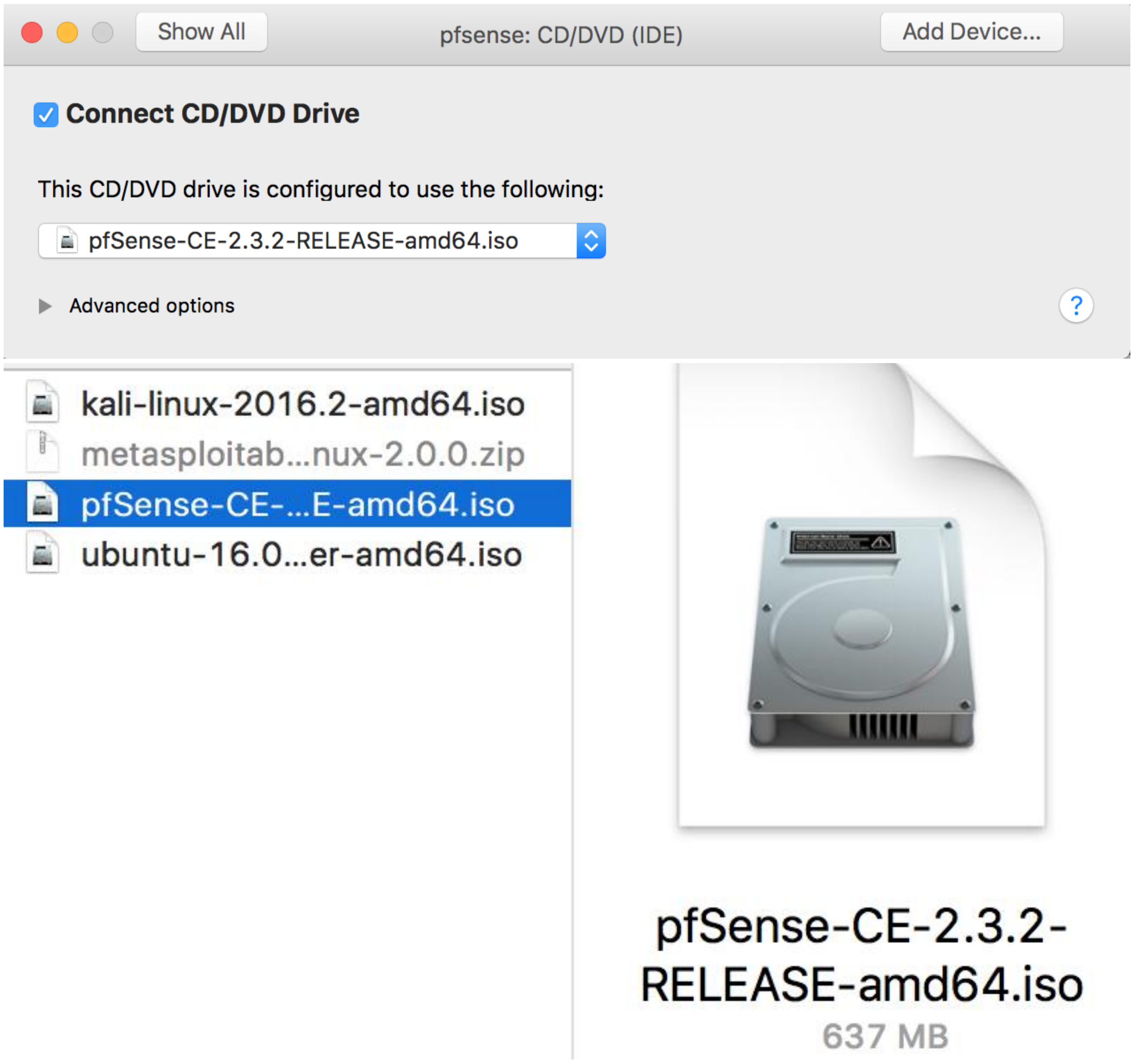
Advanced options

Bus type: SCSI

✓ Pre-allocate disk space

☐ Split into multiple files

Back in the main menu, click *CD/DVD (IDE)* in the Removable Devices section.. Next, in the drop-down labeled, T*his CD/DVD drive is configured to use the following:*, select the *Choose a disc or disc image…* option. This opens a Finder window.. Navigate to where you stored the uncompressed/unzipped pfSense ISO. In my case, I placed the file in `/Users/[username]/Documents/ISOs/`. After selecting the pfSense ISO, click *Show All* to return to the main menu.



✓ Connect CD/DVD Drive

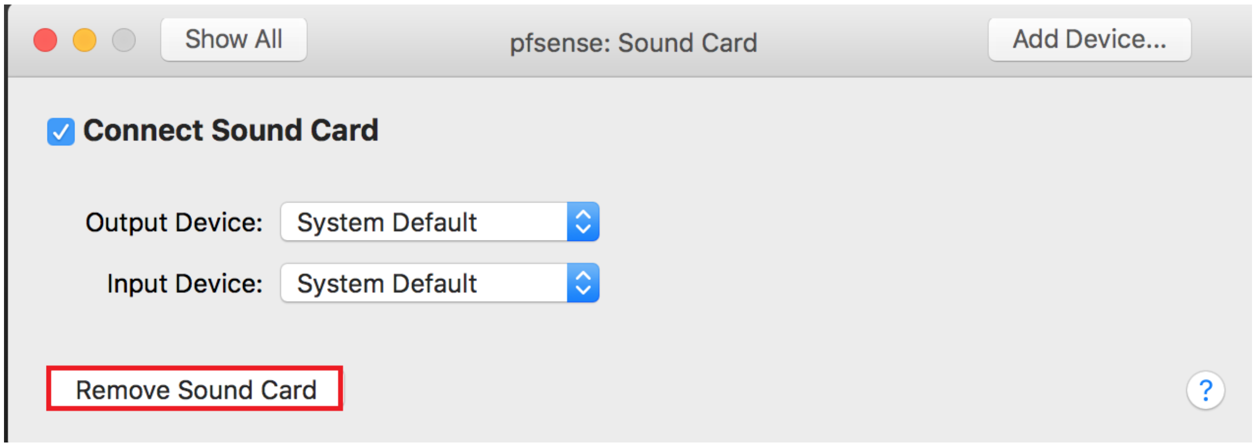
This CD/DVD drive is configured to use the following:

pfSense-CE-2.3.2-RELEASE-amd64.iso

kali-linux-2016.2-amd64.iso
metasploitab...nux-2.0.0.zip
pfSense-CE-...E-amd64.iso
ubuntu-16.0...er-amd64.iso

pfSense-CE-2.3.2-RELEASE-amd64.iso
637 MB

The upcoming tasks follow a similar pattern, so I'm so I'm going to cover them together. I'm going to have you remove the Sound Card, the Camera, and the USB & Bluetooth options. First, select *Sound Card* on the main menu. On the page that appears, click the *Remove Sound Card* button. VMware Fusion will ask you to confirm this action; click the *Remove* button. Doing so should bring you back to the main menu automatically.

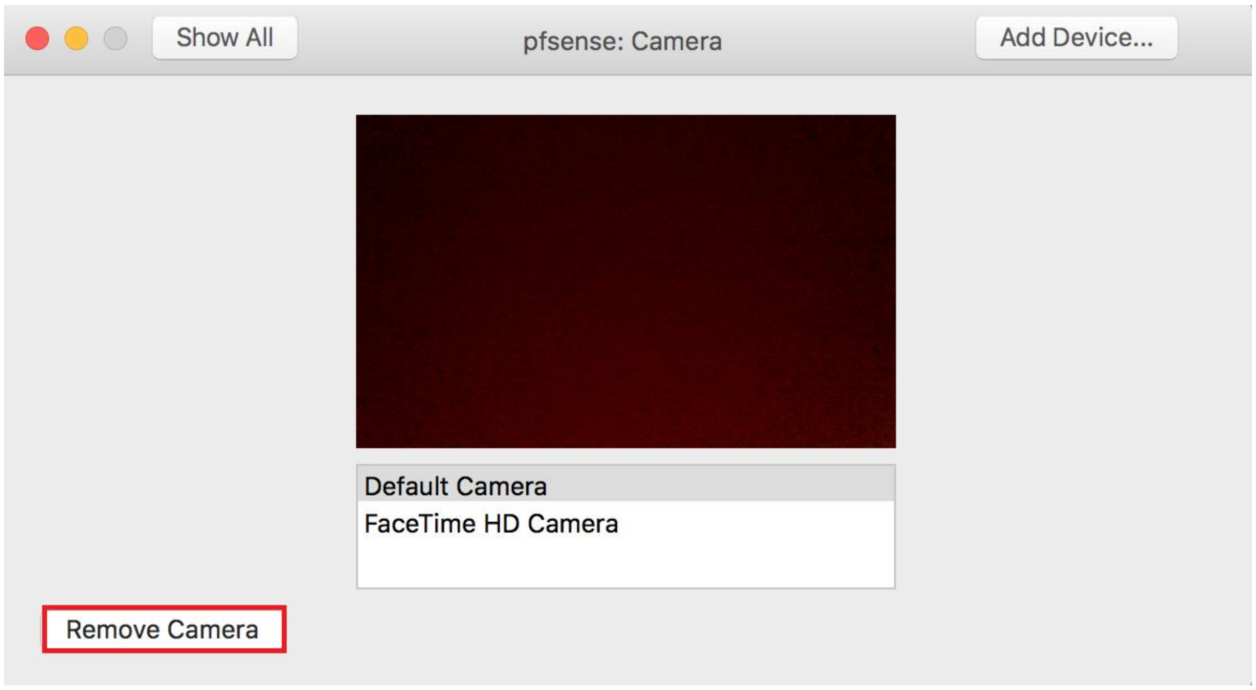


☑ Connect Sound Card

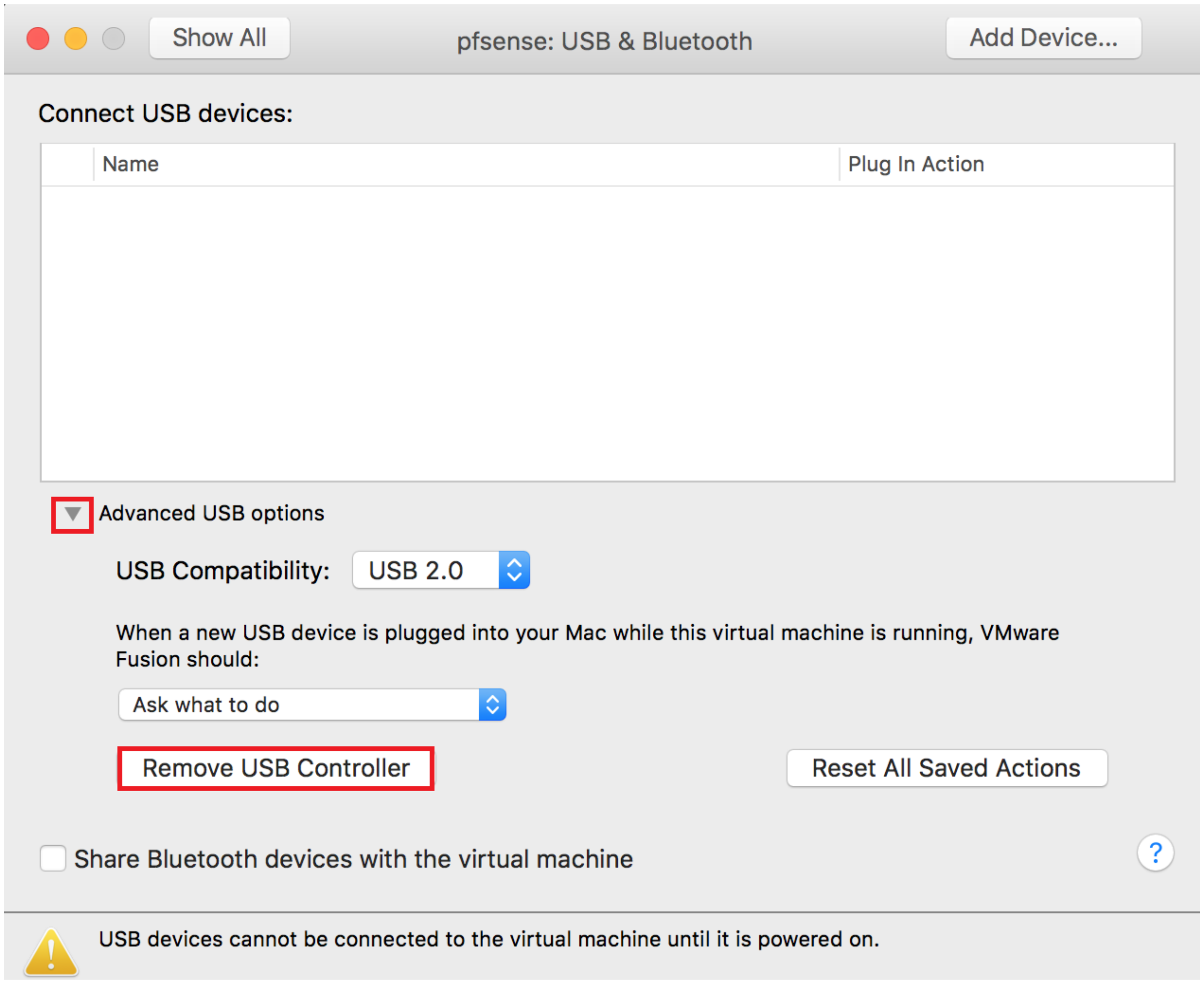
Output Device: System Default

Input Device: System Default

Repeat these steps for the *Camera* option accordingly. Select the corresponding icon on the main menu, click the *Remove Camera* button, then confirm the removal by clicking *Remove* in the dialogue box that appears.. This should return you to the main menu again.



Finally, click on *USB & Bluetooth*. Fusion gets tricky and hides the *Remove USB Controller* button under *Advanced USB options*. Click the triangle next to it to expose and then click said button. Confirm removal of the USB Controller by clicking *Remove* in the dialogue box. Afterwards, return to the main menu by clicking Show All in the title bar of the window..The *USB & Bluetooth* option still shows up in the *Removable Devices* section, but there will be no USB controller available for the virtual machine anymore.



USB Compatibility: USB 2.0

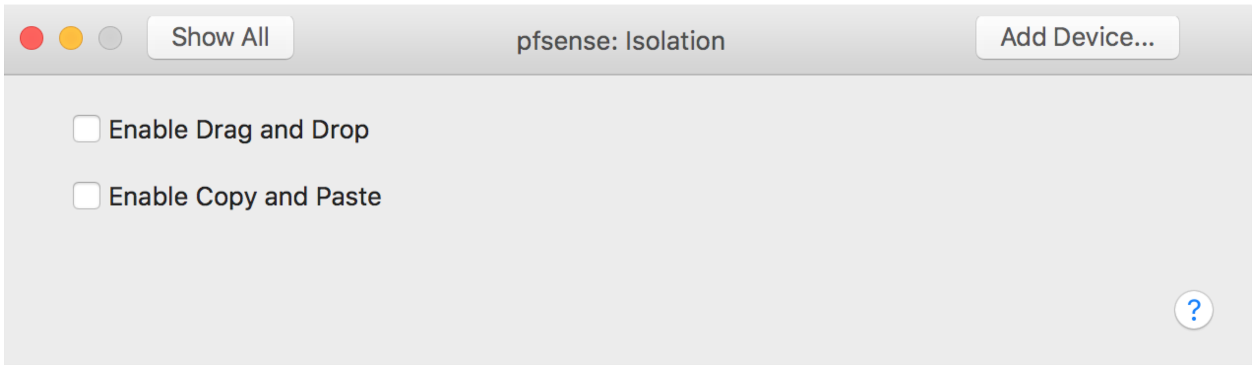
When a new USB device is plugged into your Mac while this virtual machine is running, VMware Fusion should:

Ask what to do

☐ Share Bluetooth devices with the virtual machine

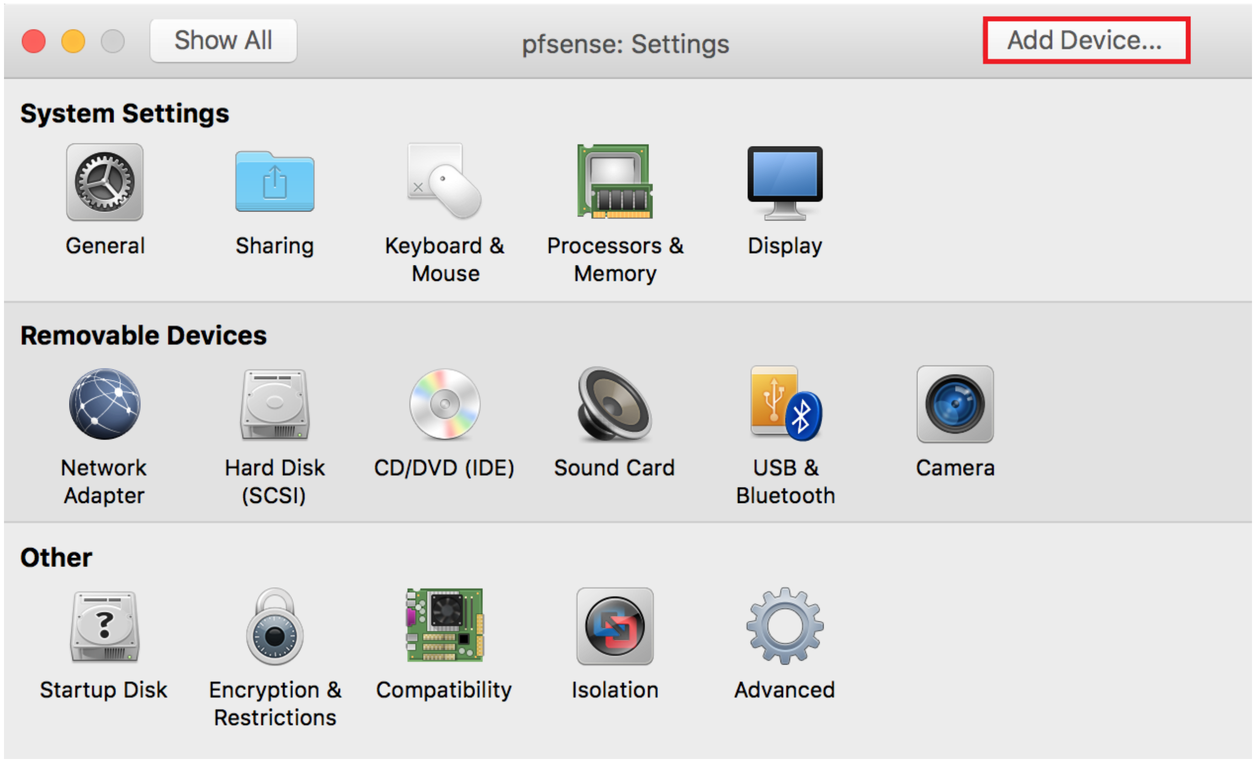
USB devices cannot be connected to the virtual machine until it is powered on.

In the main menu, under the *Other* section, click on *Isolation*. These settings govern what kind of interaction is allowed between the VM and OS X, and vice-versa. Since we want to maintain separation as much as we can, ensure that the checkboxes next to *Enable Drag and Drop* and *Enable Copy and Paste* are unchecked. Afterwards, click *Show All* to return to the main menu.

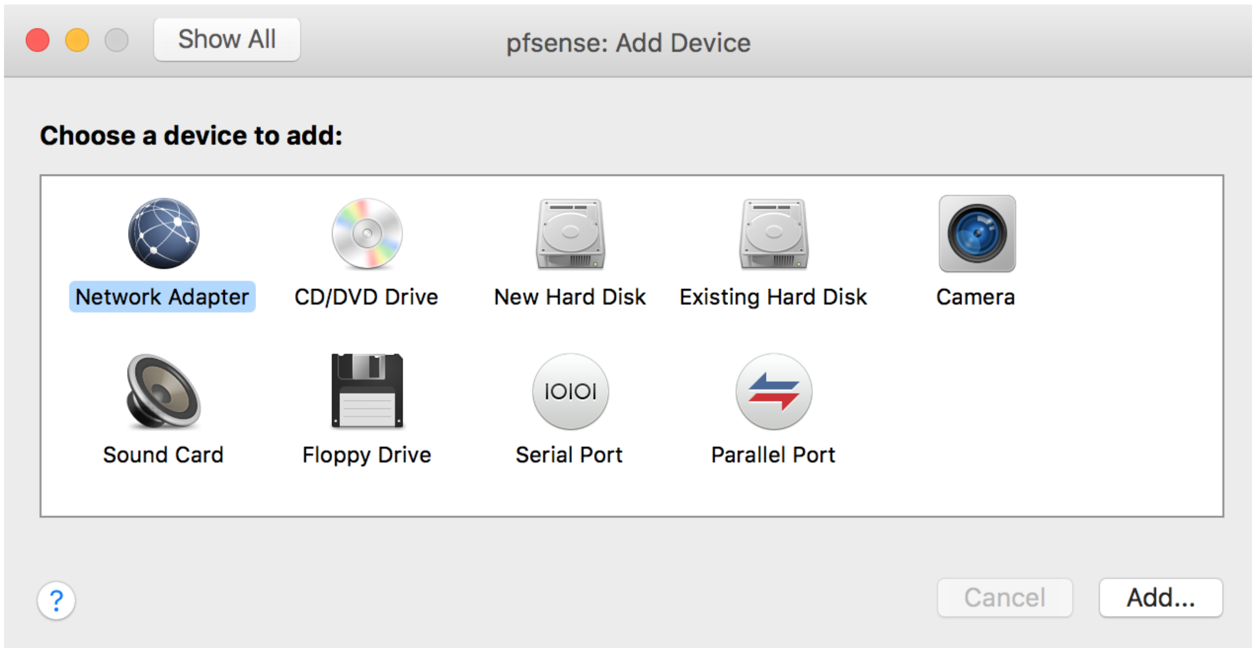


☐ Enable Drag and Drop
☐ Enable Copy and Paste

In the main menu, click the *Add Device...* button in the upper right corner.

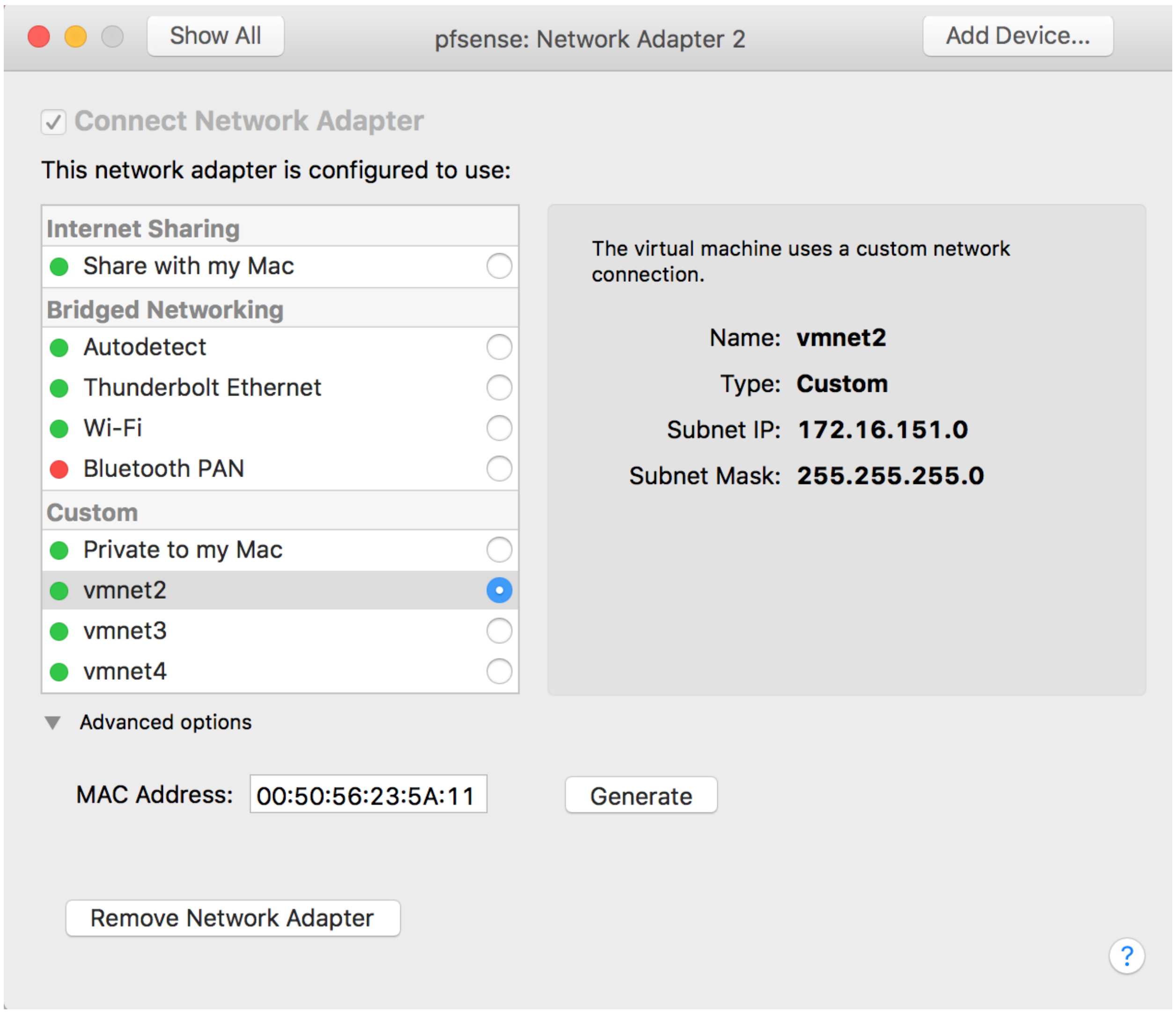


The next screen asks you choose a device to add. Click on *Network Adapter* to highlight it, then click the *Add…* button. This brings you to the configuration settings page for *Network Adapter 2*.



Ensure the *Connect Network Adapter* checkbox in the top left corner is checked. In the network list pane, select the *vmnet2* radio button under *Custom*. Click the triangle next to *Advanced options*, then click the *Generate* button. **Record this MAC address, and document that it is from the adapter that is attached to the *Management* Network.** After you are done, click on the *Add Device…* button in the upper right corner again.

Note: Do not worry about the *Subnet IP* and *Subnet Mask* settings that are displayed. These settings DO NOT apply, since we disabled Fusion's DHCP server for all the virtual networks we created for our lab environment. This also applies to the rest of the VMs we are going to create; do not worry about the displayed IP address or subnet mask in the network adapter settings.



✓ Connect Network Adapter

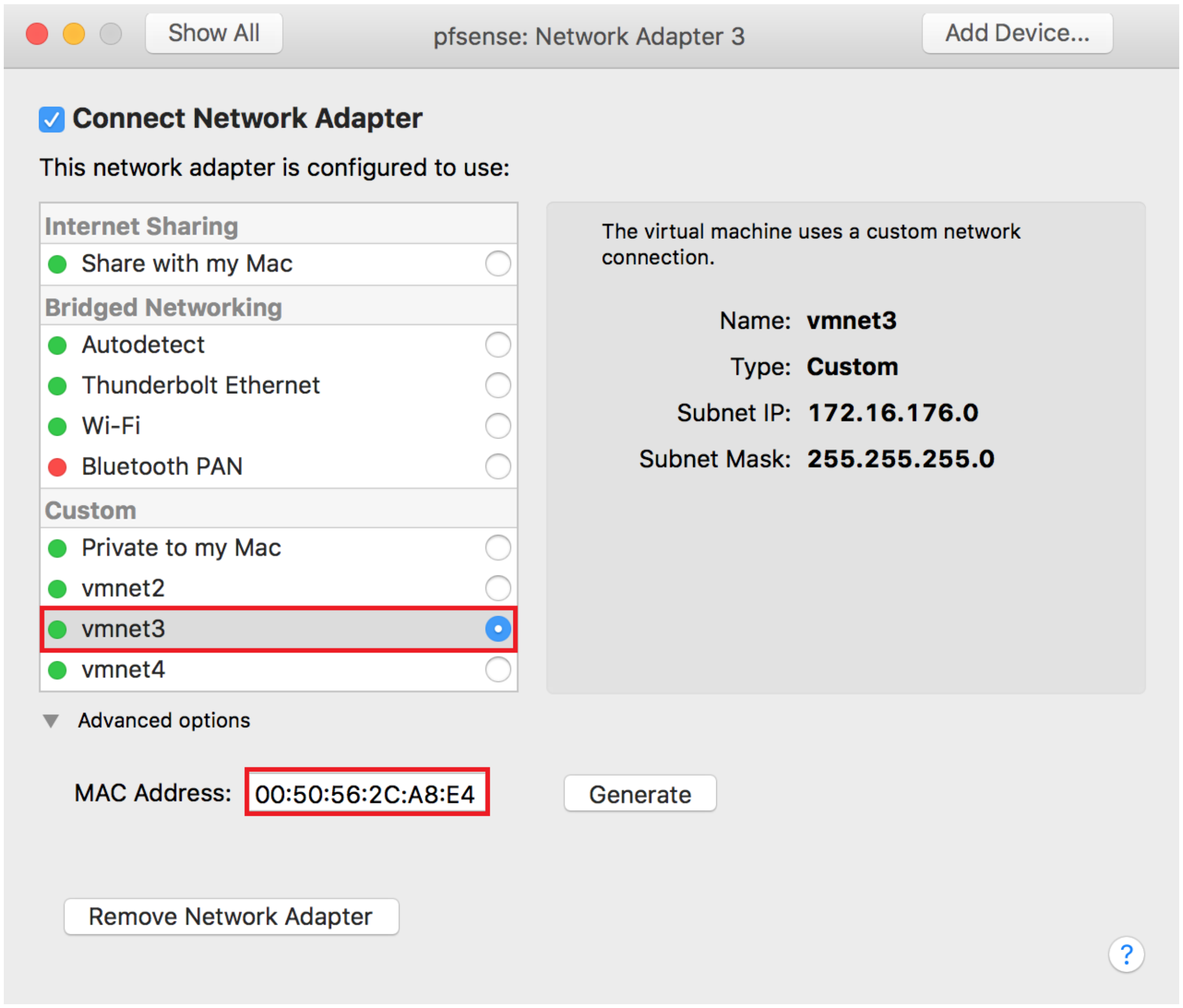
This network adapter is configured to use: vmnet2

The virtual machine uses a custom network connection.

Name: vmnet2
Type: Custom
Subnet IP: 172.16.151.0
Subnet Mask: 255.255.255.0

MAC Address: 00:50:56:23:5A:11

In the *Add Device* menu, click on *Network Adapter* again to highlight it, then click the *Add…* button. You should be in the settings page for *Network Adapter 3*. Perform the same steps you performed for *Network Adapter 2* but instead, connect this network adapter to *vmnet3*, **document the MAC address, along with a note that it belongs to the adapter that is attached to the *IPS 1* Network.** Click the *Show All* button to return to the main menu one more time, then click the red circle in the upper left corner of the settings menu to exit.



☑ Connect Network Adapter

This network adapter is configured to use: vmnet3

The virtual machine uses a custom network connection.

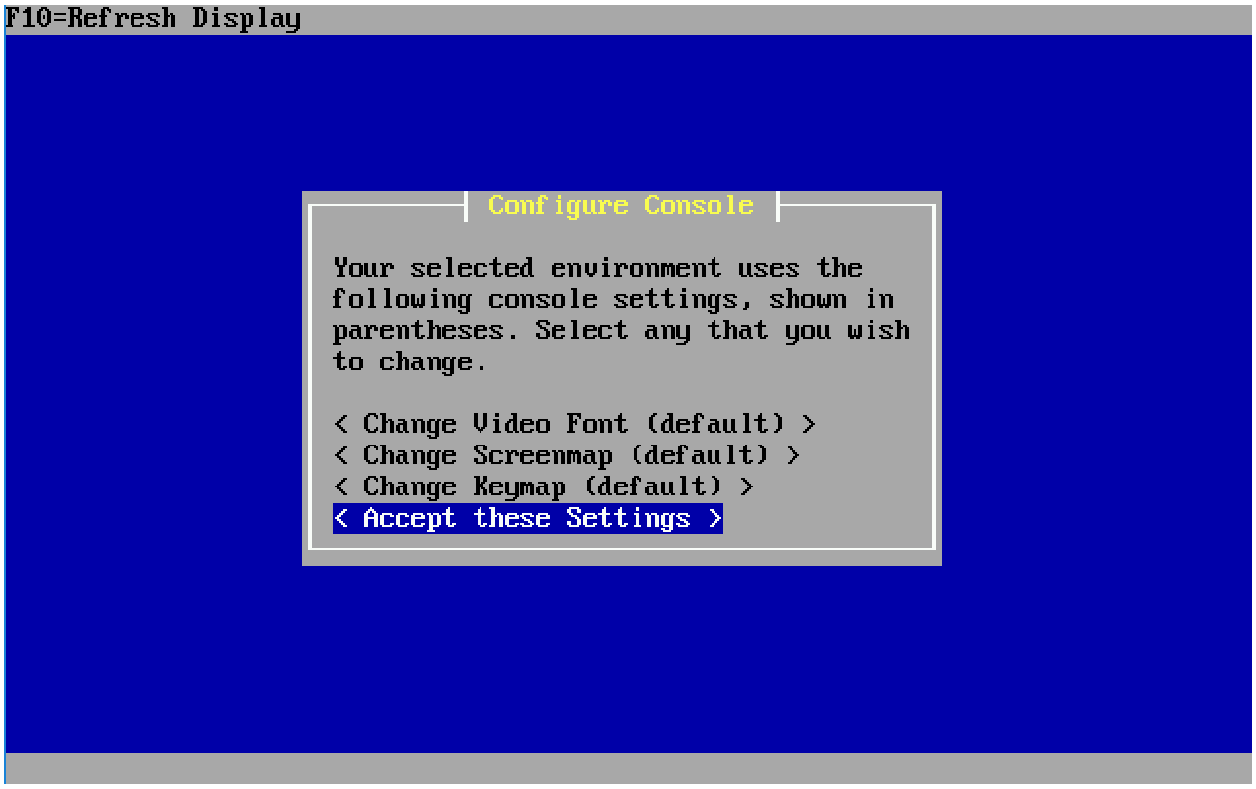
Name: vmnet3
Type: Custom
Subnet IP: 172.16.176.0
Subnet Mask: 255.255.255.0

MAC Address: 00:50:56:2C:A8:E4

Installing pfSense

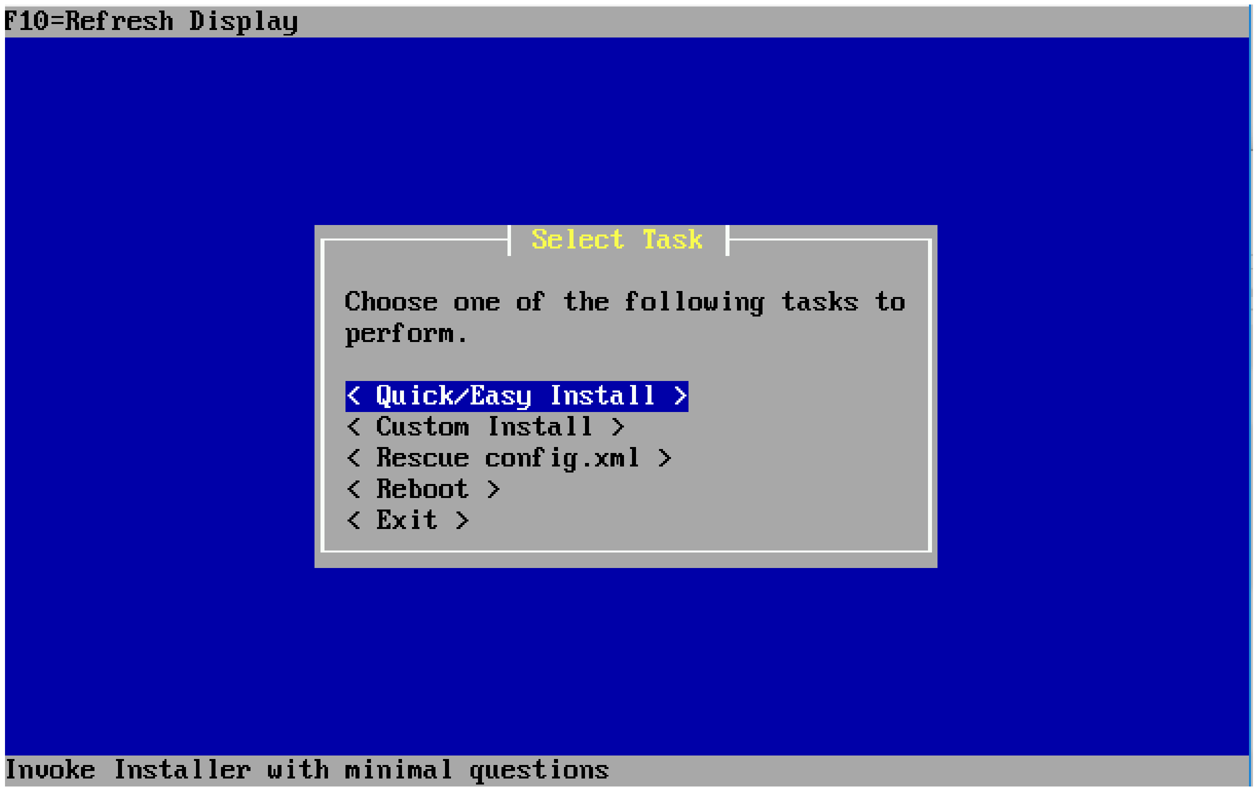
To start the pfSense VM, you can either click the big play button on the window entitled *pfSense*, or choose *Start Up* from the *Virtual Machine* menu in the menu bar.. After doing so, the primary window will switch to display the console of the powered on VM. By clicking within this console view, the host's mouse and keyboard become attached to the VM. To detach again, press ctrl+command.

Let pfSense boot up, and the system should automatically run the installer. Adjust your video, screenmap, and keymap settings as necessary, then select < *Accept these Settings* >.



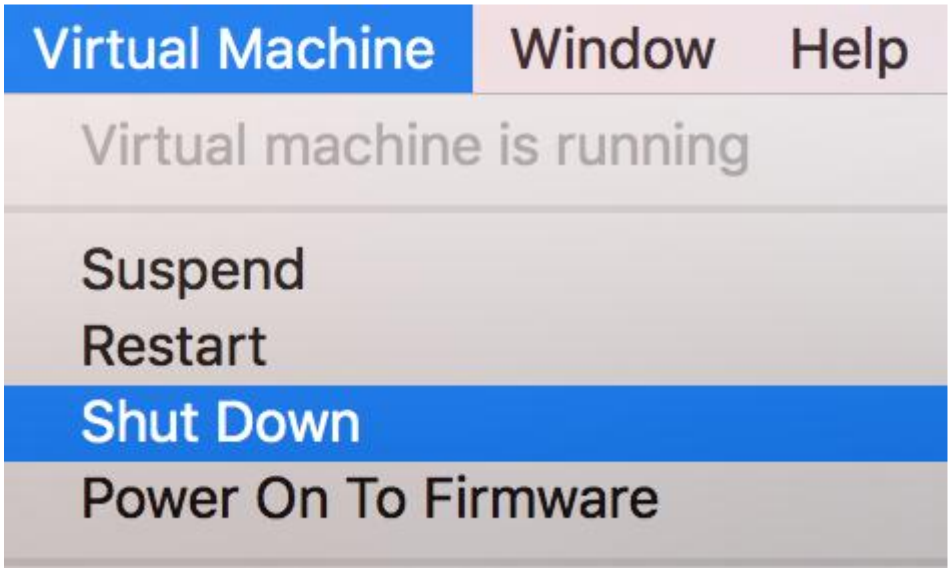
```
F10=Refresh Display

Configure Console

Your selected environment uses the
following console settings, shown in
parentheses. Select any that you wish
to change.

< Change Video Font (default) >
< Change Screenmap (default) >
< Change Keymap (default) >
< Accept these Settings >
```

On the next screen, select < *Quick/Easy Install* > and let pfSense do all the heavy lifting. The next screen will inform you that the install will erase the contents of the hard disk. Since our virtual disk is already empty, this doesn't matter in the least. Select OK. and let the install run.



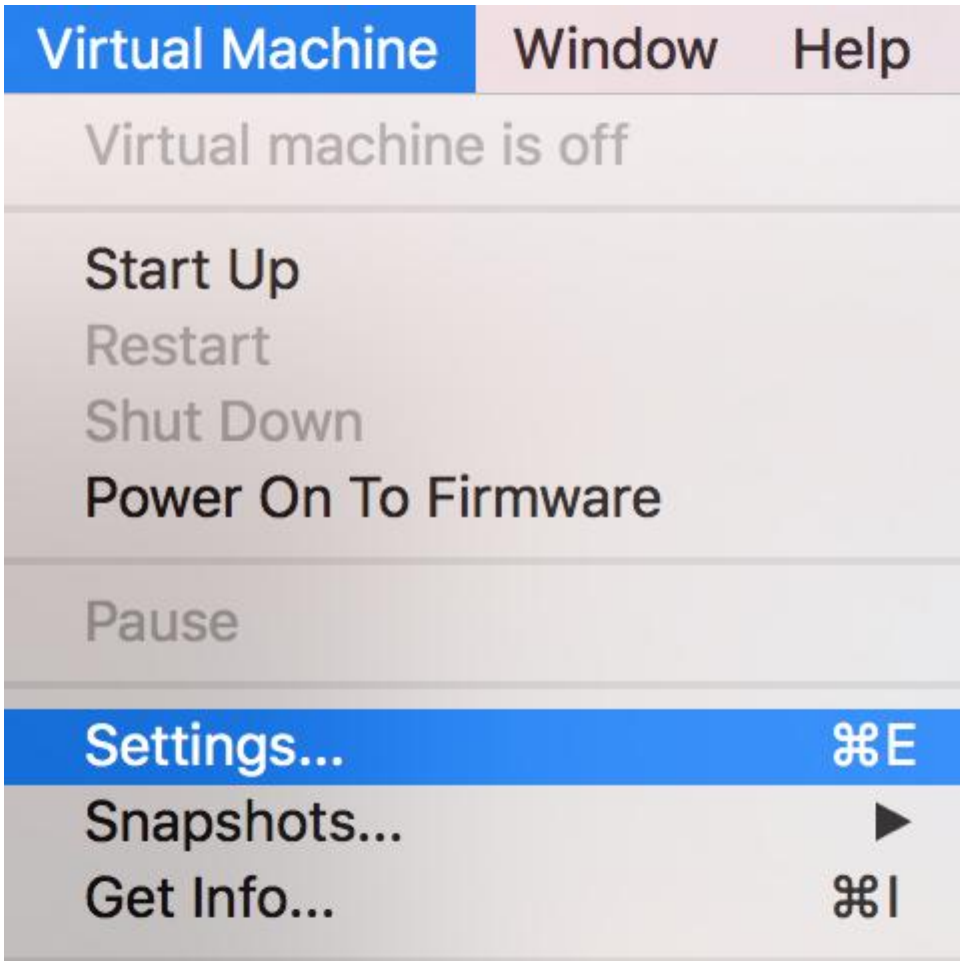
```
F10=Refresh Display

                 Select Task

        Choose one of the following tasks to
        perform.

        < Quick/Easy Install >
        < Custom Install >
        < Rescue config.xml >
        < Reboot >
        < Exit >

Invoke Installer with minimal questions
```

The installer will go through and install pfSense to the virtual hard disk. At a certain point you will be asked whether the standard kernel or the embedded one should be used. Make sure to select `< Standard Kernel >`. Finally, the installer informs you to reboot the machine to boot from the hard drive. The installation is done, however, we're not going to reboot. Use ctrl+command to exit the VM window, then click on *Virtual Machine* in the menu bar, and select *Shut Down*.



Final VM Settings

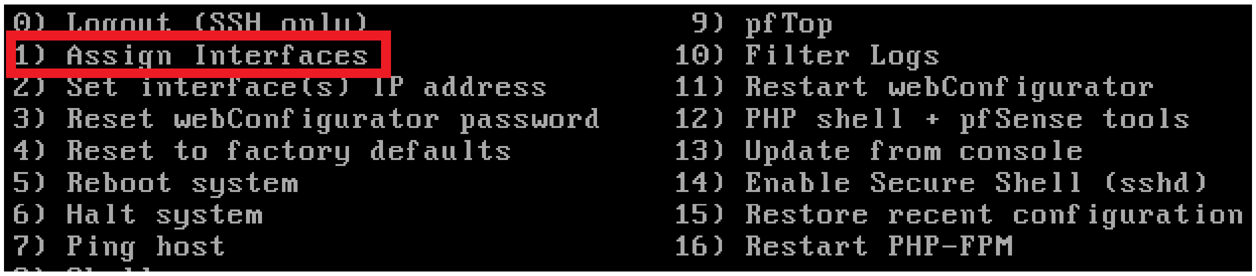
There is one last configuration change we want to make to the pfSense VM before we boot it up to finish configuring it. Make sure the virtual machine is powered off, then click on *Virtual Machine* on the menu bar, then select *Settings…*to enter the Settings menu for the pfsense VM.



From the main menu, click *CD/DVD (IDE)* to enter the configuration menu for the CD/DVD drive. Since pfSense is installed, we no longer need the drive anymore. Click the triangle next to *Advanced options*, then click on the *Remove CD/DVD Drive* button. Click on the *Remove* button in the dialogue box to confirm your choice. Afterwards, you should be returned the main menu. Exit the settings menu by clicking the red circle in the upper left corner of the window.

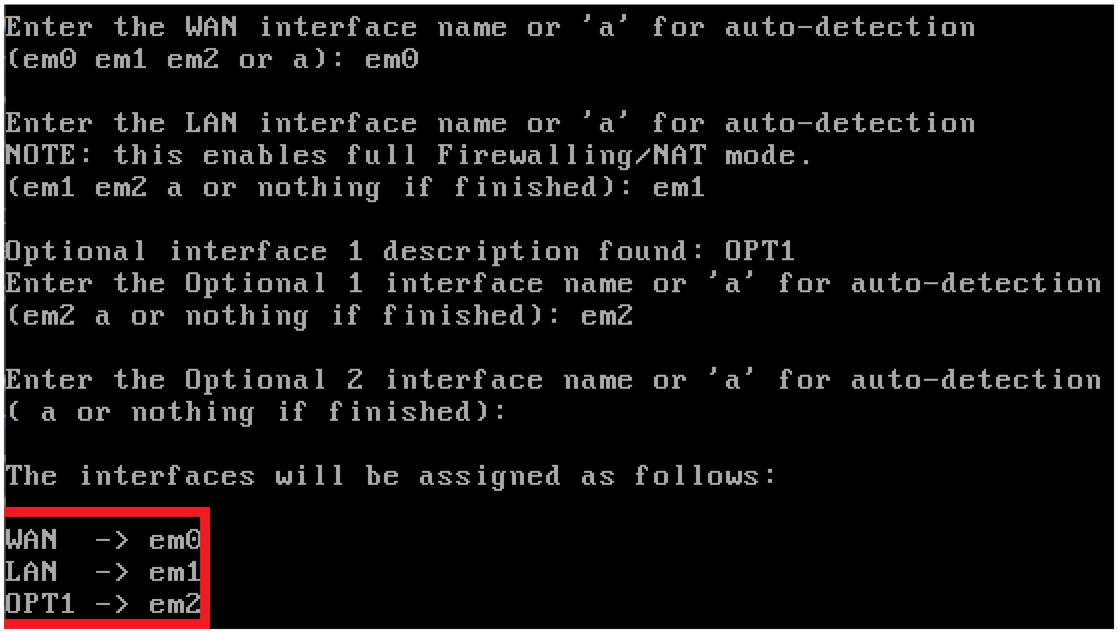
Network Configuration

Power on the pfSense VM, and allow it to boot up. Eventually you'll be greeted with the main pfSense menu on the console with 16 options. Select option 1, *Assign Interfaces* to run the interface assignment wizard. To manage its network services and firewall policies, pfSense uses internal network interface descriptors (*WAN, LAN, OPT1*, etc.). Running the wizard and assigning the interfaces defines which of them *(em0, em1, em2*, etc.) corresponds to which of pfSense's internal descriptors.



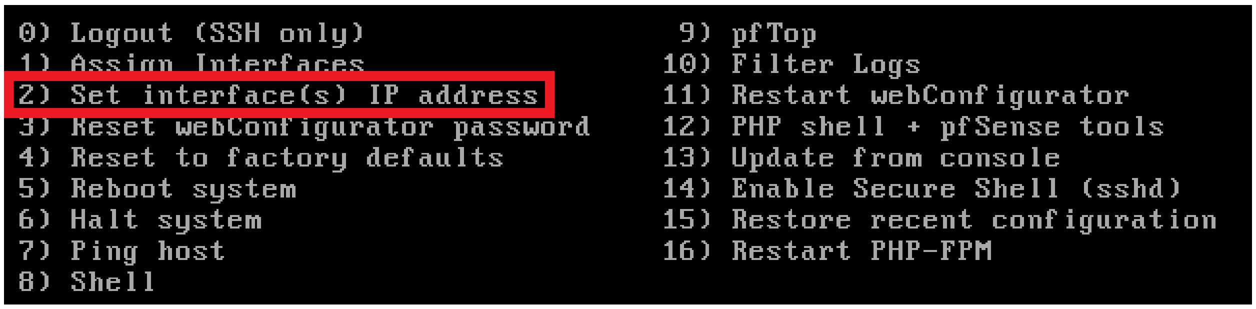
```
 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
```

The *WAN* interface needs to be assigned to the virtual interface bridged to the host (the *Bridged* network). The *LAN* interface needs to be assigned to virtual interface connected to *vmnet2* (the *Management* network), and finally, *OPT1* needs to be assigned to the network adapter attached to *vmnet3* (the *IPS 1* network). In my case, *em0* became the WAN interface, *em1* became the *LAN* interface, and *em2* became the *OPT1* interface. **Pay close attention to the MAC addresses displayed for *em0*, *em1*, and *em2*, and correlate them to those to the MAC addresses you documented earlier for *Network Adapter, Network Adapter 2*, and *Network Adapter 3*.** After confirming that the settings are correct, you should automatically be returned to the pfSense main menu.



```
Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): em0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): em1

Optional interface 1 description found: OPT1
Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished): em2

Enter the Optional 2 interface name or 'a' for auto-detection
( a or nothing if finished):

The interfaces will be assigned as follows:

WAN  -> em0
LAN  -> em1
OPT1 -> em2
```

Now that we have assigned the network interfaces, we have to configure their IP addresses. In most cases, the *WAN* interface will automatically get an IP address from the device on your physical network that provides DHCP services. If this is not happening, you may have to troubleshoot your physical network. This is beyond the scope of our guide. As for the *LAN* and *OPT1* networks, we have to manually set the IP address, subnet mask, and DHCP scopes for these networks. Select option 2 in the pfSense main menu to get started.



```
 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell
```

The configuration wizard starts by asking for the interface the settings should be changed for. We have to go through the process two times, for the LAN and the OPT1 interface individually. Here are the settings I recommend:

LAN IP: **172.16.1.1**
LAN Subnet bit count: **24**
LAN upstream gateway address: **<empty>**
LAN IPv6 address: **<empty>**
Do you want to enable the DHCP server on LAN? **y**
LAN DHCP start address: **172.16.1.10**
LAN DHCP end address: **172.16.1.254**
Do you want to revert to HTTP as the webConfigurator protocol? **N**

OPT1 IP: **172.16.2.1**
OPT1 Subnet bit count: **24**
OPT1 upstream gateway address: **<empty>**
OPT1 IPv6 address: **<empty>**
Do you want to enable the DHCP server on OPT1? **y**
OPT1 DHCP start address: **172.16.2.10**
OPT1 DHCP end address: **172.16.2.254**
Do you want to revert to HTTP as the webConfigurator protocol? **n**

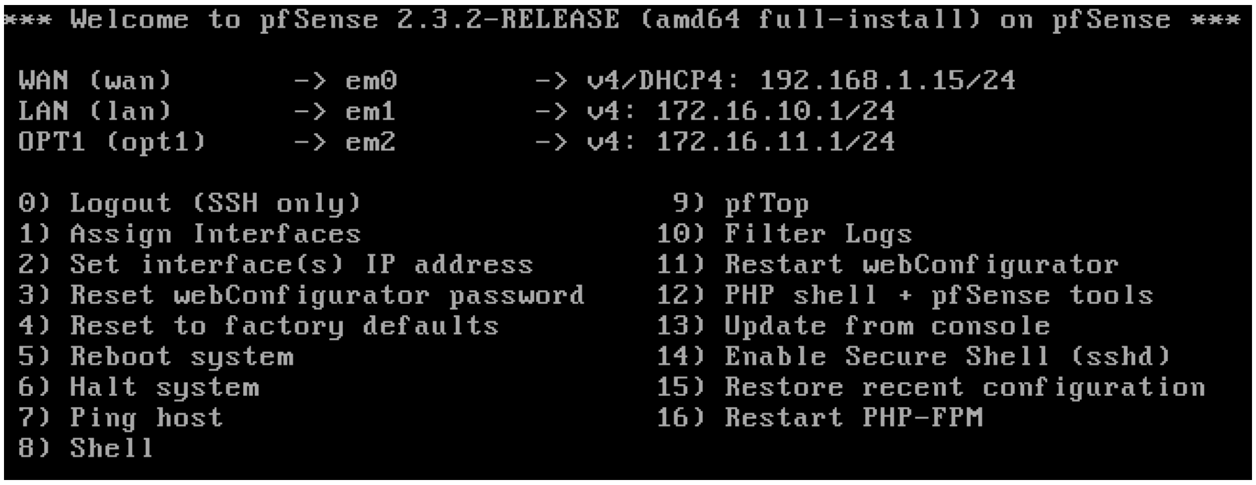
The reason we choose 1.10 and 2.10 as the DHCP start addresses is to reserve a few IP addresses for static DHCP allocations. Static DHCP allocations allow you to configure a DHCP server to always serve the same IP address when requested from a particular MAC address. We will talk about this a little bit more later. For now, we are done mucking around in the pfSense CLI. From here on out, we will be using the webConfigurator to manage and configure our pfSense VM.

Note: If you are using 172.16.1.0/24 or the 172.16.2.0/24 network ranges in your physical network, choose another network range to assign to the LAN and OPT1 interfaces to avoid network conflicts.

Note: When you are configuring the LAN and OPT1 interface IP addresses, the configuration script will ask “Do you want to revert to HTTP as the webConfigurator protocol?” **Always say no**

to this. This allows the web UI to accept HTTP logins, and makes your firewall credentials vulnerable to sniffing over the network.

Your pfSense main menu should look something like this when you're all done:



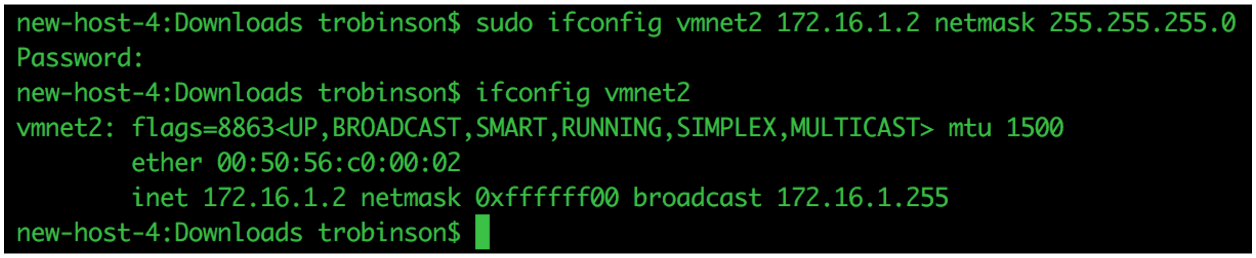
```
*** Welcome to pfSense 2.3.2-RELEASE (amd64 full-install) on pfSense ***

 WAN (wan)       -> em0        -> v4/DHCP4: 192.168.1.15/24
 LAN (lan)       -> em1        -> v4: 172.16.10.1/24
 OPT1 (opt1)     -> em2        -> v4: 172.16.11.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell
```

Web Configurator - Initial Setup

Before you are able to log in to the web interface for pfSense, you must configure the Host-Only adapter vmnet2 (aka the management network) on your host. Open up the terminal application for your mac and enter the command:

```
sudo ifconfig vmnet2 172.16.1.2 netmask 255.255.255.0
```

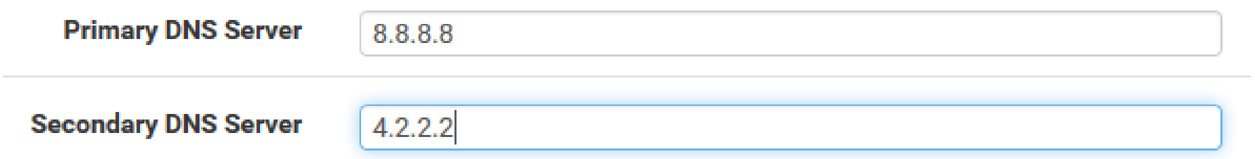
This command will require you to enter your password. This sets the vmnet2 adapter on your macbook to use the IP address 172.16.1.2 with a /24 subnet mask. Run the command `ifconfig vmnet2` to verify that the “inet” field reads 172.16.1.2, and the “broadcast” field reads 172.16.1.255. **Please be aware that using `ifconfig` to set a static IP address does not persist across system reboots or even VMware Fusion being shut down**. Every time VMware Fusion is shut down (e.g. Command+Q) or the system is shut down/rebooted, the vmnet interfaces are deleted and recreated with their default values. If VMware Fusion is restarted, or the system is rebooted/restarted, simply re-run the `ifconfig` command above to give your hypervisor host direct access to the *Management* virtual network (*vmnet2*).



```
new-host-4:Downloads trobinson$ sudo ifconfig vmnet2 172.16.1.2 netmask 255.255.255.0
Password:
new-host-4:Downloads trobinson$ ifconfig vmnet2
vmnet2: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 00:50:56:c0:00:02
        inet 172.16.1.2 netmask 0xffffff00 broadcast 172.16.1.255
new-host-4:Downloads trobinson$
```

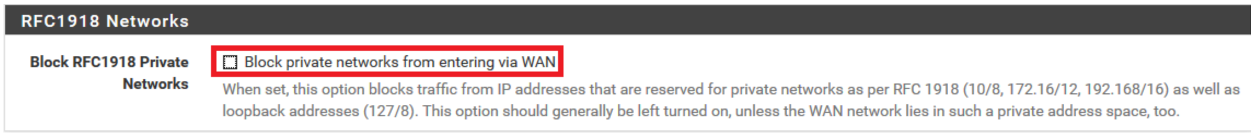
Once you have properly configured the *vmnet2* interface, open your favorite web browser (I prefer Firefox - https://www.mozilla.org/en-US/firefox/new/) and navigate to https://172.16.1.1 (or the IP address you assigned to the LAN interface of your pfSense system). **The default credentials for access to the web interface are admin/pfsense.** If this is your first time logging in, pfSense takes you through a nice little setup wizard. I’ll highlight some of the important things to take note of, and/or change as necessary:

Set the primary and secondary DNS servers you plan on using. I typically use 8.8.8.8 (Google public DNS) and 4.2.2.2 (Level 3 public DNS). If you're using this lab at work, your workplace may have restrictions and you may have to use their DNS servers instead.


Primary DNS Server: 8.8.8.8

Secondary DNS Server: 4.2.2.2

Uncheck the checkbox next to “Block private networks from entering via WAN” rule. Uncheck the checkbox next to “Block Bogon Networks” on this page as well.

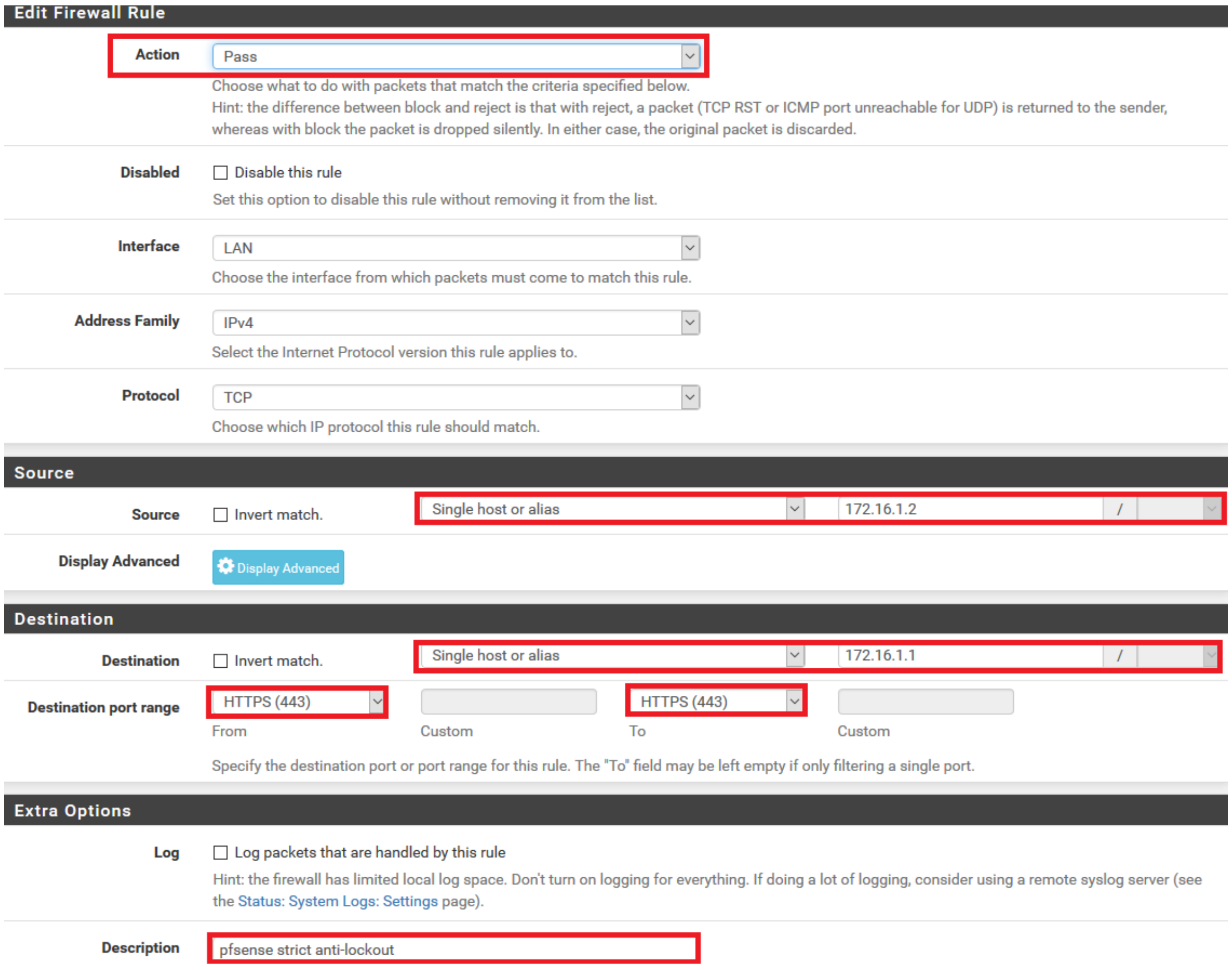

RFC1918 Networks

Block RFC1918 Private Networks: ☐ Block private networks from entering via WAN

When set, this option blocks traffic from IP addresses that are reserved for private networks as per RFC 1918 (10/8, 172.16/12, 192.168/16) as well as loopback addresses (127/8). This option should generally be left turned on, unless the WAN network lies in such a private address space, too.

The other default settings on the first time setup wizard should be fine. Please note that you will be asked to change the password for the admin account as a part of the setup wizard. Document the password, and keep it somewhere safe.

Our next order of business is to restrict access to the web interface of the firewall to the machine with the IP address 172.16.1.2. On the menu bar at the top of the page, select *Firewall > Rules*. Click on *LAN* to modify the firewall policy for the *LAN* interface. Click the *Add* button with the arrow facing up. This will add the firewall rule to the top of the firewall policy to where it will evaluated first. The *Edit Firewall Rule* page is fairly straightforward. I've highlighted the options to be aware of.



Edit Firewall Rule

Action: Pass

Choose what to do with packets that match the criteria specified below.
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.

Disabled: ☐ Disable this rule

Set this option to disable this rule without removing it from the list.

Interface: LAN

Choose the interface from which packets must come to match this rule.

Address Family: IPv4

Select the Internet Protocol version this rule applies to.

Protocol: TCP

Choose which IP protocol this rule should match.

Source

Source: ☐ Invert match. Single host or alias 172.16.1.2 /

Display Advanced: Display Advanced

Destination

Destination: ☐ Invert match. Single host or alias 172.16.1.1 /

Destination port range: From: HTTPS (443) Custom: To: HTTPS (443) Custom:

Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.

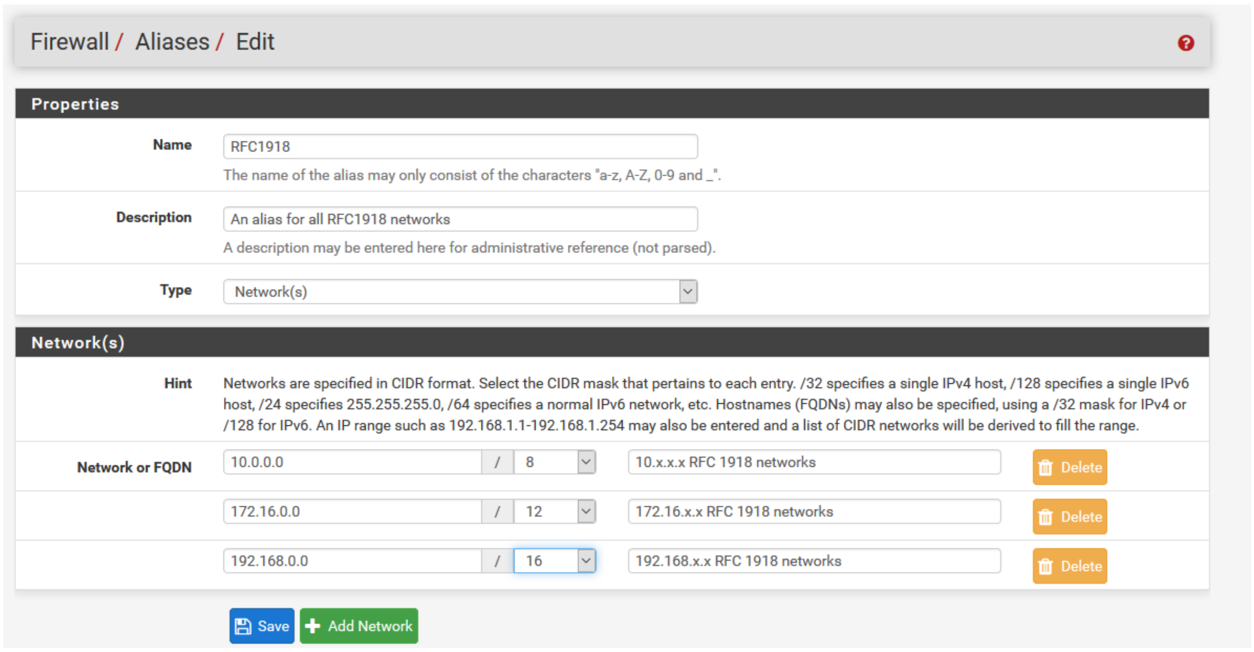
Extra Options

Log: ☐ Log packets that are handled by this rule

Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the Status: System Logs: Settings page).

Description: pfsense strict anti-lockout

When you are done, click the Save icon at the bottom of the page. This will take you back to the previous page. A yellow dialogue box with a green button called “Apply Changes” will appear. Click this button to apply this new firewall rule. Next, we want to an alias. Navigate to *Firewall > Aliases*. On the Firewall Aliases page, click on *IP*, then click *Add*. Create an alias with the following settings:


Firewall / Aliases / Edit

Properties

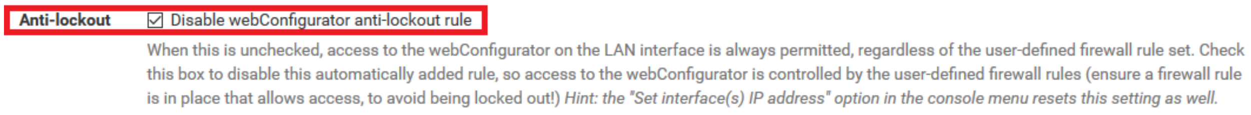
Name	RFC1918 The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".
Description	An alias for all RFC1918 networks A description may be entered here for administrative reference (not parsed).
Type	Network(s)

Network(s)

Hint: Networks are specified in CIDR format. Select the CIDR mask that pertains to each entry. /32 specifies a single IPv4 host, /128 specifies a single IPv6 host, /24 specifies 255.255.255.0, /64 specifies a normal IPv6 network, etc. Hostnames (FQDNs) may also be specified, using a /32 mask for IPv4 or /128 for IPv6. An IP range such as 192.168.1.1-192.168.1.254 may also be entered and a list of CIDR networks will be derived to fill the range.

Network or FQDN	/	Description
10.0.0.0	8	10.x.x.x RFC 1918 networks
172.16.0.0	12	172.16.x.x RFC 1918 networks
192.168.0.0	16	192.168.x.x RFC 1918 networks

Click Save to be brought back to the previous page, then click *Apply Changes*. We just created an alias for RFC1918 networks (local networks that are not routable through the public internet) This will come in handy later for creating firewall rules around these networks. Our final action for now is to navigate to *System > Advanced*, and click to fill in the checkbox next to the option *Disable webConfigurator anti-lockout rule*, and click *Save* on the bottom of the page. **Be aware that if you did not create the firewall rule above, this <u>will</u> lock you out of the webConfigurator as soon as you click *Save*.**

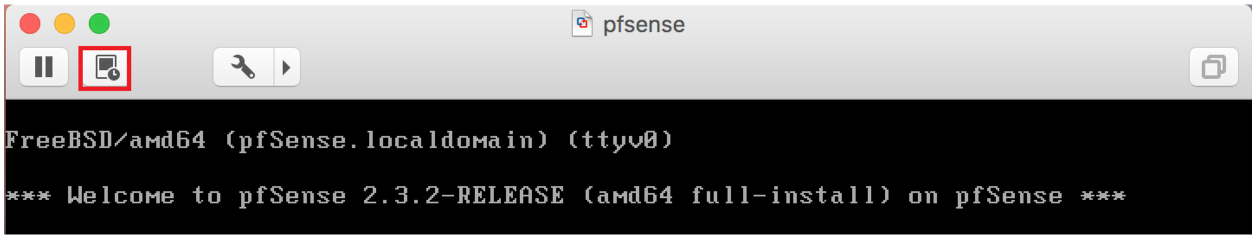

Anti-lockout ☑ Disable webConfigurator anti-lockout rule

When this is unchecked, access to the webConfigurator on the LAN interface is always permitted, regardless of the user-defined firewall rule set. Check this box to disable this automatically added rule, so access to the webConfigurator is controlled by the user-defined firewall rules (ensure a firewall rule is in place that allows access, to avoid being locked out!) *Hint: the "Set interface(s) IP address" option in the console menu resets this setting as well.*

Take a Snapshot

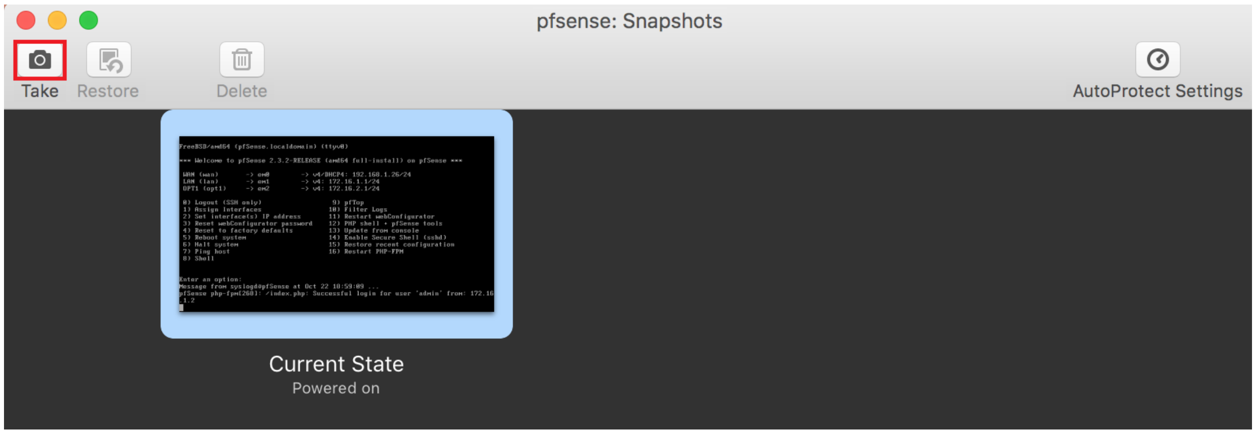
Snapshotting is a VERY important concept with VMs. Snapshots let you restore your virtual machines to a point in time in the past when you took that snapshot. This lets you undo misconfigurations, problems, or potential issues that affect the VM relatively easily. We’re going to take a snapshot of pfSense in its current state.

In the pfSense window, click on the icon to the right of the “pause” button. This will open the snapshot manager window.



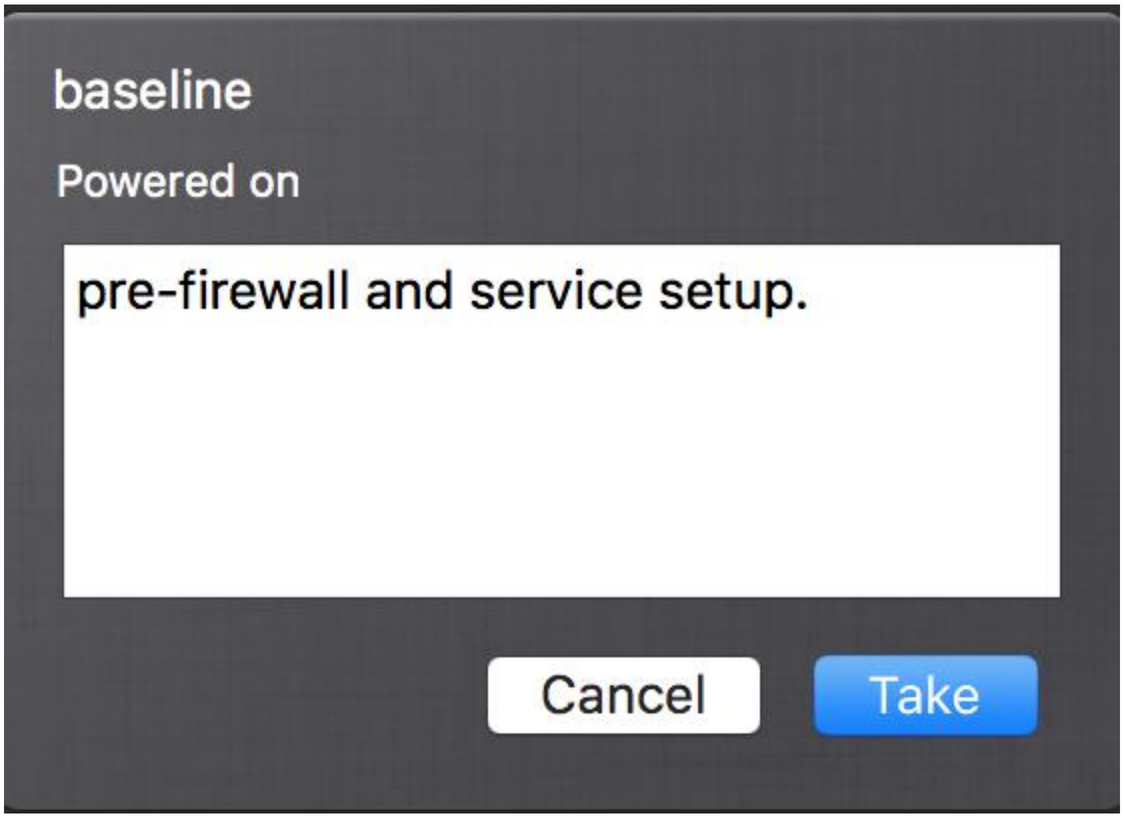
```
FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

*** Welcome to pfSense 2.3.2-RELEASE (amd64 full-install) on pfSense ***
```

Once in the snapshot manager, click the icon that looks like a camera labeled “Take”.



```
FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

*** Welcome to pfSense 2.3.2-RELEASE (amd64 full-install) on pfSense ***

 WAN (wan)       -> em0       -> v4/DHCP4: 192.168.1.26/24
 LAN (lan)       -> em1       -> v4: 172.16.1.1/24
 OPT1 (opt1)     -> em2       -> v4: 172.16.2.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell


Enter an option:
Message from syslogd@pfSense at Oct 22 10:59:09 ...
pfSense php-fpm[268]: /index.php: Successful login for user 'admin' from: 172.16
.1.2
```

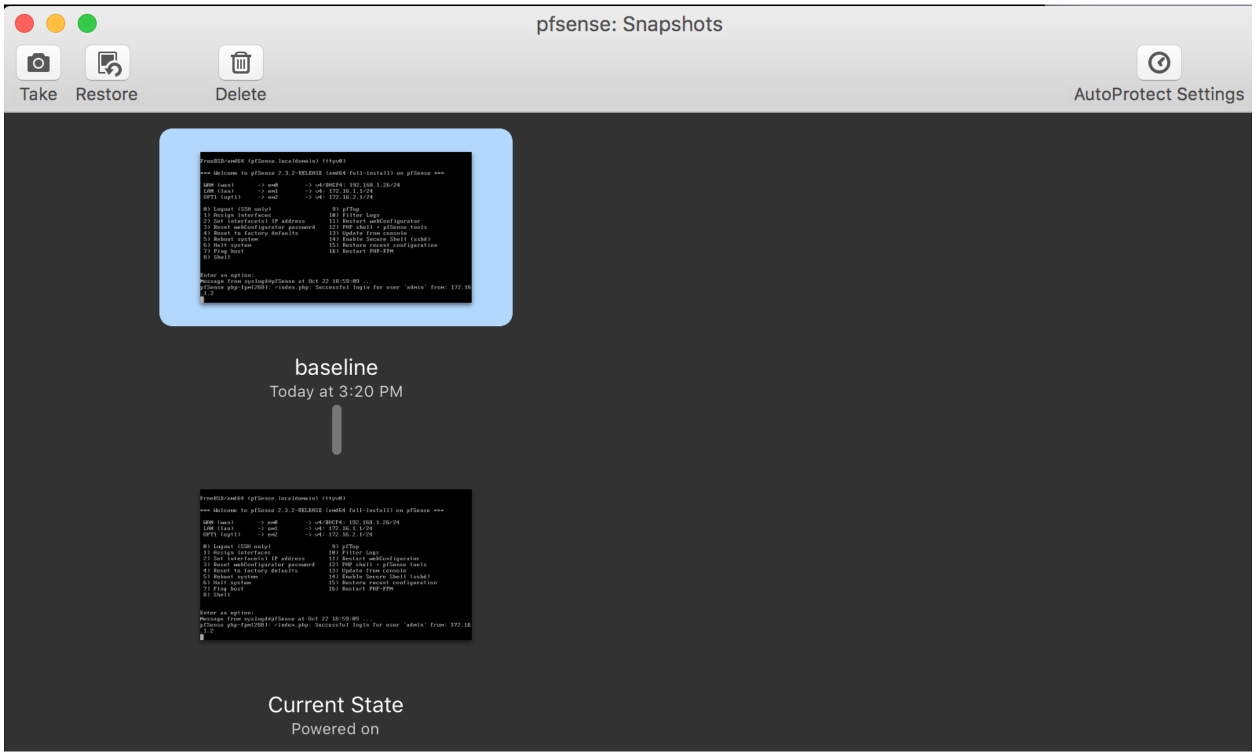
A window pops up asking you to name the snapshot, and add notes. I named mine “baseline” with the notes “pre-firewall and service setup”. If you keep multiple snapshots make sure to use good descriptions that include <u>WHEN</u> the snapshot was taken, and <u>WHAT STATE</u> the VM is in when you took the snapshot. After you are done, click the “Take” button for VMware Fusion to generate a snapshot of the VM.


baseline

Powered on

pre-firewall and service setup.

This snapshot will appear in the snapshot manager. You can click on it, and a small “i” will pop up if you hover over it. If you click the “i”, then the name and snapshot notes will appear. If you ever need to restore to a previous snapshot, you can click to highlight the snapshot, then click “Restore” on the top of the snapshot manager window. When you are finished, click the red circle in the upper left corner of the window.



```
FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

*** Welcome to pfSense 2.3.2-RELEASE (amd64 full-install) on pfSense ***

 WAN (wan)       -> em0       -> v4/DHCP4: 192.168.1.26/24
 LAN (lan)       -> em1       -> v4: 172.16.1.1/24
 OPT1 (opt1)     -> em2       -> v4: 172.16.2.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell


Enter an option:
Message from syslogd@pfSense at Oct 22 10:59:09 ...
pfSense php-fpm[268]: /index.php: Successful login for user 'admin' from: 172.16
.1.2
```

baseline
Today at 3:20 PM

```
FreeBSD/amd64 (pfSense.localdomain) (ttyv0)

*** Welcome to pfSense 2.3.2-RELEASE (amd64 full-install) on pfSense ***

 WAN (wan)       -> em0       -> v4/DHCP4: 192.168.1.26/24
 LAN (lan)       -> em1       -> v4: 172.16.1.1/24
 OPT1 (opt1)     -> em2       -> v4: 172.16.2.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell


Enter an option:
Message from syslogd@pfSense at Oct 22 10:59:09 ...
pfSense php-fpm[268]: /index.php: Successful login for user 'admin' from: 172.16
.1.2
```

Current State
Powered on

pfSense Summary

At the end of all this, you should have a pfSense VM with the following settings:

- 512MB of RAM
- 5GB of Disk
- 1 Virtual CPU
- DVD drive disabled
- Audio disabled
- USB controller disabled
- 3 Network interfaces:
    - 1 Interface connected to the Autodetect/*Bridged* network (*WAN* interface)
    - 1 Interface connected to the *vmnet2* (*Management*) network (*LAN* interface)
    - 1 Interface connected to the *vmnet3* (*IPS 1*) network (*OPT1* interface)

pfSense should be configured as followed:

- The WAN interface should be the VirtualBox network adapter connected to the Bridged Adapter. This interface should automatically get an IP address via DHCP. If this isn't happening, get some help with whoever administers your local network, or start troubleshooting. **Alternatively, configure a static IP address**
- The LAN interface should have an IP address of 172.16.1.1.
- The LAN network should be 172.16.1.0 with a /24 (255.255.255.0) subnet mask
- DHCP should be configured for this network with a scope of 172.16.1.10-172.16.1.254
    - 172.16.1.3-172.16.1.9 are available for setting static DHCP mappings
- The OPT1 interface should have an IP address of 172.16.2.1
- The OPT1 network should be 172.16.2.0 with a /24 (255.255.255.0) subnet mask
- DHCP should be configured for this network with a scope of 172.16.2.10-172.16.2.254
    - 172.16.2.2-172.16.2.9 are available for static DHCP mappings.
- 172.16.1.2 should have a firewall rule configured to allow it access to the firewall over HTTPS; this is the anti-lockout firewall rule.
- You should have an alias for RFC1918 networks configured for use with the firewall policy.

You should have at least one good snapshot you can revert to if you need to redo a step or somehow got lost/confused.

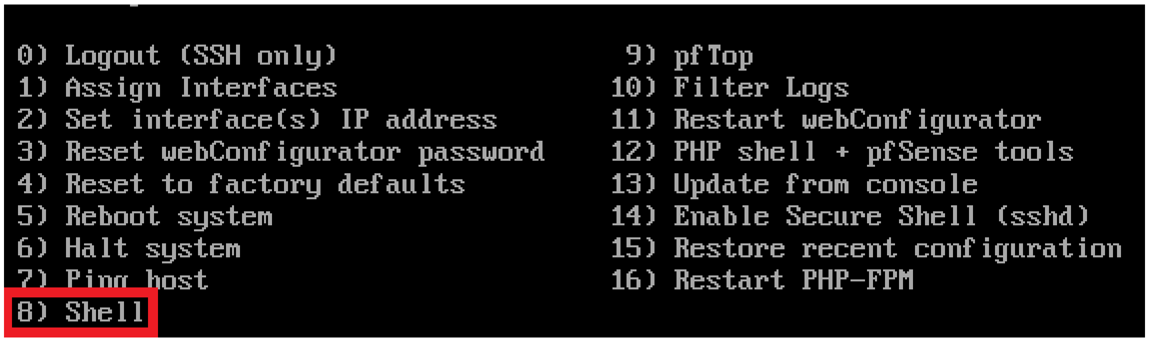
•

What's Next?

Now that you have an operational pfSense VM, your next steps are to read the pfSense Firewall Rules and Network Services Guide. Make sure the firewall policies match those illustrated in the segmentation guide, and make sure that the pfSense VM is hosting all the services mentioned in the Core Network Services guide to make your firewall fully functional and ready to handle your lab network.

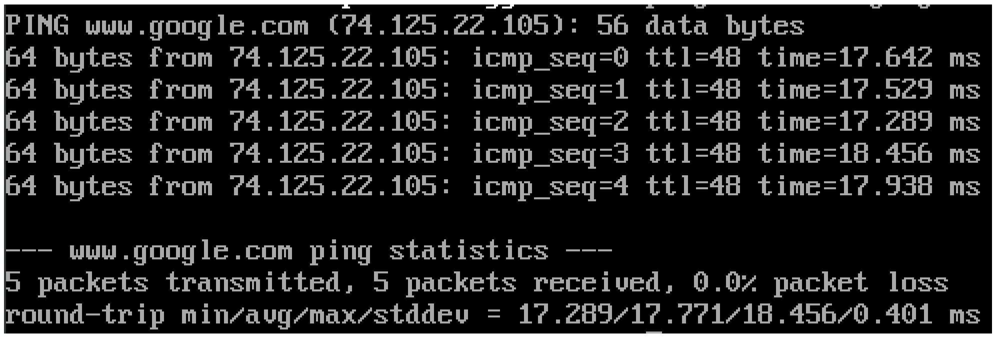
Final Connectivity Checks and Troubleshooting

Once you have finished setting up your pfSense VM according to the instructions laid out, Open a console session to the pfSense VM, and attach your mouse and keyboard by clicking on the console. If the screen is blank, hit the *Enter* key to bring up the pfSense main menu. Select option *8) Shell* to start a shell session on pfSense.



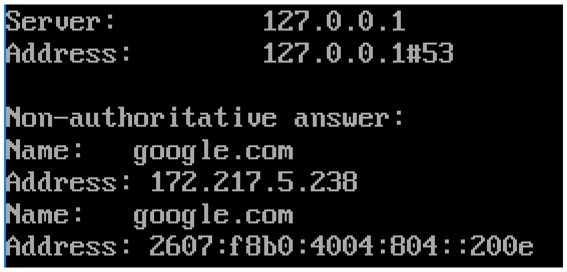
```
0) Logout (SSH only)                    9) pfTop
1) Assign Interfaces                   10) Filter Logs
2) Set interface(s) IP address         11) Restart webConfigurator
3) Reset webConfigurator password      12) PHP shell + pfSense tools
4) Reset to factory defaults           13) Update from console
5) Reboot system                       14) Enable Secure Shell (sshd)
6) Halt system                         15) Restore recent configuration
7) Ping host                           16) Restart PHP-FPM
8) Shell
```

The first, run the command `ping -c 5 www.google.com`



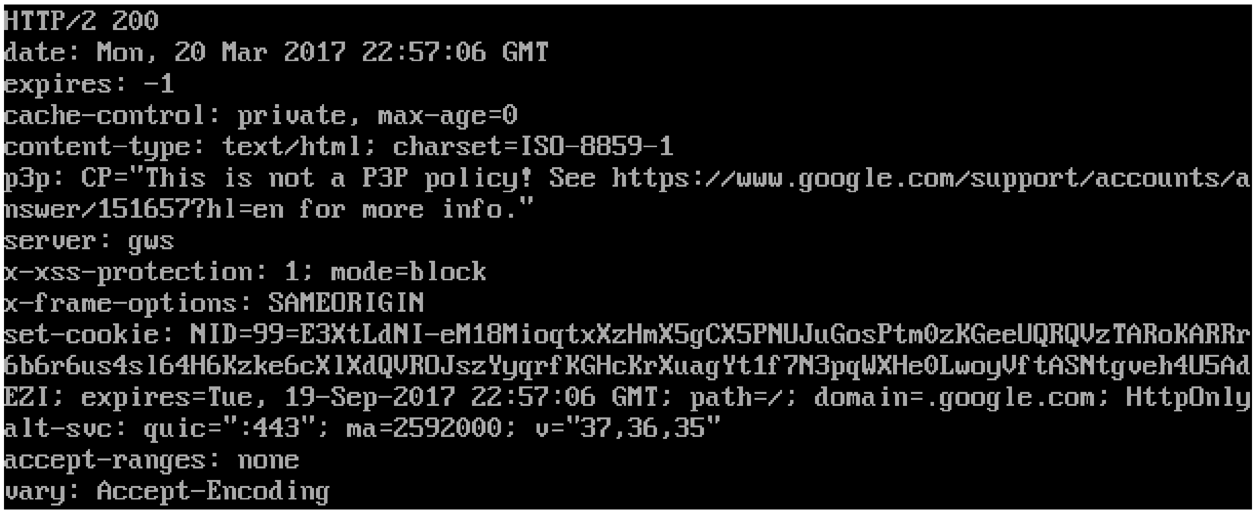
```
PING www.google.com (74.125.22.105): 56 data bytes
64 bytes from 74.125.22.105: icmp_seq=0 ttl=48 time=17.642 ms
64 bytes from 74.125.22.105: icmp_seq=1 ttl=48 time=17.529 ms
64 bytes from 74.125.22.105: icmp_seq=2 ttl=48 time=17.289 ms
64 bytes from 74.125.22.105: icmp_seq=3 ttl=48 time=18.456 ms
64 bytes from 74.125.22.105: icmp_seq=4 ttl=48 time=17.938 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 17.289/17.771/18.456/0.401 ms
```

The output from the `ping` command should look similar to the output above. Next, run the command `nslookup google.com`



```
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.5.238
Name:   google.com
Address: 2607:f8b0:4004:804::200e
```

Again, you should have output similar to the illustration above. Finally, run `curl -I https://www.google.com`



```
HTTP/2 200
date: Mon, 20 Mar 2017 22:57:06 GMT
expires: -1
cache-control: private, max-age=0
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See https://www.google.com/support/accounts/a
nswer/151657?hl=en for more info."
server: gws
x-xss-protection: 1; mode=block
x-frame-options: SAMEORIGIN
set-cookie: NID=99=E3XtLdNI-eM18MioqtxXzHmX5gCX5PNUJuGosPtm0zKGeeUQRQVzTARoKARRr
6b6r6us4sl64H6Kzke6cXlXdQVROJszYyqrfKGHcKrXuagYt1f7N3pqWXHe0LwoyVftASNtgveh4U5Ad
EZI; expires=Tue, 19-Sep-2017 22:57:06 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: quic=":443"; ma=2592000; v="37,36,35"
accept-ranges: none
vary: Accept-Encoding
```

The output of your `curl` command should be similar to the screen capture above. If so, you may type `exit` to leave the shell, and close your connection to the pfSense console. If you had any problems with any of the commands above, you've got troubleshooting to do. When it comes to troubleshooting, start at layer 1 of the OSI model (Physical) and work your way up. Here are some questions to help the troubleshooting process:

- Which of the commands above failed? Did they all fail?
- Have you checked your physical cabling or wireless network connectivity?
- Does your company/LAN have MAC filtering/port security on? You might need to modify switch settings for your drop, or talk to your network administrator.
- Have you checked your upstream gateway/SOHO router? Trying pinging the gateway's IP address to see if your pfSense VM can reach the gateway. Try running the `traceroute` command to google.com to see if and where your ping command is failing.

- Do you have an upstream firewall in place? Is 443/TCP allowed outbound? Is 53/UDP allowed outbound? Is ICMP allowed?
- Did you input the commands correctly? Try entering them again. No, I’m not patronizing you.

Once you have verified that your firewall’s connectivity, policies, and services are properly configured, **SNAPSHOT THE VM <u>BEFORE</u>** moving on and trying your hand at setting up the remaining VMs.

Your Turn

The pfSense virtual machine required a lot of custom configuration, both in VMware Fusion, as well as in the VM itself once the OS had been installed. Now that we have done that together, I'm going to have you create the rest of the virtual machines yourself -- with the exception of the Metasploitable 2 VM since it's a special case for reasons you will see momentarily. Below are a set of spec sheets to help you create the remaining virtual machines for the lab on your own. Think of them as checklists you should be able to run through unaccompanied.

All of our Linux-based virtual machines are built on Debian Linux based distros (both, Ubuntu and Kali Linux are derived from Debian), so setup and configuration of all of our lab VMs should be very similar.

Kali Linux VM

Kali Linux is a popular penetration testing distro. Popular because hackers and script kiddies are lazy, and practically everything you need for performing a penetration test or joining anonymous is installed by default. Kali is going to be our loud, obnoxious attacker that we're going to use as a noise generator for the express purpose of generating events on our IPS VM. I want you to perform the following tasks:

- Download Kali Linux 64-bit here: https://www.kali.org/downloads/
- Create a new, custom virtual machine with the following settings:
    - Configure the VM to install from the Kali Linux ISO you downloaded
    - Set the Guest OS family to *Linux*
    - Set the Guest OS version to *Debian 8.x 64-bit*
    - Create a new virtual disk
    - Name the VM "Kali"
    - Note that after saving the VM and granting it a name, VMware Fusion immediate assumes that you want to turn the virtual machine on. In order to adjust the hardware settings as described below, go to the main menu, click *Virtual Machine*, then select *Shut Down*.
- Change the following hardware settings (Virtual Machine > Settings, or "Customize Settings" on the "Finish" page of the new VM wizard):
    - Under *CD/DVD (IDE)*, choose the *Choose a disc or disc image* and find the Kali Linux ISO you previously downloaded
        - Ensure that the *Connect CD/DVD Drive* option is checked
    - Allocate 4GB (4096MB) of RAM
    - Modify the *Hard Disk (SCSI)* settings
        - Change the disk size to 80.00GB

        - Uncheck the Advanced option *Split into multiple files*
        - Check the Advanced option *Pre-allocate disk space*
    - Allocate 1-2 processors, where total number of cores is no more than 2
    - Connect the *Network Adapter* to *vmnet3*
        - Ensure that Connect Network Adapter checkbox is checked
        - Under *Advanced options*, click the *Generate* button to generate a new MAC address. **Record this MAC address as well as the VM it corresponds to.**
    - Remove the USB Controller
    - Remove the Sound Card
    - Remove the Printer
    - Remove the Camera
    - In the *Isolation* settings, uncheck the *Enable Drag and Drop* and *Enable Copy and Paste*
- Install Kali Linux
    - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.2.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy
- After the installation is completed, power off the VM and adjust the following settings:
    - Remove the virtual CD/DVD drive
- Log in to the pfSense web UI, navigate to *Services > DHCP Server*, and add a static mapping to the OPT1 interface DHCP for the MAC address of the Kali VM's network adapter; assign it the IP address 172.16.2.2, making sure to enter a description that tells you which VM this static mapping belongs to
- Power on the virtual machine and do the following:
    - Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense
        - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
    - Verify the virtual machine can reach the internet
        - In a terminal/shell run the command `curl -I https://www.google.com`
            - This will confirm DNS can resolve domain names, and that the VM can reach the internet
            - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense
                - **Note:** The `ping` command, in this case, will work from this virtual machine, due to the unique firewall policy on the

OPT1 network

- Update/patch the VM
  - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
    - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
- Create a snapshot for the VM

Now you should have a functional Kali Linux VM. At this point, the virtual machine should be fully operational, and you should be able to control it from the VM's console. However, if you want to enable and configure SSH access to this host, I have configured a guide for doing so. Jump to "How to Enable SSH on Kali Linux" to enable the ssh service for this VM.

SIEM VM

SIEM is shorthand for Security Intrusion Events Manager. This is fancy security nomenclature for "log aggregator"; we will be having our IPS VM log its events here. We're going to be running Splunk on this VM. Splunk is a commercial program for managing logs on a pretty large scale. By default, and with no licensing, Splunk only allows you to collect or "index" 500MB worth of logs per day, however, there are /ways/ around this that we'll talk about later. I want you to perform the following tasks to set up the SIEM VM:

- Download the latest Ubuntu Server 64-bit LTS release. In our case, this is Ubuntu Server 16.04.1, which can be found here: http://www.ubuntu.com/download/server
- Create a new, custom virtual machine with the following settings:
  - Configure the VM to install from the Ubuntu Linux ISO you downloaded
  - Set the Guest OS family to *Linux*
  - Set the Guest OS version to *Ubuntu 64-bit*
  - Create a new virtual disk
  - Name the VM "SIEM"
  - Note that after saving the VM and granting it a name, VMware Fusion immediately assumes that you want to turn the virtual machine on. In order to adjust the hardware settings as described below, go to the main menu, click *Virtual Machine*, then select *Shut Down*.
- Change the following hardware settings (*Virtual Machine* > *Settings* or you can click the *Customize Settings* button on the finish page of the new VM wizard):
  - Under *CD/DVD (IDE)*, select the *Choose a disc or disc image* option, and find the Ubuntu 16.04.1 LTS 64-bit Linux ISO you previously downloaded
    - Ensure that the *Connect CD/DVD Drive* option is checked
  - Allocate 4GB (4096MB) of RAM
  - Modify the *Hard Disk (SCSI)* settings
    - Change the disk size to 80.00GB
    - Uncheck the Advanced option *Split into multiple files*

        - Check the Advanced option *Pre-allocate disk space*
    - Allocate 1-2 processors, where total number of cores is no more than 2
    - Connect the *Network Adapter* to *vmnet2*
        - Ensure that *Connect Network Adapter* checkbox is checked
        - Under *Advanced settings*, click the *Generate* button to generate a new MAC address. **Record this MAC address as well as the VM and network it corresponds to.**
    - Remove the USB Controller
    - Remove the Sound Card
    - Remove the Printer
    - Remove the Camera
    - In the *Isolation* settings, uncheck the *Enable Drag and Drop* and *Enable Copy and Paste*
- Install Ubuntu Server
    - Be sure to install the Standard System Utilities and OpenSSH Server role when asked. You should not need to install any other roles or features for this server.
    - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.1.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy.
- After the installation is completed, power off the VM and adjust the following settings:
    - Remove the virtual CD/DVD drive
- Log in to the pfSense web UI, navigate to *Services > DHCP Server*, and add a static mapping to the *LAN* interface DHCP, for the MAC address of the SIEM VM's network adapter; assign it the IP address 172.16.1.3, making sure to enter a description that tells you which VM this static mapping belongs to!
- Power on the virtual machine and do the following:
    - Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense
        - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
    - Verify the virtual machine can reach the internet
        - In a terminal/shell run the command `curl -I https://www.google.com`
            - This will confirm DNS can resolve domain names, and that the VM can reach the internet
            - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense
                - **Note:** The `ping` command will NOT work from any of your

virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail

- Update/patch the VM
  - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
    - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
- Create a snapshot for the VM

IPS VM

This VM is going to be responsible for running the AFPACKET bridge between the *IPS 1* (*vmnet3*) and *IPS 2* (*vmnet4*) virtual networks. Perform the following tasks to install the IPS VM:

- Download the latest Ubuntu Server 64-bit LTS release. In our case, this is Ubuntu Server 16.04.1, which can be found here: http://www.ubuntu.com/download/server
- Create a new, custom virtual machine with the following settings:
  - Configure the VM to install from the Ubuntu Linux ISO you downloaded
  - Set the Guest OS family to *Linux*
  - Set the Guest OS version to *Ubuntu 64-bit*
  - Create a new virtual disk
  - Name the VM “IPS”
  - Note that after saving the VM and granting it a name, VMware Fusion immediately assumes that you want to turn the virtual machine on. In order to adjust the hardware settings as described below, go to the main menu, click *Virtual Machine*, then select *Shut Down*.
- Change the following hardware settings (*Virtual Machine > Settings* or you can click the *Customize Settings* button on the finish page of the new VM wizard):
  - Under *CD/DVD (IDE)*, select the *Choose a disc or disc image* option, and find the Ubuntu 16.04.1 LTS 64-bit Linux ISO you previously downloaded
    - Ensure that the *Connect CD/DVD Drive* option is checked
  - Allocate 2GB (2048MB) of RAM
  - Modify the *Hard Disk (SCSI)* settings
    - Change the disk size to 80.00GB
    - Uncheck the Advanced option *Split into multiple files*
    - Check the Advanced option *Pre-allocate disk space*
  - Allocate 1-2 processors, where total number of cores is no more than 2
  - Connect the *Network Adapter* to *vmnet2*
    - Ensure that *Connect Network Adapter* checkbox is checked
    - Under *Advanced settings*, click the *Generate* button to generate a new MAC address. **Record this MAC address as well as the VM and network it corresponds to.**
  - Remove the USB Controller

- Remove the Sound Card
- Remove the Printer
- Remove the Camera
- In the *Isolation* settings, uncheck the *Enable Drag and Drop* and *Enable Copy and Paste*
- Add two more network adapters
  - Attach Network Adapter 2 to vmnet3 (IPS 1)
    - Under *Advanced settings*, click the *Generate* button to generate a new MAC address. **Record this MAC address as well as the VM and network it corresponds to.**
    - **Uncheck the *Connect Network Adapter* checkbox!**
  - Attach Network Adapter 3 to vmnet4 (IPS 2)
    - Under *Advanced settings,* click the *Generate* button to generate a new MAC address. **Record this MAC address as well as the VM and network it corresponds to.**
    - **Uncheck the *Connect Network Adapter* checkbox!**

- Install Ubuntu Server
  - The installer will reach a section titled *Configure the network* and ask you which network card is the primary network interface for this system
    - Usually, these interfaces are labeled `eth0`, `eth1`, and `eth2`, but may be labeled differently in your case
    - In almost all cases, the first network adapter added to a VM in VMware Fusion will line up with the first network interface listed on the *Configure the network* screen (which is almost always `eth0`). Hit Enter, and wait for it to try and configure the interface via DHCP.
    - The installer will try to get an IP address for the network interface via DHCP. If this is not successful, select the `<Go Back>` option, and select one of the other two network interfaces listed, and try again.
    - If none of the interfaces work, verify the pfSense VM is running, connected to the *Management* network, and that the DHCP service is enabled, then try again until DHCP configuration is successful.
  - Be sure to install the Standard System Utilities and OpenSSH Server role when asked. You should not need to install any other roles or features for this server.
  - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.1.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy.
- After the install is completed, power off the VM and adjust the following settings:
  - Remove the virtual CD/DVD drive
  - Navigate to *Network Adapter 2*, and check the *Connect Network Adapter* checkbox

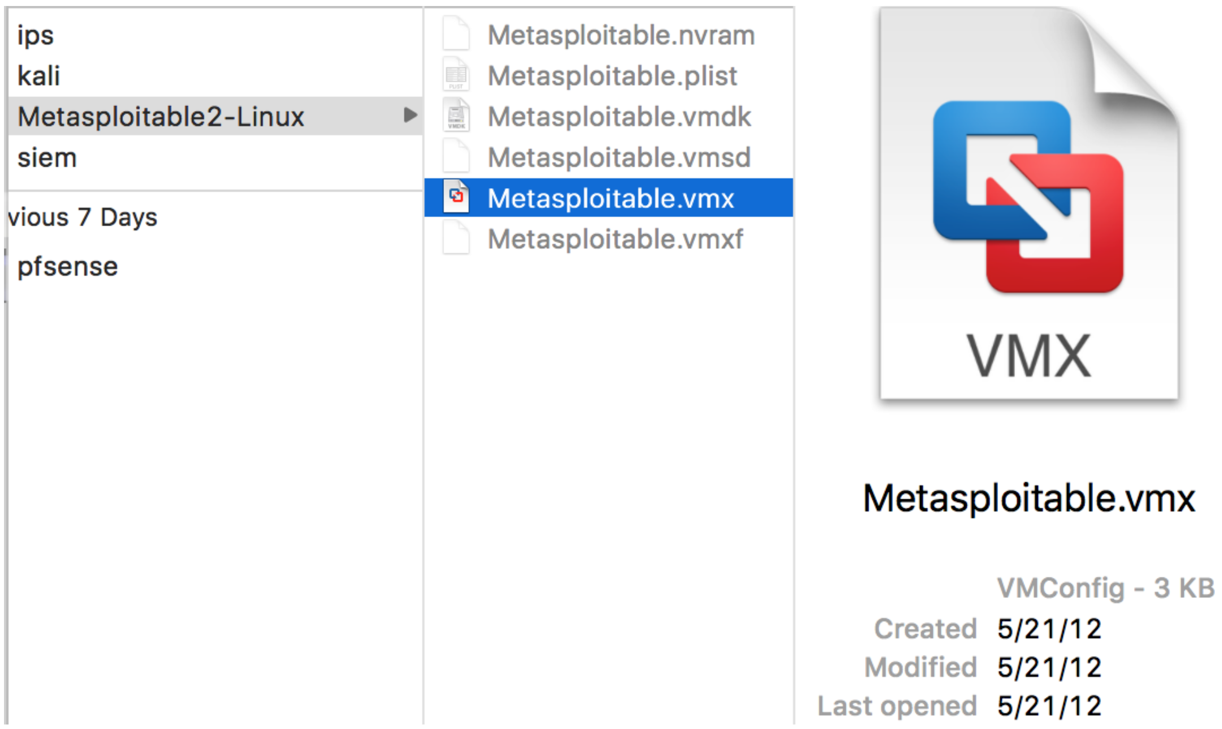
        - Confirm *Network Adapter 2* is connected to *vmnet3*
    - Navigate to *Network Adapter 3*, and check the *Connect Network Adapter* checkbox
        - Confirm *Network Adapter 3* is connected to *vmnet4*
- Log in to the pfSense web UI, navigate to *Services > DHCP Server*, and add a static mapping to the *LAN* interface DHCP, for the MAC address of the original network adapter for the IPS VM (attached to the *Management* network, *vmnet2*); assign it the IP address 172.16.1.4, making sure to enter a description that tells you which VM this static mapping belongs to!
- Power on the virtual machine and do the following:
    - Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense.
        - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
    - Verify the virtual machine can reach the internet
        - In a terminal/shell run the command `curl -I https://www.google.com`
            - This will confirm DNS can resolve domain names, and that the VM can reach the internet
            - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense
                - **Note:** The `ping` command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail
    - Update/patch the VM
        - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
            - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
    - Create a snapshot for the VM

Metasploitable 2

Metasploitable 2 is a little bit different from the other VMs. As it is already pre-created, all we need to do is import it, and reconfigure some of the virtual hardware. This should be pretty simple to do. First, download a copy of Metasploitable 2 from https://sourceforge.net/projects/metasploitable/files/Metasploitable2/. Metasploitable 2 is distributed as a zip archive. The Finder application will unzip the file if you double click on it; alternatively you can use the command line utility `unzip` to do so.

After downloading and decompressing the archive, you will be left with a folder labeled `metasploitable2-linux-2.0.0.0` in your Downloads directory. Inside of that directory is another directory named `Metasploitable2-Linux`. Move the entire `Metasploitable2-Linux` folder into the directory where the rest of your VMware virtual machines' folders are stored; in my case, this is `/Users/[username]/Documents/Virtual Machines/`.

In VMware Fusion, click *Open…* from the *File* option in the menu bar. A Finder window appears in which you can. navigate to the folder `/Users/[username]/Documents/Virtual Machines/Metasploitable2-Linux.` By double clicking on the file labeled "Metasploitable.vmx", the Metasploitable VM gets registered with VMware Fusion, and becomes ready to be run.



Before powering it on though, we have to adjust some of its settings. Click on the wrench in the Metasploitable2-Linux console window or use "Virtual Machine > Settings…" to bring up the Settings menu for Metasploitable 2. Perform the following actions:

- Reconfigure the *Network Adapter* and connect it to *vmnet4*(*IPS 2*)
    - Under Advanced options, record the MAC address of this adapter, or generate a new one via the *Generate* button. **Record this MAC address as well as the VM it corresponds to.**
- Remove *Network Adapter 2*
- Remove the USB Controller

- Remove the CD/DVD drive
- In the *Isolation* settings, uncheck the *Enable Drag and Drop* and *Enable Copy and Paste*
- Log in to the pfSense web UI, navigate to *Services > DHCP Server*, and add a static mapping to the OPT1 interface DHCP, for the MAC address of the Network Adapter for the Metasploitable 2 VM (attached to the *IPS 2* network, *vmnet4*); assign it the IP address 172.16.2.3, making sure to enter a description that tells you which VM this static mapping belongs to!

Power on the Metasploitable 2 VM, make sure it is bootable. If so, take a snapshot of the virtual machine, it's ready to go for now. You won't be able to verify whether the DHCP mapping works until AFTER the IPS VM is configured.

Next Steps

The VMware Fusion initial setup is all but done at this point. However, you're not quite out of the woods just yet. Here is a checklist of tasks to complete:

- While not strictly necessary, you may also want to complete the "Remote Lab Management" guide. Specifically, the sections related to Linux, BSD and OSX Remote Access, Enabling SSH on Kali Linux, and if you're lazy like me, the section on Securing root SSH access. This will allow you to remotely manage all of your lab VMs, as well as finish the IPS and Splunk setup guides much more easily than through the VirtualBox console.
- You still need to install IDS/IPS software on the IPS VM to get a functioning AFPACKET bridge between the "IPS 1" and "IPS 2" networks. Check out the "IPS Installation Guide" to learn how to do this with either Snort or Suricata as your IPS software of choice.
- The SIEM VM needs to have Splunk installed and configured. You'll need to complete the "Splunk Installation Guide".
- You may want to consider reading the Automated Patching for Linux Lab VMs chapter, and implementing the `updater.sh` script for your Linux VMs.
- Do you want some ideas on where to take your lab? Check out the chapter "In Your Own Image", for some tips on how to mold your VM lab to better suit your needs.

Setup - VMware Workstation Pro

Note: This guide is written for VMware Workstation Pro, specifically, the latest version as of this writing, version 12.5. I'll be writing this guide using Windows as the host operating system, though the host OS itself should not matter too much; you should be able to easily follow along using Windows or Linux as the host OS.

VMware Workstation is a hosted hypervisor offered by VMware. Our guide specifically needs the Pro edition of workstation due to enhanced virtual networking features that aren't available in the standard edition of VMware Workstation.

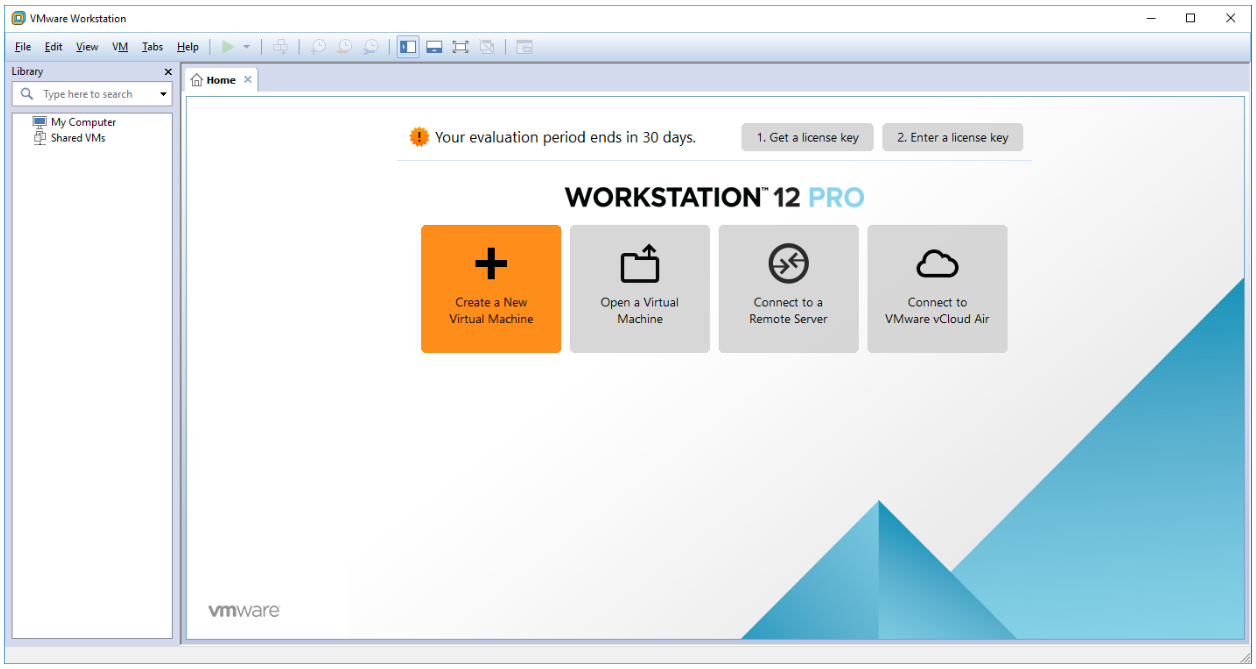
Installation

Installing VMware Workstation is pretty straightforward. You can get a trial copy to download and use for free, without even having an account with VMware. As of right now, this is the link to VMware Workstation (Windows and Linux can be downloaded on this page): http://www.vmware.com/products/workstation/workstation-evaluation.html

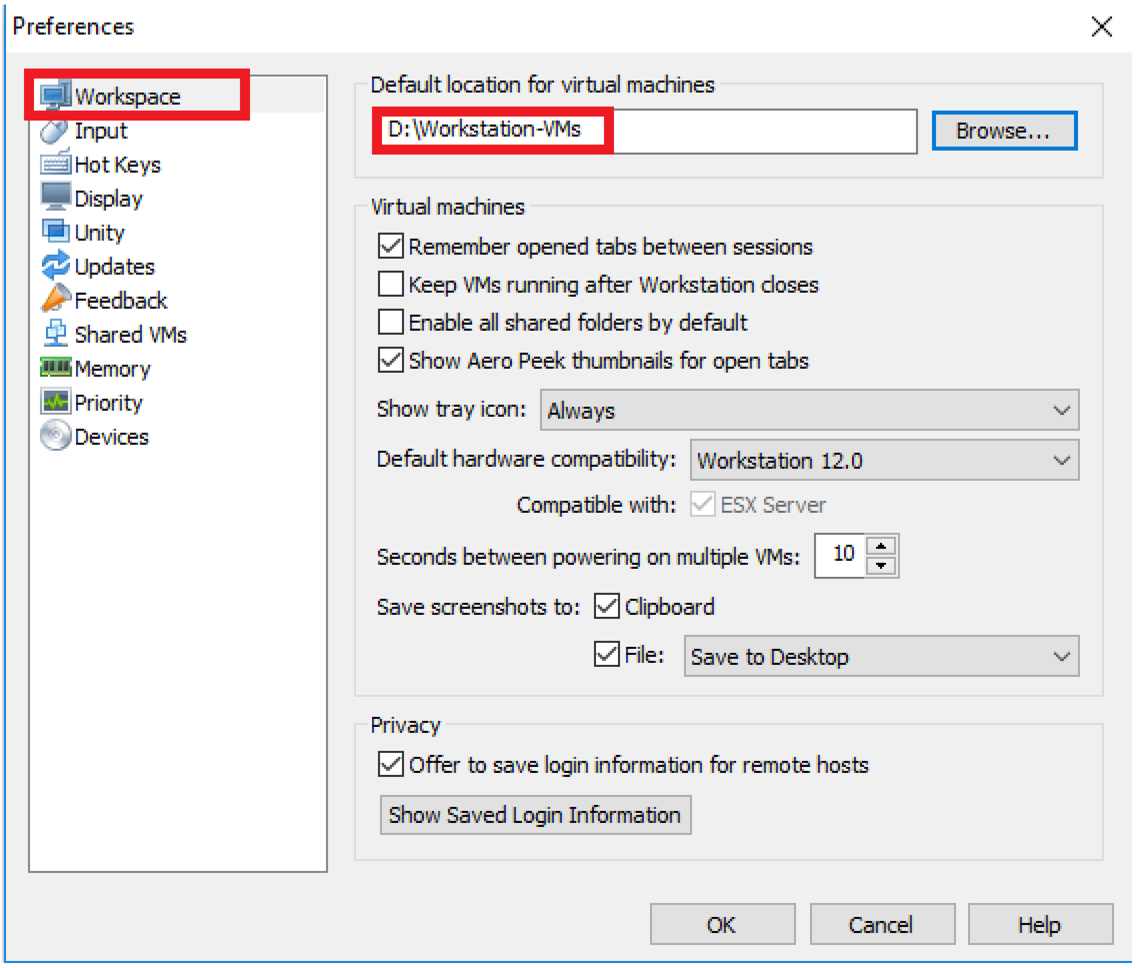
Run the installer and go through the installation steps; the whole process should finish quickly. Please note that when installing on Linux, you'll need to install kernel modules for VMware as a part of the software installation. To do this, usually you'll need the headers for the kernel you're currently running available. While the name of the package(s) that include the headers varies from distro to distro, and from package manager to package manager, usually the headers are included in some sort of a kernel development package. Consult documentation for your Linux distro on how to acquire the kernel headers.

Hypervisor Preferences

Now that you have VMware Workstation installed, start it up. We're going to make some configuration changes here and there, in order to support our lab environment. On first start, the program asks you if you want to input a license key or use the trial period. If you opt to use the trial period, you have to provide a valid e-mail address for VMware to send spam to your inbox. Well, they asked for a *valid* one, which doesn't necessarily mean *your* e-mail address. I suggest you to have a look at 10 Minute Mail, Guerilla Mail or a similar services. I used 10 Minute Mail for my trial period with no problems to report. After entering your license key or e-mail address, you're greeted with a main menu that looks like this:



Click “Edit” on the menu bar, then click “Preferences”. A new window appears where you can modify a variety of settings for VMware Workstation. You may be interested in checking the other settings and setting things to your liking, but for the time being, all we’re concerned with is the option titled “Workspace”, specifically, the setting related to “Default location for virtual machines”. By default, virtual machine files and folders will be placed under your home directory. If you want to change where they are stored, click the Browse button, and choose a new folder to store your VM files in. In my case, I chose to store my virtual machines in `D:\Workstation-VMs`. When you are done configuring this, or if you are satisfied with the defaults, click OK to close the Preferences window.



Default location for virtual machines: D:\Workstation-VMs

Virtual machines
- [✓] Remember opened tabs between sessions
- [ ] Keep VMs running after Workstation closes
- [ ] Enable all shared folders by default
- [✓] Show Aero Peek thumbnails for open tabs

Show tray icon: Always

Default hardware compatibility: Workstation 12.0

Compatible with: [✓] ESX Server

Seconds between powering on multiple VMs: 10

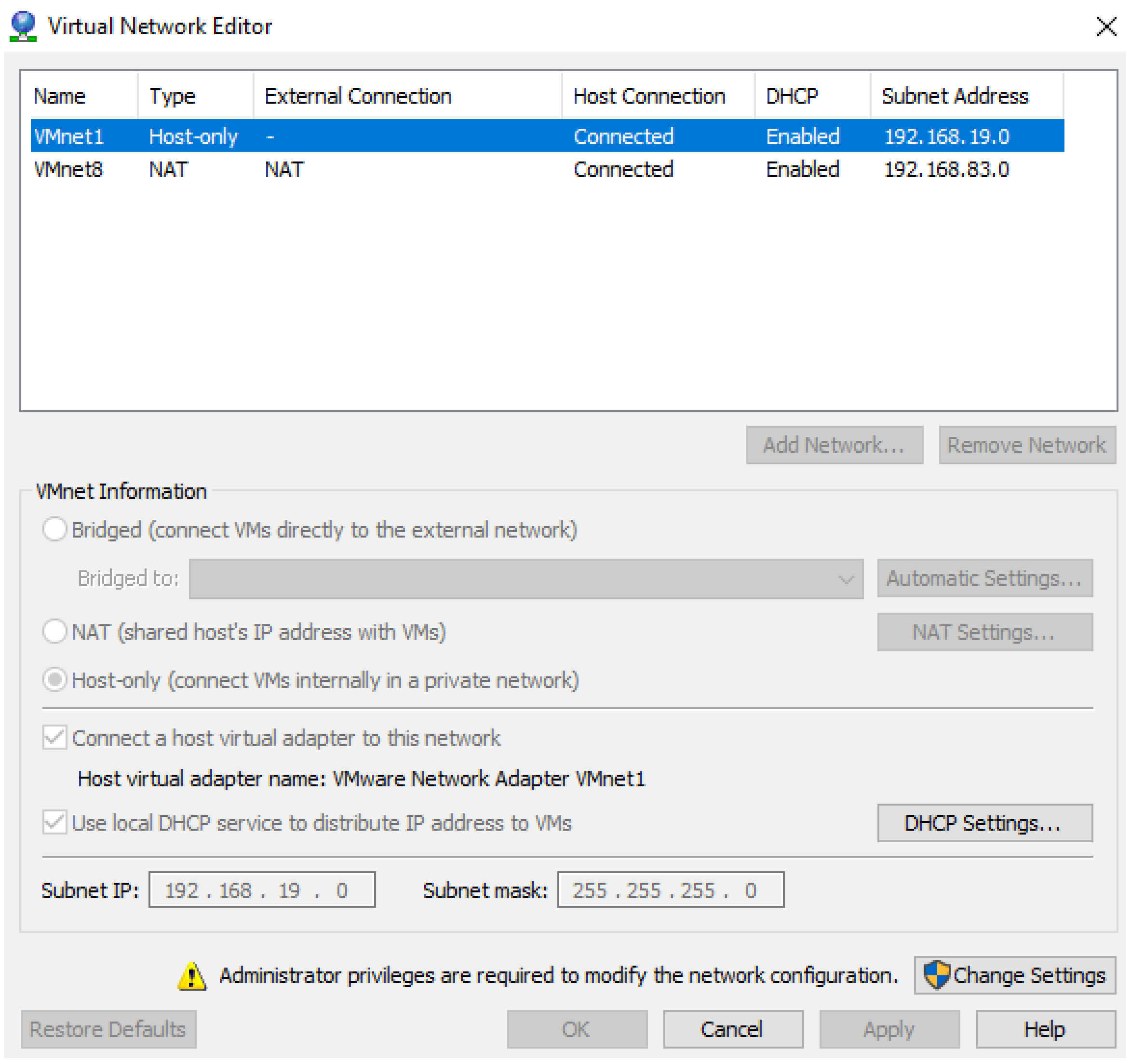
Save screenshots to: [✓] Clipboard

[✓] File: Save to Desktop

Privacy
- [✓] Offer to save login information for remote hosts

Virtual Networks

Back in the main menu, choose “Edit” from the menu bar, then click on “Virtual Network Editor” to open the corresponding window.


Virtual Network Editor

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet1	Host-only	-	Connected	Enabled	192.168.19.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.83.0

VMnet Information

Bridged (connect VMs directly to the external network)

Bridged to:

NAT (shared host's IP address with VMs)

[✓ Host-only (connect VMs internally in a private network)]

[✓] Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet1

[✓] Use local DHCP service to distribute IP address to VMs

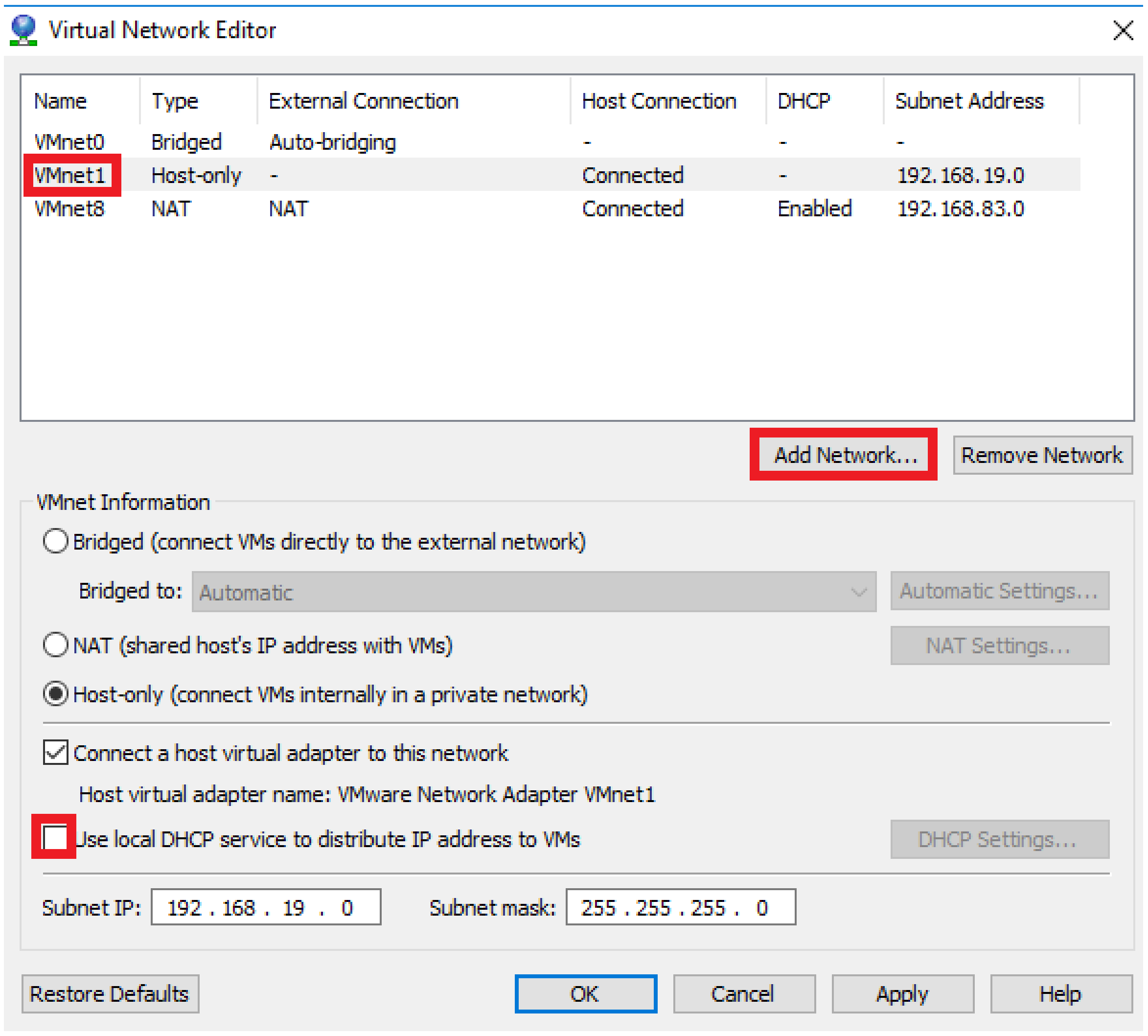
Subnet IP: 192 . 168 . 19 . 0 Subnet mask: 255 . 255 . 255 . 0

Administrator privileges are required to modify the network configuration.

By default, VMware Workstation has three virtual networks defined VMnet1, a host-only network, and VMnet8, a NAT network, and VMnet0, a network for bridging to the host system's network connection. For our lab, VMnet0 will serve as our Bridged Network, VMnet1 will be our Management Network. Additionally, we will be creating two more host-only virtual networks to server as the IPS 1 Network and the IPS 2 Network, respectively.

Note: If you are using VMware Workstation on Windows, the VMnet0 network will not be visible, and you will NOT be able create additional virtual networks until you click the "Change Settings" button in the lower right corner of the Virtual Network Editor window. Afterwards, the greyed out options should be available to edit, and VMnet0 should be visible. Linux users should not have to worry about this.

Click to select VMnet1 in the main window pane, then uncheck the checkbox "Use local DHCP service to distribute IP address to VMs" in the VMnet Information section. Next, we are going to create two additional host-only networks. To start, click on the "Add Network…" button underneath the main window pane.



Virtual Network Editor

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Auto-bridging	-	-	-
VMnet1	Host-only	-	Connected	-	192.168.19.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.83.0

VMnet Information

Bridged (connect VMs directly to the external network)

Bridged to: Automatic

NAT (shared host's IP address with VMs)

[✓ Host-only (connect VMs internally in a private network)]

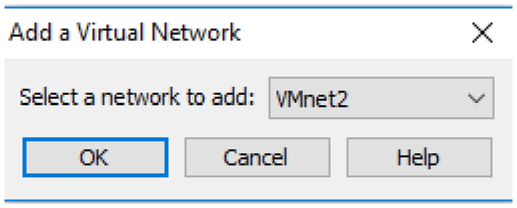
[✓ Connect a host virtual adapter to this network]

Host virtual adapter name: VMware Network Adapter VMnet1

[ ] Use local DHCP service to distribute IP address to VMs

Subnet IP: 192 . 168 . 19 . 0   Subnet mask: 255 . 255 . 255 . 0

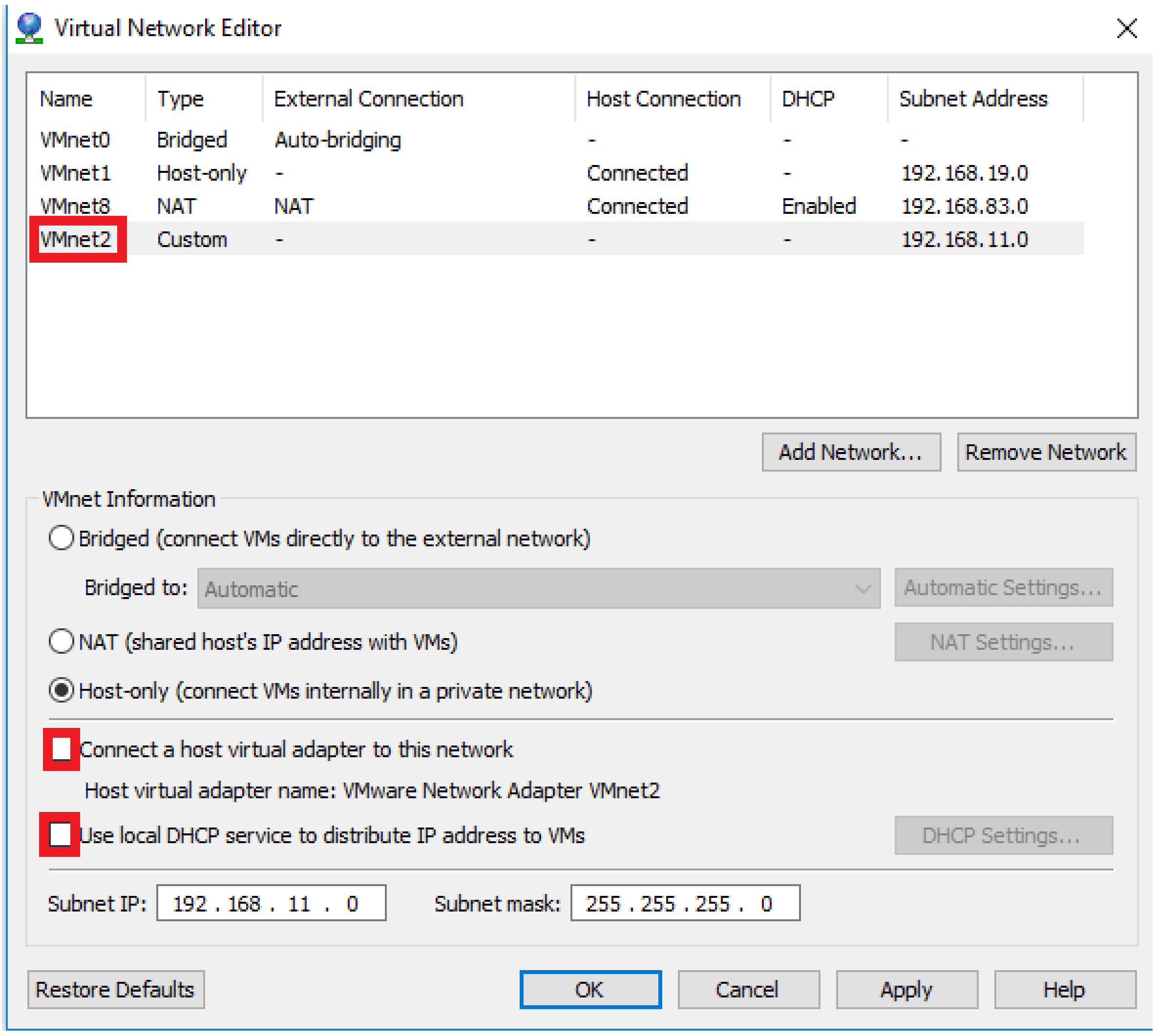
A small window pops up with a drop-down labeled "Select a network to add:". Click OK to choose the default network name (in my case, this was VMnet2). This network is going to serve as our lab's IPS 1 Network.



Add a Virtual Network

Select a network to add: VMnet2

After a moment or two, you should be back at the Virtual Network Editor. The window should have highlighted VMnet2 in the primary pane for you already; if it did not, click to select VMnet2. In the VMnet Information section, verify that the radio button in front of the "Host-only (connect VMs internally in a private network)" option is selected; this should be the default setting. Uncheck the option "Connect a host virtual adapter to this network" and the option "Use local DHCP service to distribute IP address to VMs".


Virtual Network Editor

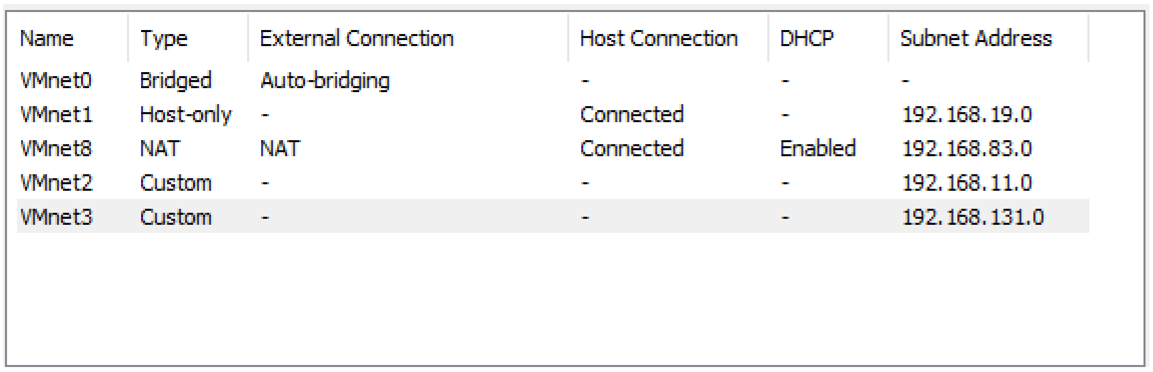
Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Auto-bridging	-	-	-
VMnet1	Host-only	-	Connected	-	192.168.19.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.83.0
VMnet2	Custom	-	-	-	192.168.11.0

VMnet Information
- ○ Bridged (connect VMs directly to the external network)
  - Bridged to: Automatic
- ○ NAT (shared host's IP address with VMs)
- ● Host-only (connect VMs internally in a private network)
- ☐ Connect a host virtual adapter to this network
  - Host virtual adapter name: VMware Network Adapter VMnet2
- ☐ Use local DHCP service to distribute IP address to VMs

Subnet IP: 192 . 168 . 11 . 0   Subnet mask: 255 . 255 . 255 . 0

You might be wondering why we're disabling DHCP and network adapters for VMnet1 and VMnet2 so far, and why we've disabled the host virtual adapter for VMnet2. The reason why we did all that is that we are going to configure pfSense to handle DHCP services, as opposed to VMware Workstation. Leaving DHCP enabled on the VMware virtual networks would cause IP address conflicts; beyond that pfSense's DHCP server offers much more flexibility. Regarding the removal of the host virtual adapter, we're doing this because VMnet2 will be one of the two IPS networks. As we want to protect the hypervisor host from potential threats, we do not want the host OS to have direct access to the VMnet2 network.

We still need an IPS 2 Network, which we are going to create by repeating the process we went through above. Add a new network (in my case the network name was VMnet3), verify that it's type is "Host-only", ensure that the checkboxes "Connect a host virtual adapter to this network", and "Use local DHCP service to distribute IP address to VMs", are unchecked. After completing this task, the primary pane of the Virtual Network Editor should look like this:



Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Auto-bridging	-	-	-
VMnet1	Host-only	-	Connected	-	192.168.19.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.83.0
VMnet2	Custom	-	-	-	192.168.11.0
VMnet3	Custom	-	-	-	192.168.131.0

The virtual networks are now configured to support our lab environment. Click the Apply button, then click OK to confirm the adjusted network settings, and close the Virtual Network Editor window.

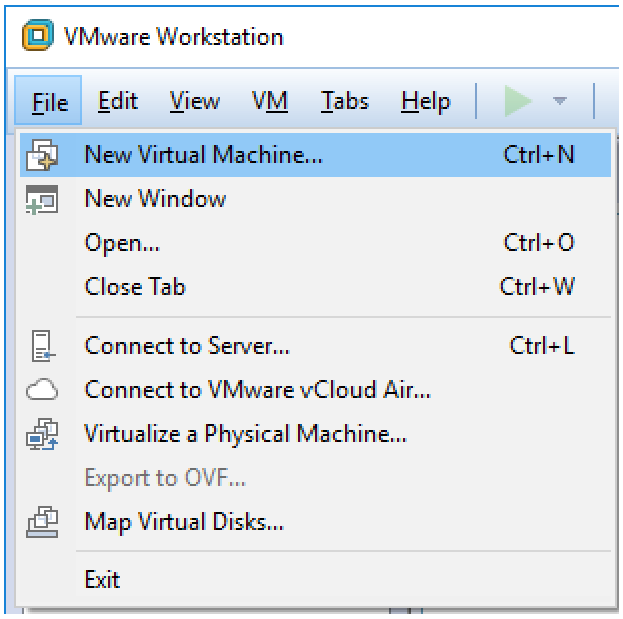
Creating the First VM, pfSense

pfSense is the keystone holding this entire configuration together. Personally, it's my favorite firewall distro due to ease of use, the amount of functionality it includes out of the box, combined with a plugin/add-on system for additional functionality. If you have the CPU, RAM, and disk, pfSense can easily be converted into a so-called "Next-Generation" firewall.

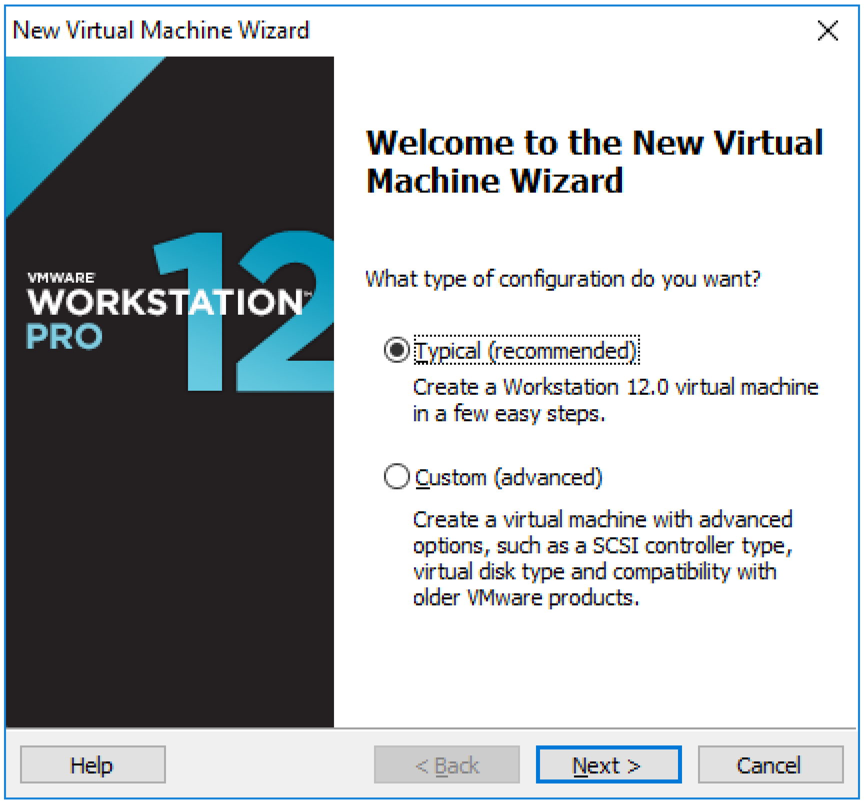
Make your way to https://www.pfsense.org/download/. Download the latest installation ISO for the amd64 architecture. While you're at it, you may want to download a compression utility. The pfSense maintainers distribute pfSense as an ISO image file compressed with `gzip`. This means that we'll need to decompress the ISO file at some point. On Windows, I prefer 7-Zip (http://www.7-zip.org/) as my compression utility of choice for decompressing files, since 7-Zip can handle zip, gzip, rar, and 7z files (among others) easily. Linux on the other hand, usually includes the `gzip` or `gunzip` command line utilities for decompressing gzipped files.

Adding a New VM

In the VMware Workstation main window, click “File” on the menu bar, and select “New Virtual Machine” to start the New Virtual Machine Wizard.



The first screen of the wizard asks you to choose the type of configuration you would like to proceed with. Select the radio button next to “Typical (recommended)”, then click Next. The custom (advanced) configuration type allows you to configure a bunch of additional settings on your VM relating to disk types and storage options. As none of them need to be adjusted for the purposes of our lab, creating virtual machines with the suggested typical configuration will serve us just fine.


New Virtual Machine Wizard

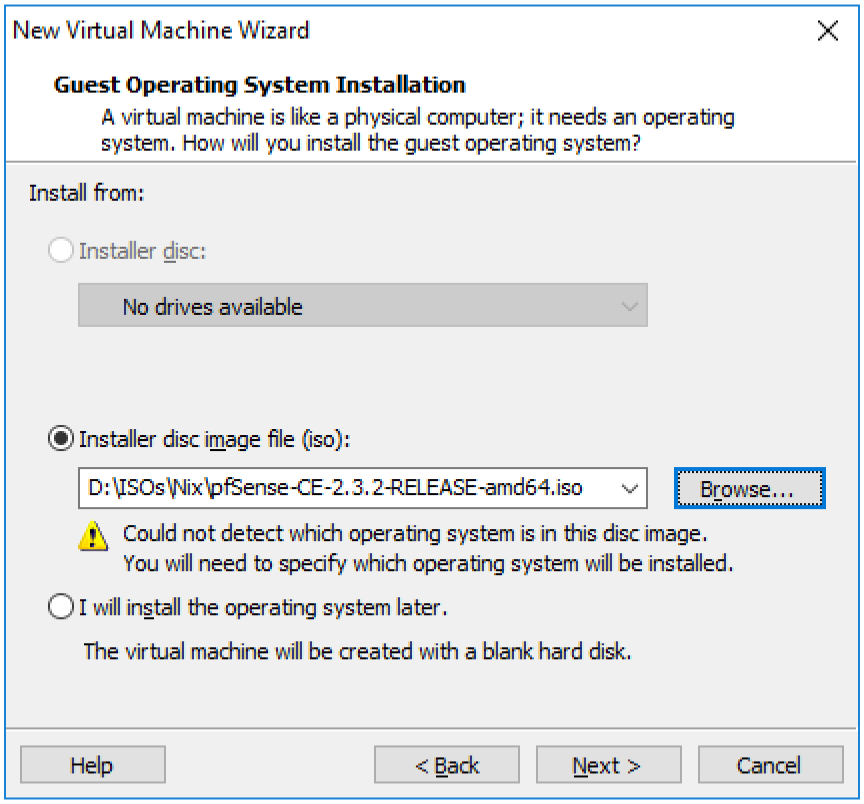
Welcome to the New Virtual Machine Wizard

What type of configuration do you want?

[✓ Typical (recommended)]
Create a Workstation 12.0 virtual machine in a few easy steps.

Custom (advanced)
Create a virtual machine with advanced options, such as a SCSI controller type, virtual disk type and compatibility with older VMware products.

The next screen allows us configure if and how we want to install our VM's operating system. Select the radio button next to "Installer disc image file (iso):" radio button, then click Browse, and an explorer window opens. Navigate to where you stored the compressed pfSense ISO file, and if you haven't already, decompress it as necessary. In my case, the ISO in `D:\ISOs\Nix`. Note that you'll get a warning from the wizard, stating,"Could not detect which operating system is in this disc image." Do not be concerned, VMware workstation attempts to make educated guesses as to what type and version of operating system you are attempting to load. We will be manually specifying the operating system and version shortly.. After selecting the ISO file, click Next.


New Virtual Machine Wizard

Guest Operating System Installation

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

○ Installer disc:

No drives available

● Installer disc image file (iso):

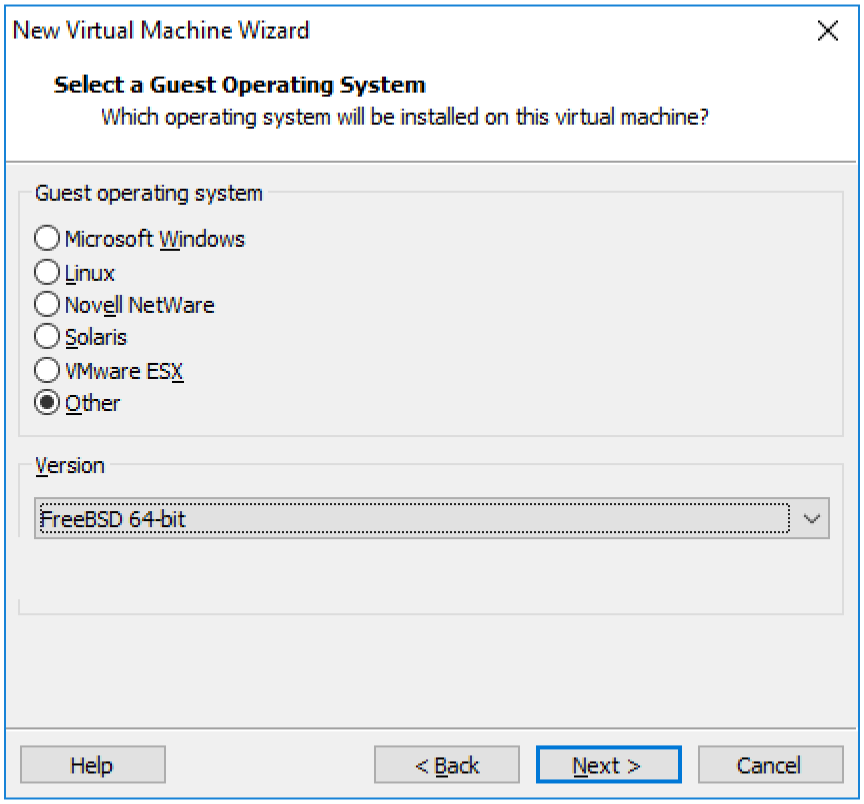
D:\ISOs\Nix\pfSense-CE-2.3.2-RELEASE-amd64.iso

Could not detect which operating system is in this disc image.
You will need to specify which operating system will be installed.

○ I will install the operating system later.

The virtual machine will be created with a blank hard disk.

On the next screen, the wizard asks you to select the guest operating system and its version you are planning to install, since it failed to detect the OS on the previous step.. Click the radio button next to “Other” in the “Guest operating system” section, then select “FreeBSD 64-bit” from the “Version” drop-down list. To go ahead, click Next.


New Virtual Machine Wizard

Select a Guest Operating System

Which operating system will be installed on this virtual machine?

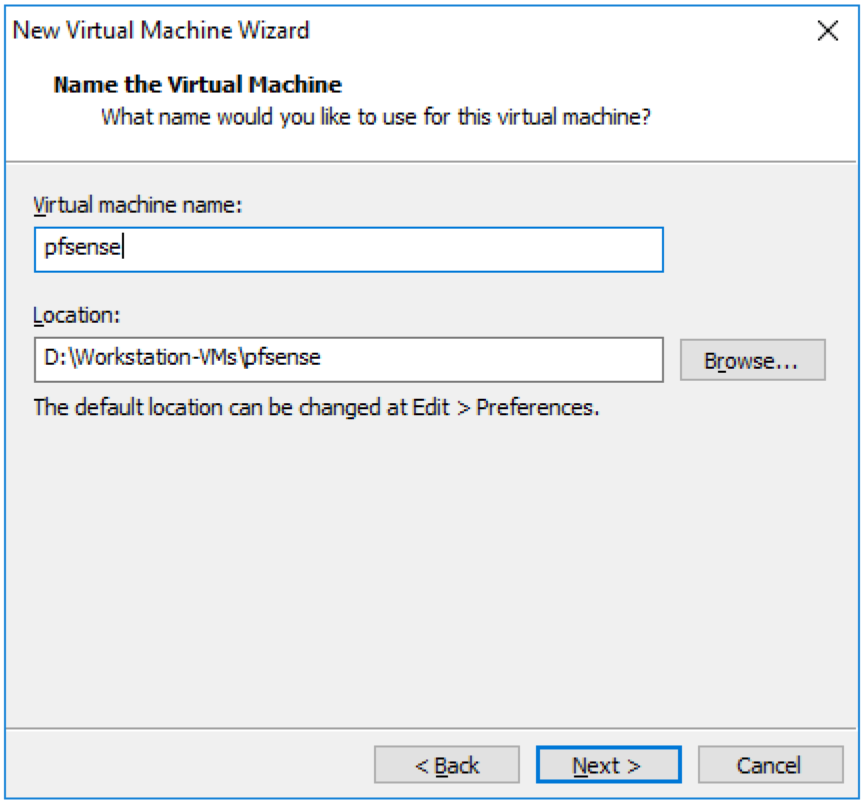
Guest operating system

- ( ) Microsoft Windows
- ( ) Linux
- ( ) Novell NetWare
- ( ) Solaris
- ( ) VMware ESX
- (●) Other

Version

FreeBSD 64-bit

The next screen asks you to name the virtual machine and to confirm the location where the corresponding files will be stored. Enter “pfSense” as the name of the VM, then click Next.


New Virtual Machine Wizard

Name the Virtual Machine

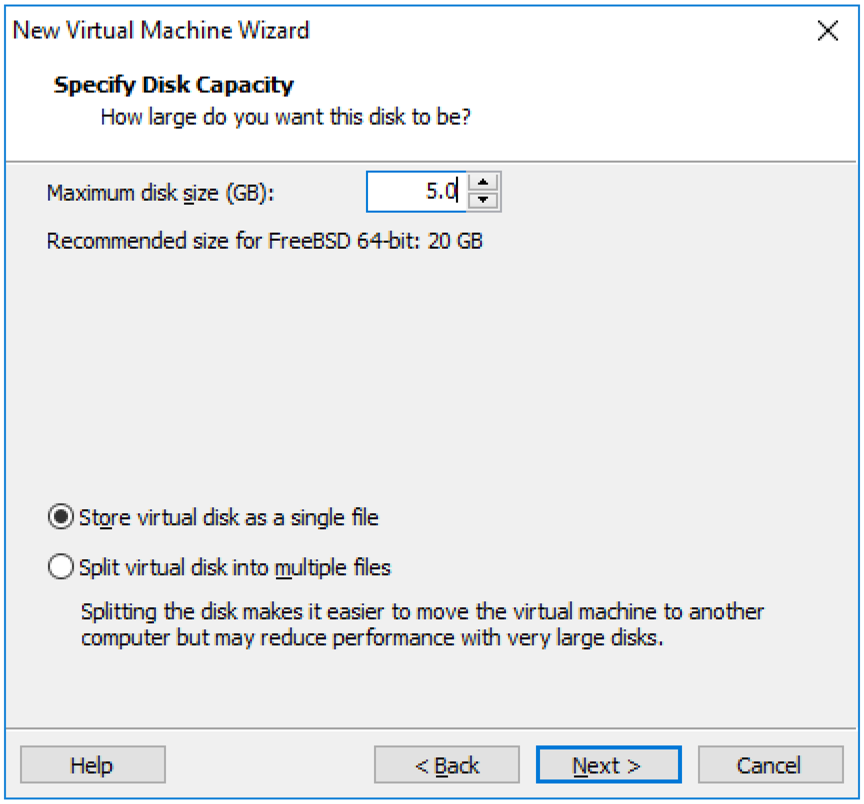
What name would you like to use for this virtual machine?

Virtual machine name: pfsense

Location: D:\Workstation-VMs\pfsense

The default location can be changed at Edit > Preferences.

On the next screen, you are asked to specify how much space you want to allocate for your VM's virtual hard disk. Enter the value 5.0 into the "Maximum disk size (GB):" input box, then select the radio button in front of "Store virtual disk as a single file". Click Next to proceed.



New Virtual Machine Wizard

Specify Disk Capacity

How large do you want this disk to be?

Maximum disk size (GB): 5.0

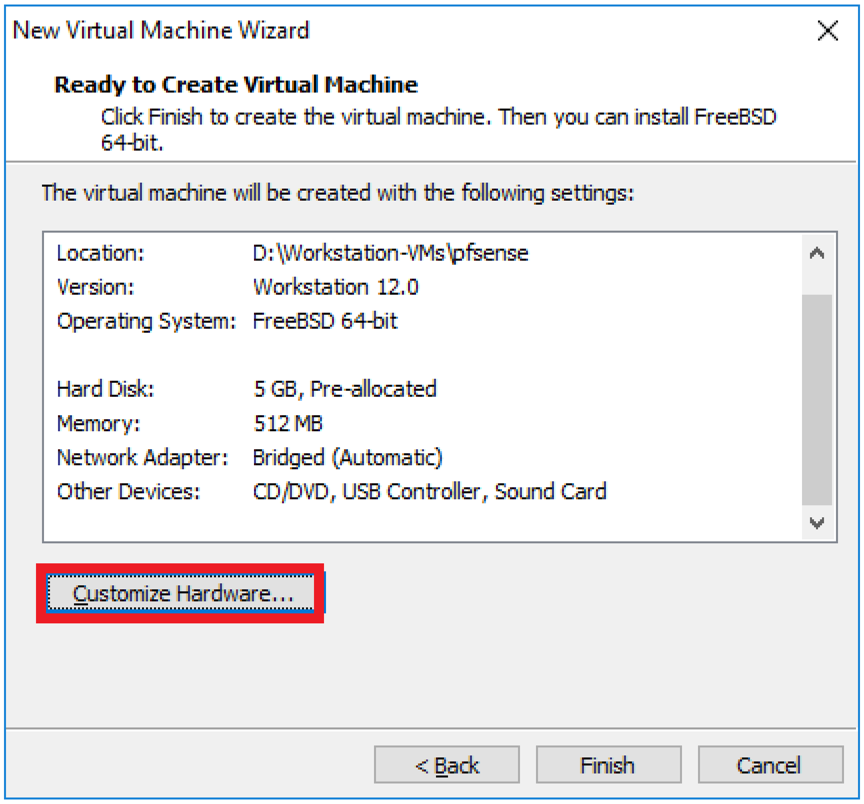
Recommended size for FreeBSD 64-bit: 20 GB

● Store virtual disk as a single file

○ Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

The final screen displays an overview of the current settings for this VM. As we have to adjust parts of the hardware present on the virtual machine, click the “Customize Hardware…” button.


New Virtual Machine Wizard

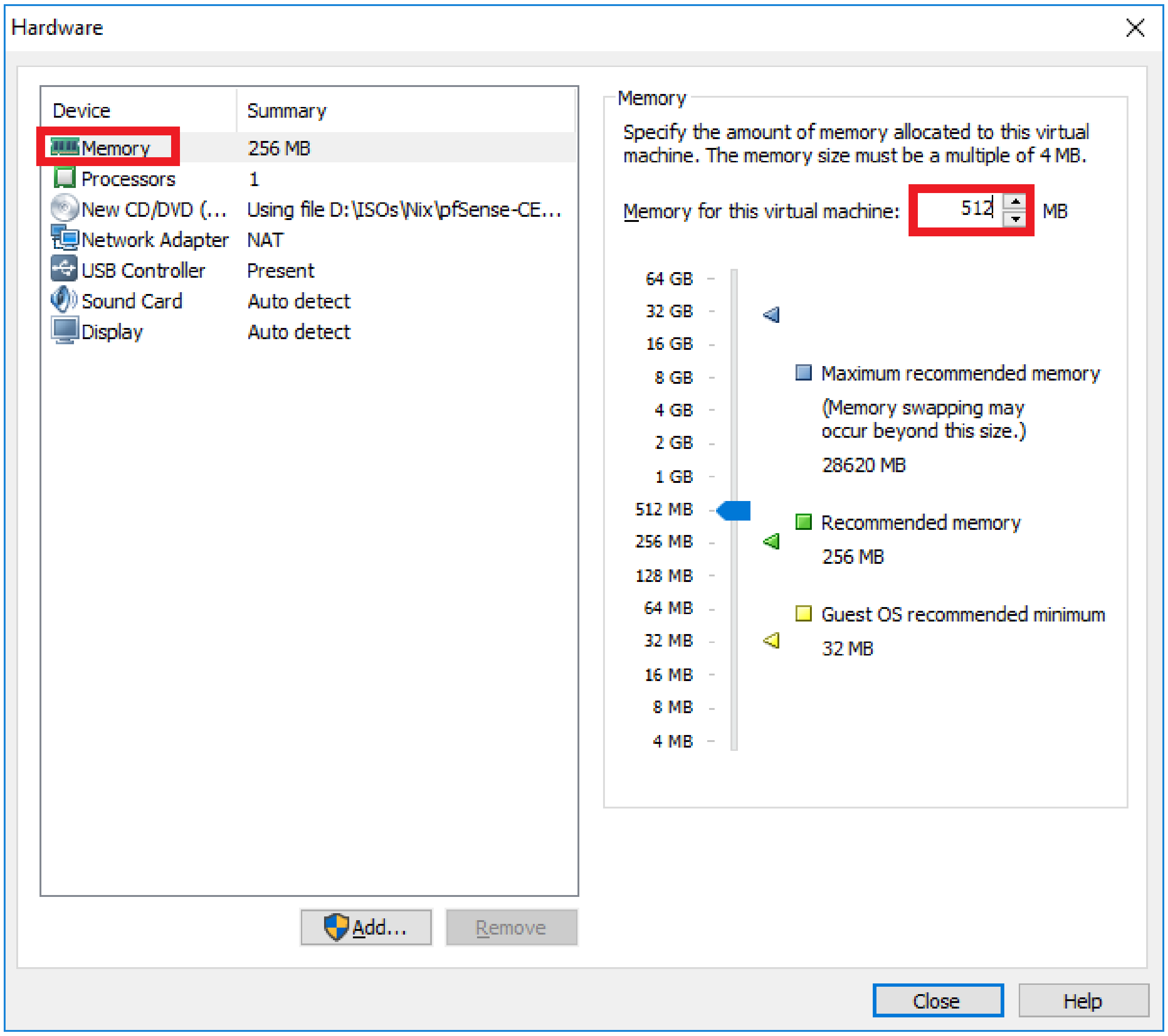
Ready to Create Virtual Machine

Click Finish to create the virtual machine. Then you can install FreeBSD 64-bit.

The virtual machine will be created with the following settings:

Location:	D:\Workstation-VMs\pfsense
Version:	Workstation 12.0
Operating System:	FreeBSD 64-bit
Hard Disk:	5 GB, Pre-allocated
Memory:	512 MB
Network Adapter:	Bridged (Automatic)
Other Devices:	CD/DVD, USB Controller, Sound Card

A new window appears, listing the hardware components currently attached to our new VM. In the “Device” pane on the left, click to highlight the “Memory” entry to bring up the related informations and settings in the pane to the right. We’re going to double the default amount of 256MB to 512MB, by inputting 512 into the “Memory for this virtual machine:” input box.



Device	Summary
Memory	256 MB
Processors	1
New CD/DVD (...	Using file D:\ISOs\Nix\pfSense-CE...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

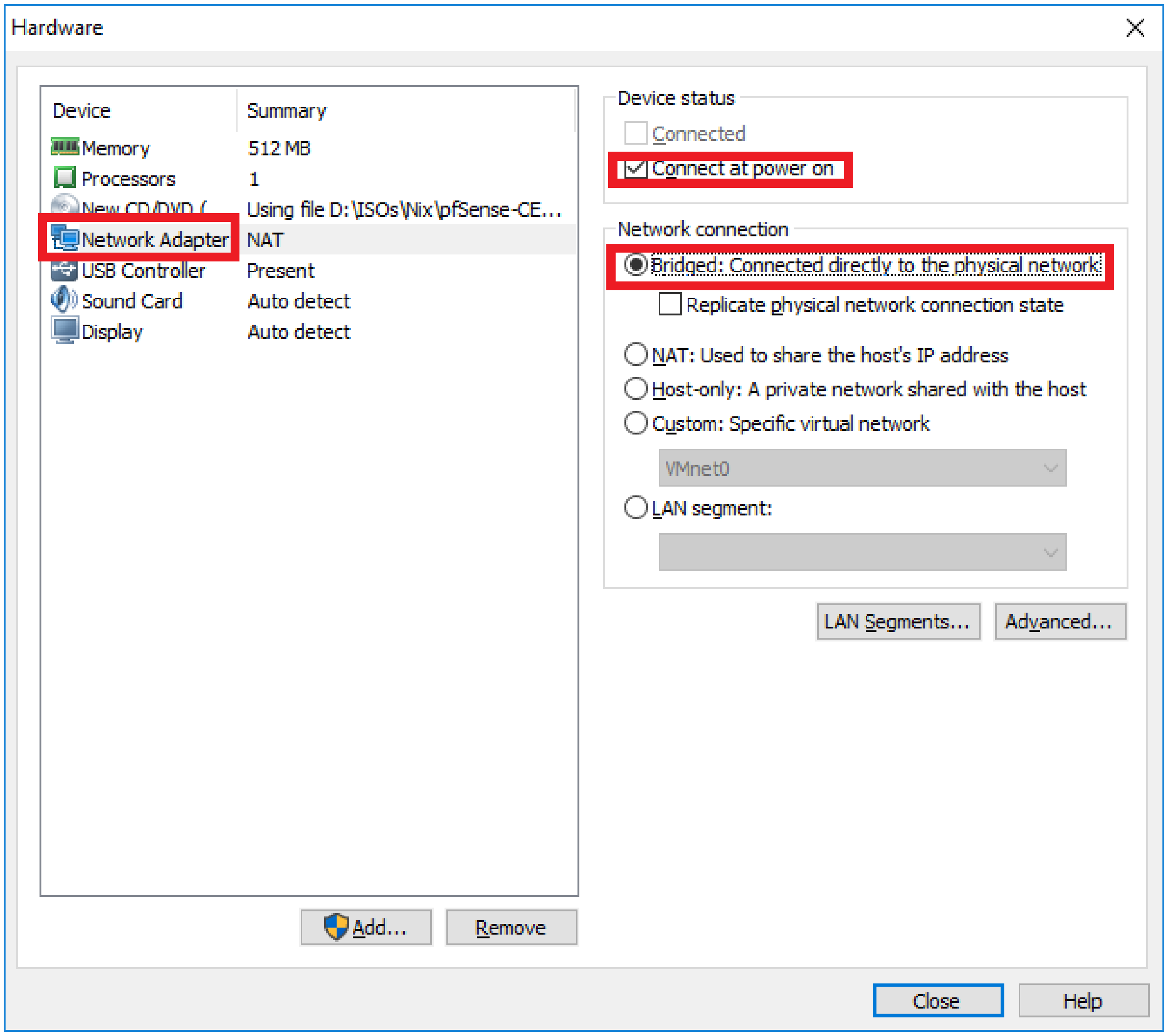
Memory for this virtual machine: 512 MB

Maximum recommended memory
(Memory swapping may occur beyond this size.)
28620 MB

Recommended memory
256 MB

Guest OS recommended minimum
32 MB

Next, click on “Network Adapter” in the “Device” pane to bring up the settings for this network adapter. Ensure that the option “Connect at power on” is checked in the “Device Status” section. In the area below, titled “Network connection”, change the radio button from the default setting of “NAT: Used to share the host’s IP address” to “Bridged: Connected directly to the physical network”.



Hardware

Device	Summary
Memory	512 MB
Processors	1
New CD/DVD (	Using file D:\ISOs\Nix\pfSense-CE…
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Device status

☐ Connected

☑ Connect at power on

Network connection

[✓ Bridged: Connected directly to the physical network]

☐ Replicate physical network connection state

NAT: Used to share the host's IP address

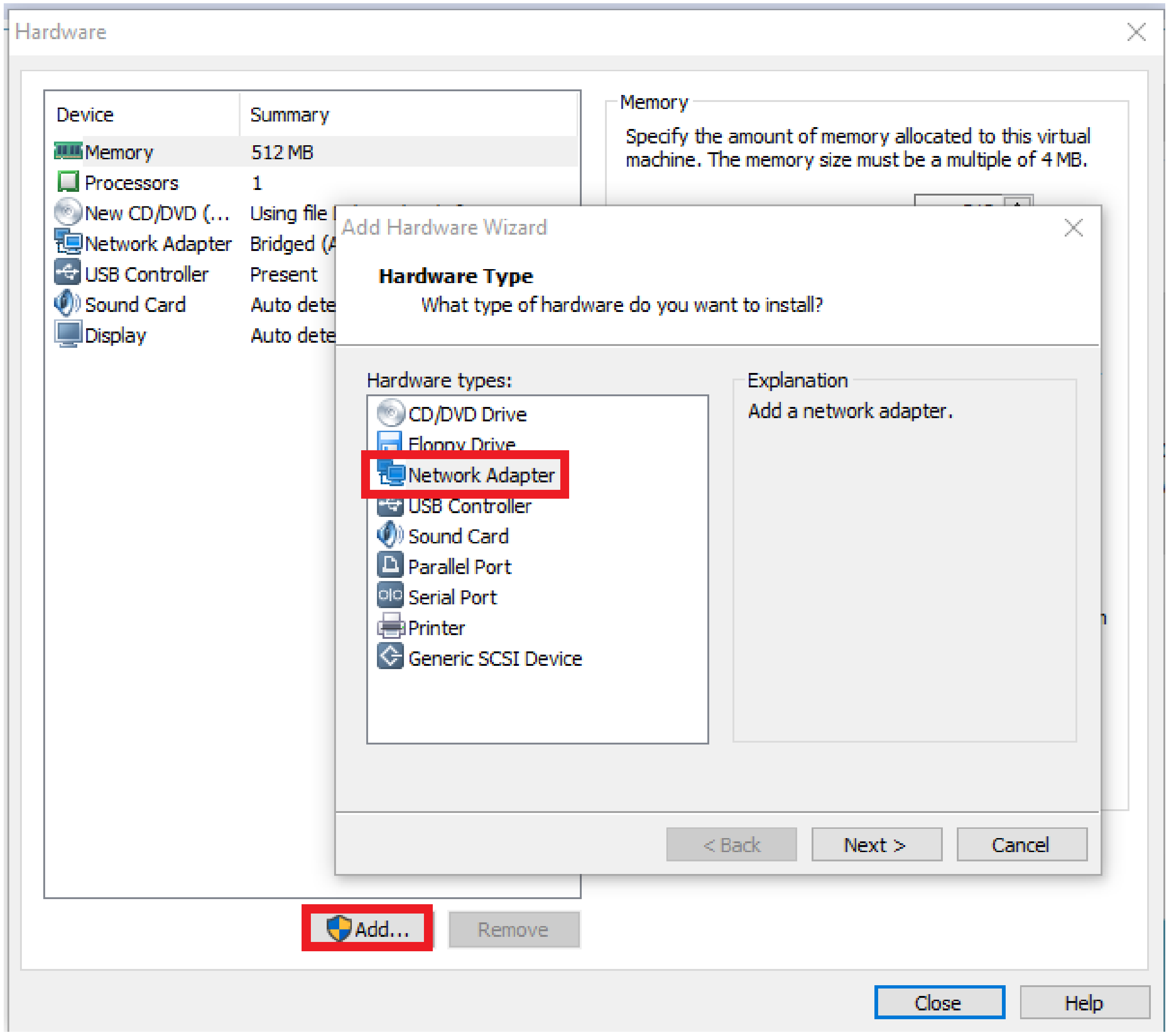
Host-only: A private network shared with the host

Custom: Specific virtual network

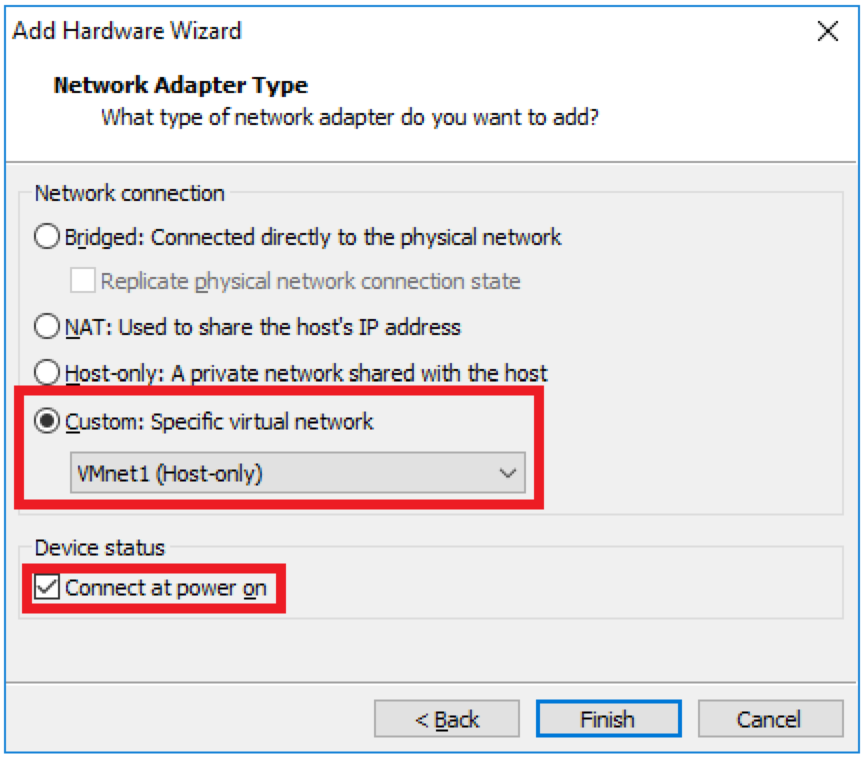
VMnet0

LAN segment:

Afterwards, click the “Add…” at the bottom of the “Device” pane to start the “Add Hardware Wizard”. Click“Network Adapter” under the pane labeled “Hardware types:”, then click Next.



On the next screen, Click the radio button next to “Custom: Specific virtual network”, then select “VMnet1” in the drop-down menu. Make sure that the “Connect at power on” checkbox is checked, then click Finish.


Add Hardware Wizard

Network Adapter Type

What type of network adapter do you want to add?

Network connection

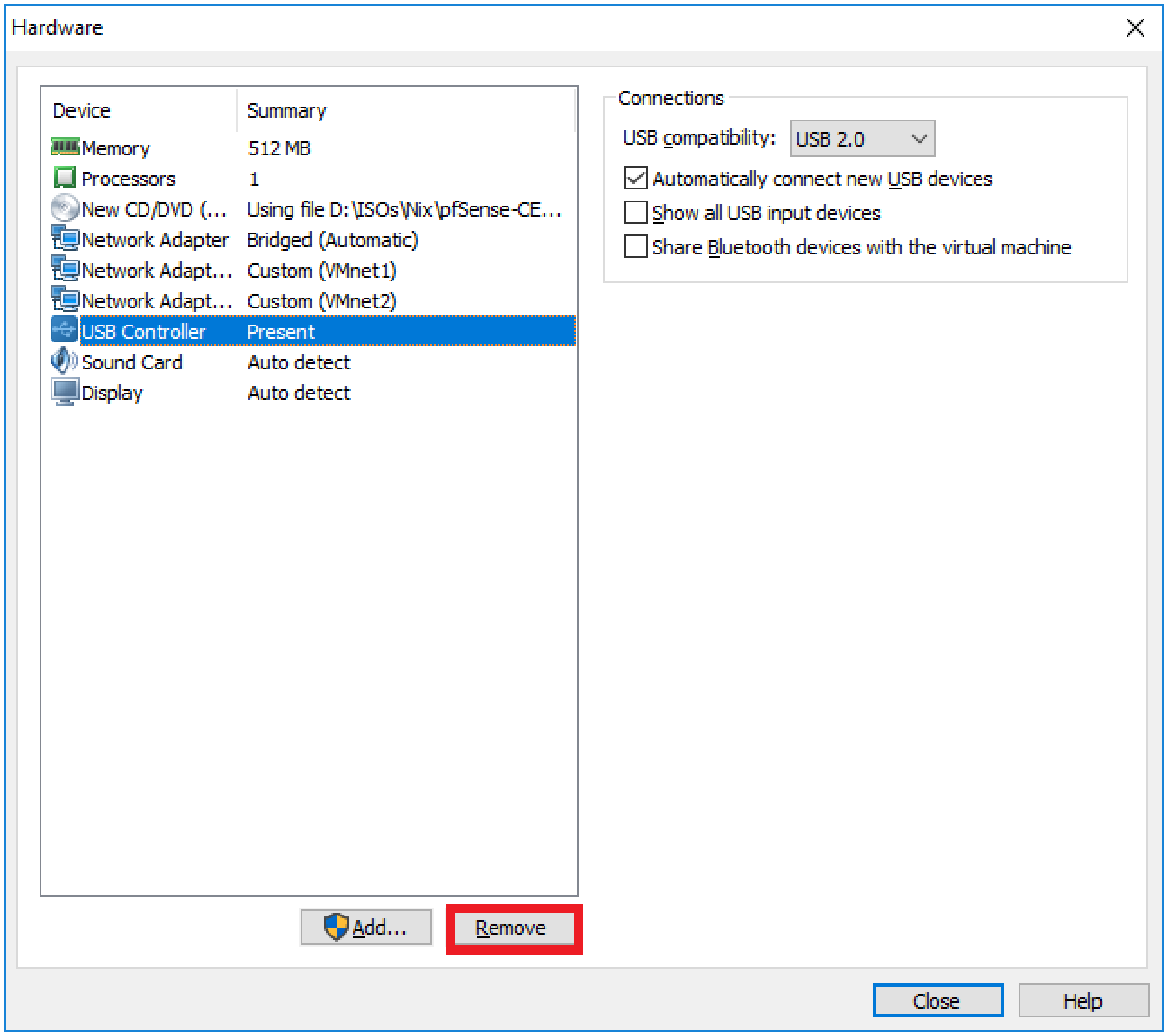
- ○ Bridged: Connected directly to the physical network
  - ☐ Replicate physical network connection state
- ○ NAT: Used to share the host's IP address
- ○ Host-only: A private network shared with the host
- ● Custom: Specific virtual network
  - VMnet1 (Host-only)

Device status

- ☑ Connect at power on

Repeat this process once more, creating a third network adapter, except this time, connect it to the VMnet2 virtual network.

After adding the two network cards (for a grand total of 3 network adapters attached to this VM), we're going to remove the USB Controller and the Sound Card. In the Device list of the Hardware window, click on the USB Controller to highlight it, then click the "Remove" button underneath the device list.



Device	Summary
Memory	512 MB
Processors	1
New CD/DVD (...	Using file D:\ISOs\Nix\pfSense-CE...
Network Adapter	Bridged (Automatic)
Network Adapt...	Custom (VMnet1)
Network Adapt...	Custom (VMnet2)
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Connections

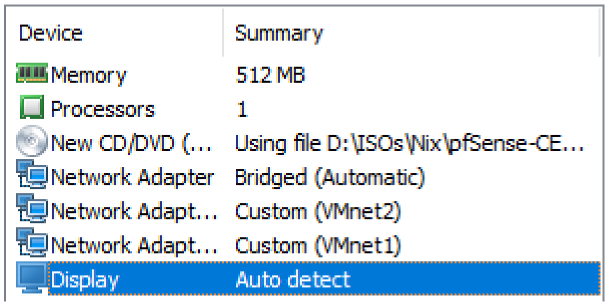
USB compatibility: USB 2.0

[✓] Automatically connect new USB devices

[ ] Show all USB input devices

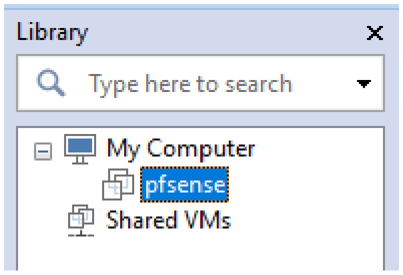
[ ] Share Bluetooth devices with the virtual machine

Repeat this process for the Sound Card to remove that device as well.
After you are finished, your hardware configuration should look like this:



Device	Summary
Memory	512 MB
Processors	1
New CD/DVD (...	Using file D:\ISOs\Nix\pfSense-CE...
Network Adapter	Bridged (Automatic)
Network Adapt...	Custom (VMnet2)
Network Adapt...	Custom (VMnet1)
Display	Auto detect

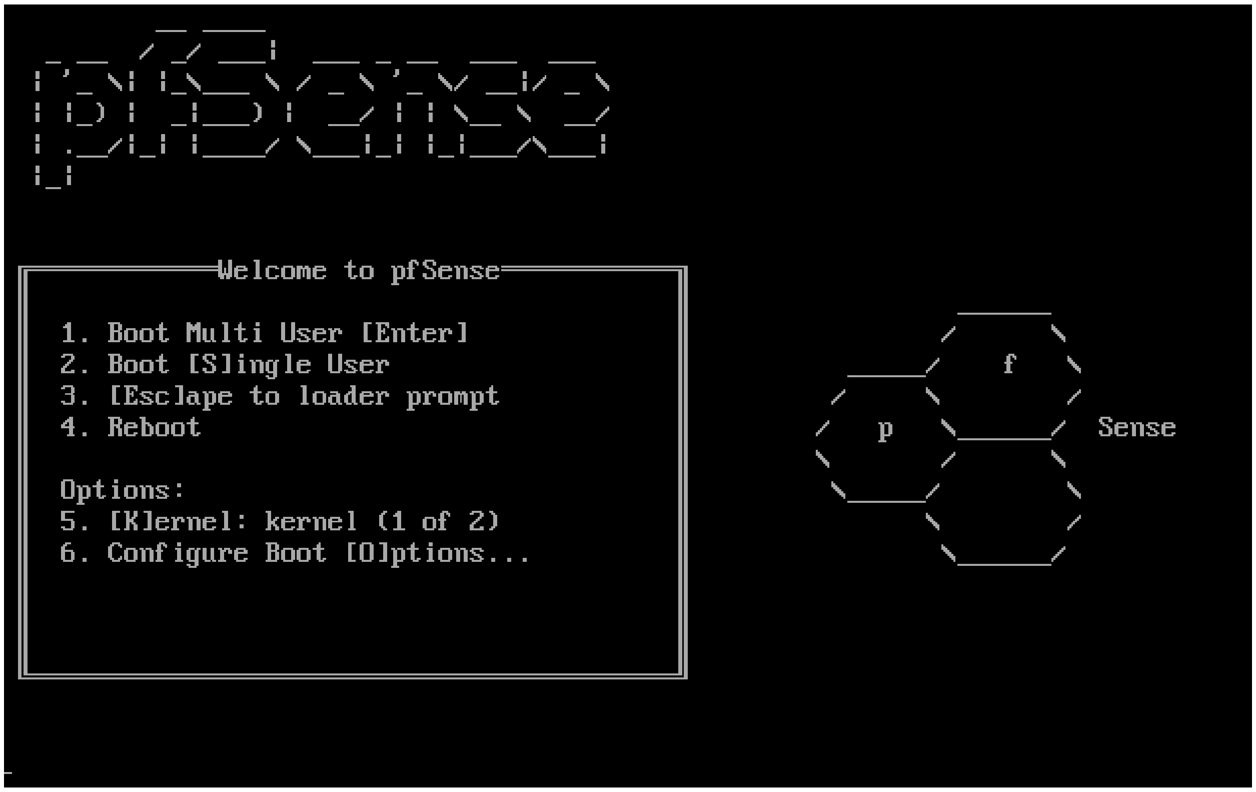
Click the Close button in the lower right corner of the Hardware window to exit. Back in the New Virtual Machine Wizard, click Finish to let VMware Workstation create the VM. The system will take a moment or two to allocate the disk space to the virtual machine. When it is finished, the main window of VMware Workstation should list the pfSense VM under the “Library” pane.



Installing pfSense

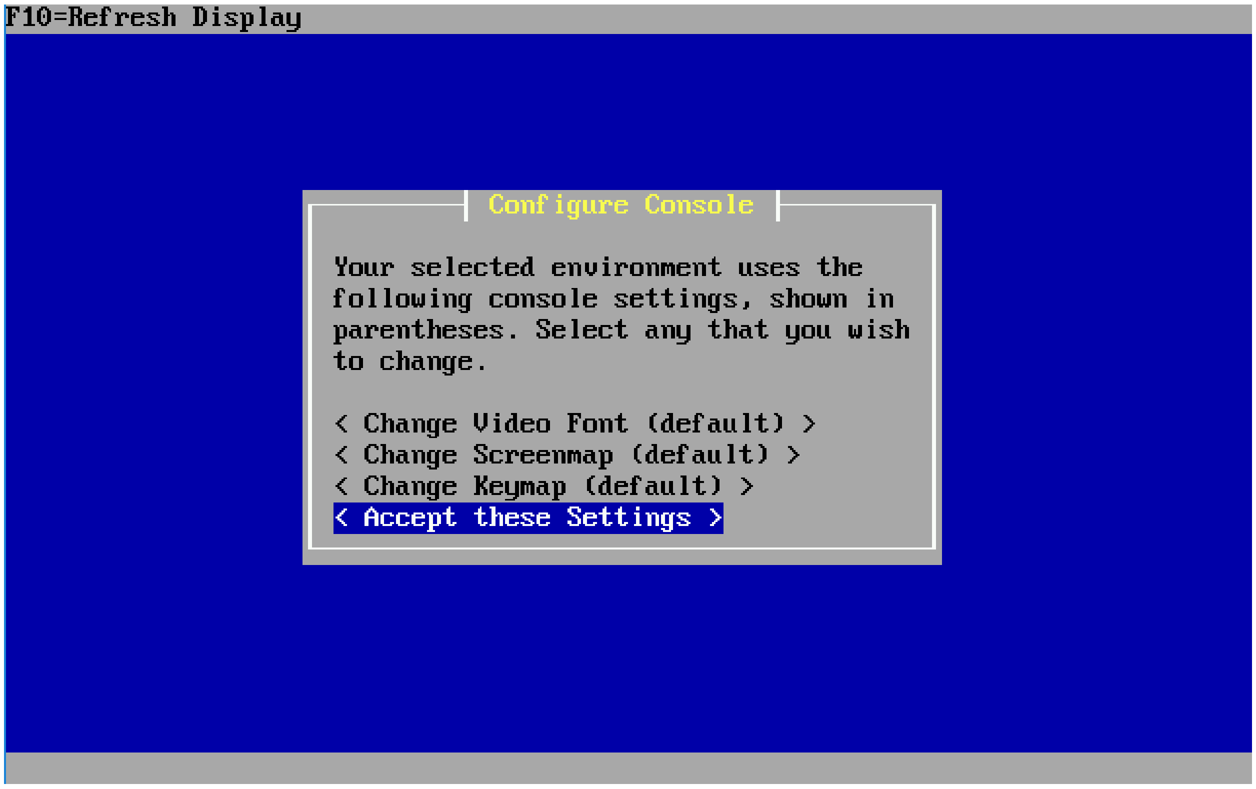
At this point, you're now ready to install pfSense on your VM. To start the virtual machine, in the VMware Workstation main menu, click on pfSense in the "Library" pane, then select "Power on this virtual machine" in the main window. After doing so, the primary window will switch to view the console of the powered on VM. Clicking on this console will attach your mouse and keyboard to this virtual machine; to detach the devices again, press ctrl+alt.

.



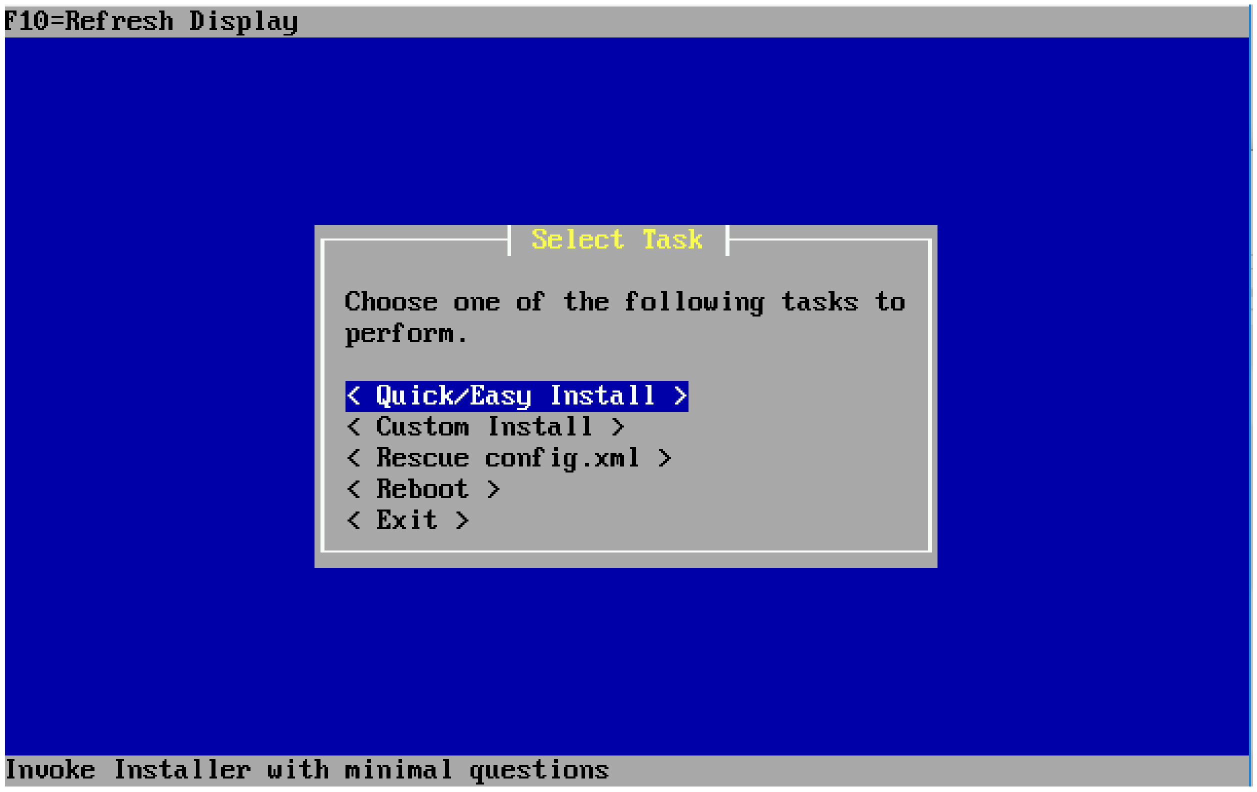
```
Welcome to pfSense

1. Boot Multi User [Enter]
2. Boot [S]ingle User
3. [Esc]ape to loader prompt
4. Reboot

Options:
5. [K]ernel: kernel (1 of 2)
6. Configure Boot [O]ptions...
```

Let pfSense boot up, and the system should automatically run the installer. Adjust your video, screenmap and keymap settings as necessary, then select “< Accept these Settings >”.



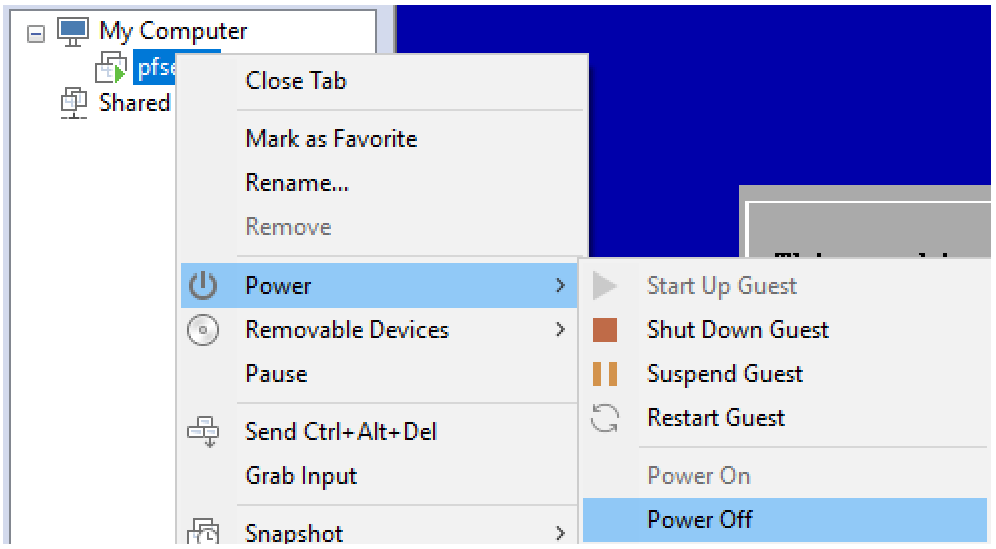
```
F10=Refresh Display

                    Configure Console

   Your selected environment uses the
   following console settings, shown in
   parentheses. Select any that you wish
   to change.

   < Change Video Font (default) >
   < Change Screenmap (default) >
   < Change Keymap (default) >
   < Accept these Settings >
```

On the next screen, select “< Quick/Easy Install >” and let pfSense do all the heavy lifting. The next screen will inform you that the install will erase the contents of the hard disk. Since our virtual disk is already empty, this doesn’t matter in the least. Select OK, and let the installer run.



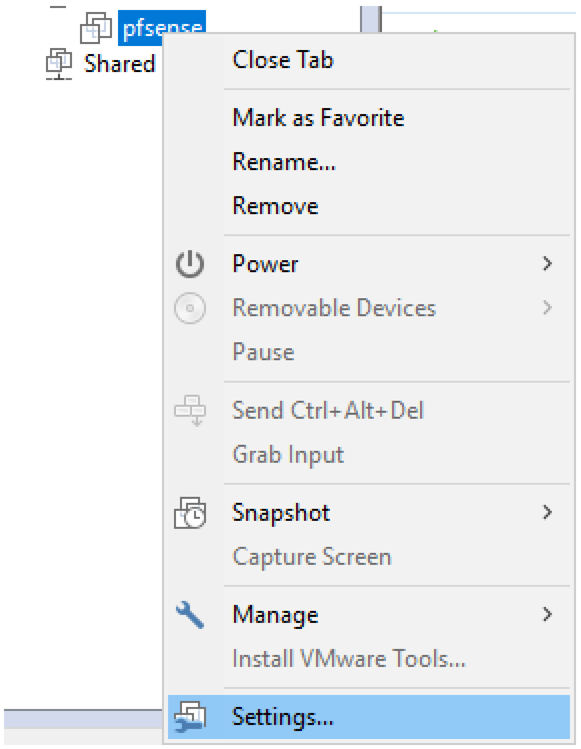
```
F10=Refresh Display

                Select Task

   Choose one of the following tasks to
   perform.

   < Quick/Easy Install >
   < Custom Install >
   < Rescue config.xml >
   < Reboot >
   < Exit >

Invoke Installer with minimal questions
```

The installer will go through and install pfSense to the virtual hard disk. At a certain point you will be asked whether the standard kernel or the embedded one should be used. Make sure to select < *Standard Kernel* >. Finally, the installer informs you to reboot the machine to boot from the hard drive. The installation is done, however, we're not going to reboot. Right click on pfSense in the *Library* pane, select *Power*, and *Power Off* to power off the VM.



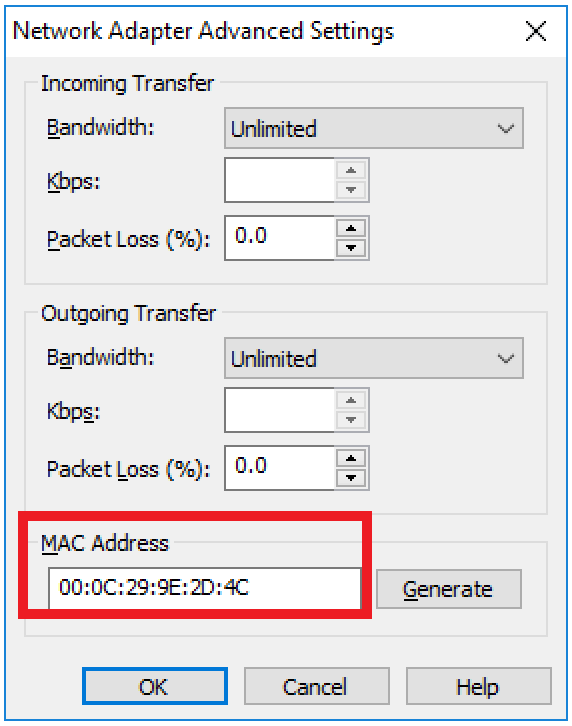
Final VM Settings

Now that pfSense is installed, we no longer need the CD/DVD drive attached to the VM, so we're going to remove it. On the VMware Workstation main screen, right click on pfSense in the "Library" pane, then select "Settings..."



In the *Hardware* tab, in the *Devices* pane, select *CD/DVD (IDE)*, then click on the Remove button to remove the CD/DVD drive, just like we did with the sound card and the USB controller earlier.

Next, select *Network Adapter* in the *Device* pane, then click on the *Advanced…* button to bring up the *Network Adapter Advanced Settings* menu. Document the contents of the *MAC Address* section of this screen; click OK when you are done. Repeat this process for *Network Adapter 2* and *Network Adapter 3*. The goal is for you to know which MAC address corresponds to which VMnet (e.g. *Bridged*, *VMnet1* (*Management*), and *VMnet2* (*IPS 1*)) as we need this information to properly configure the network settings in pfSense later.

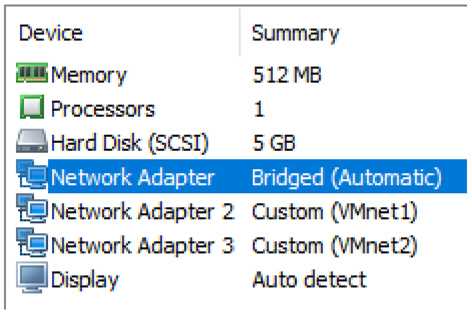

Network Adapter Advanced Settings

Incoming Transfer
Bandwidth: Unlimited
Kbps:
Packet Loss (%): 0.0

Outgoing Transfer
Bandwidth: Unlimited
Kbps:
Packet Loss (%): 0.0

MAC Address
00:0C:29:9E:2D:4C

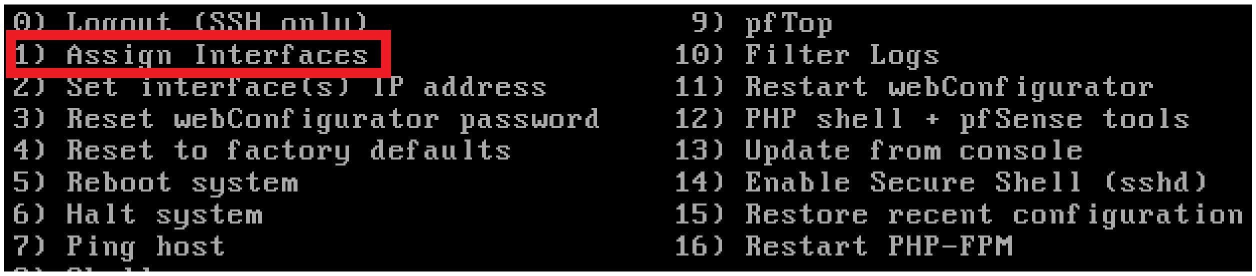
Click OK to exit the Virtual Machine Settings menu. Your VM's Device listing should look like this:



Device	Summary
Memory	512 MB
Processors	1
Hard Disk (SCSI)	5 GB
Network Adapter	Bridged (Automatic)
Network Adapter 2	Custom (VMnet1)
Network Adapter 3	Custom (VMnet2)
Display	Auto detect

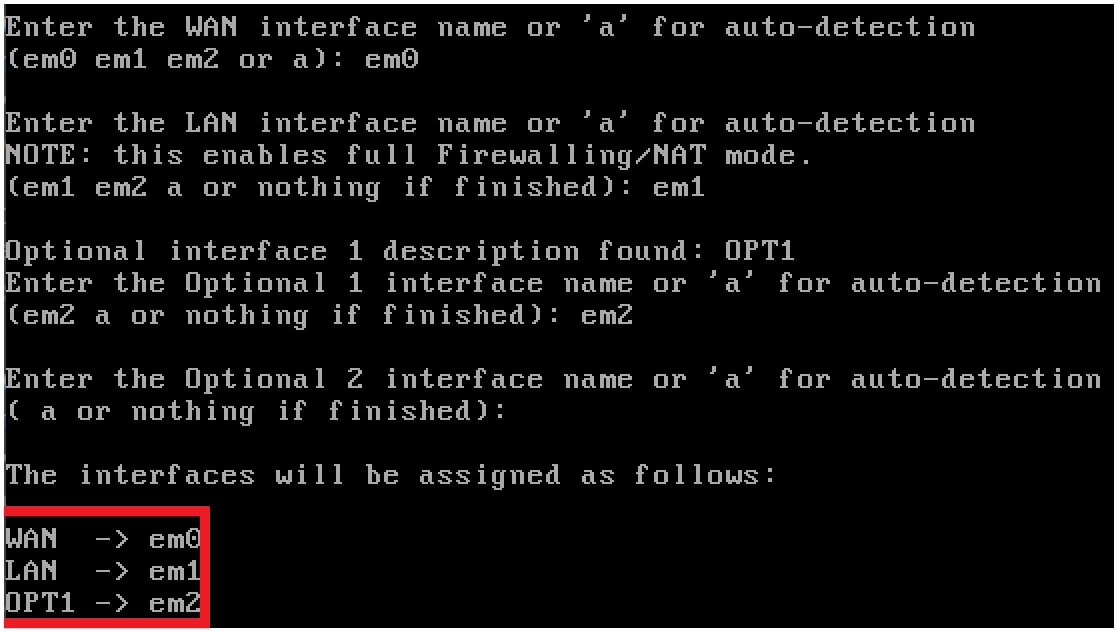
Network Configuration

Power on the pfSense VM, and allow it to boot up. Eventually you'll be greeted with the main pfSense menu on the console with 16 options. Select option 1, "Assign Interfaces" to run the interface assignment wizard.



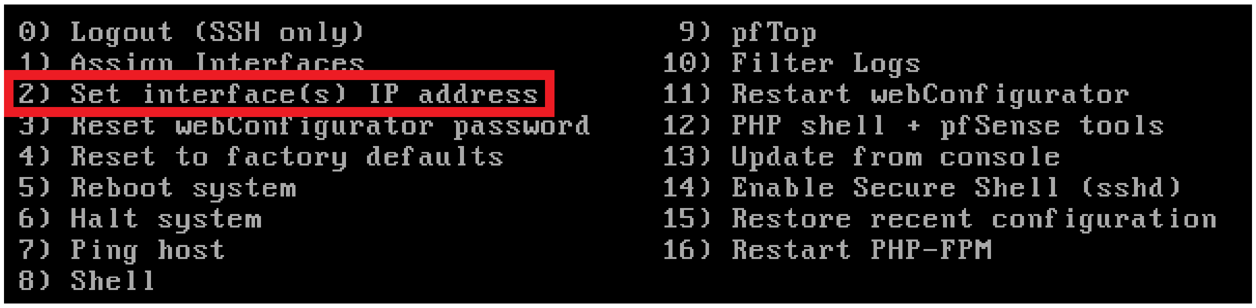
```
 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
```

The WAN interface needs to be assigned to the bridged network adapter (Network Adapter, the Bridged network). The LAN interface needs to be assigned to our Host-Only adapter (Network Adapter 2, the Management network).Finally, OPT1 needs to be assigned to our internal network adapter (Network Adapter 3, the IPS 1 network). In my case, em0 became the WAN interface, em1 the LAN interface, and em2 was mapped to the OPT1 interface. **Be sure to pay attention to the MAC addresses for em0, em1, and em2, and correlate that to the MAC addresses you documented earlier for Network Adapter, Network Adapter 2, and Network Adapter 3.** After confirming that the settings are correct, you should automatically be returned to the pfSense main menu.



```
Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): em0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): em1

Optional interface 1 description found: OPT1
Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished): em2

Enter the Optional 2 interface name or 'a' for auto-detection
( a or nothing if finished):

The interfaces will be assigned as follows:

WAN  -> em0
LAN  -> em1
OPT1 -> em2
```

Now that we have assigned the network interfaces, we have to configure IP addresses for these interfaces. In most cases, the “WAN” interface (bridged network) will automatically get an IP address from the device on your physical network that provides DHCP services. If this is not happening, you may have to troubleshoot your physical network. This is beyond the scope of our guide. As for LAN and OPT1, we have to manually set the IP address, subnet mask, and DHCP scopes for these networks. Select option 2 in the pfSense main menu to get started.



```
 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell
```

The configuration wizard starts by asking for the interface the settings should be changed for. We have to go through the process two times, for the LAN and the OPT1 interface individually. Here are the settings I recommend:

LAN IP: **172.16.1.1**
LAN Subnet bit count: **24**
LAN upstream gateway address: **<empty>**
LAN IPv6 address: **<empty>**
Do you want to enable the DHCP server on LAN? **y**
LAN DHCP start address: **172.16.1.10**

LAN DHCP end address: **172.16.1.254**
Do you want to revert to HTTP as the webConfigurator protocol? **N**

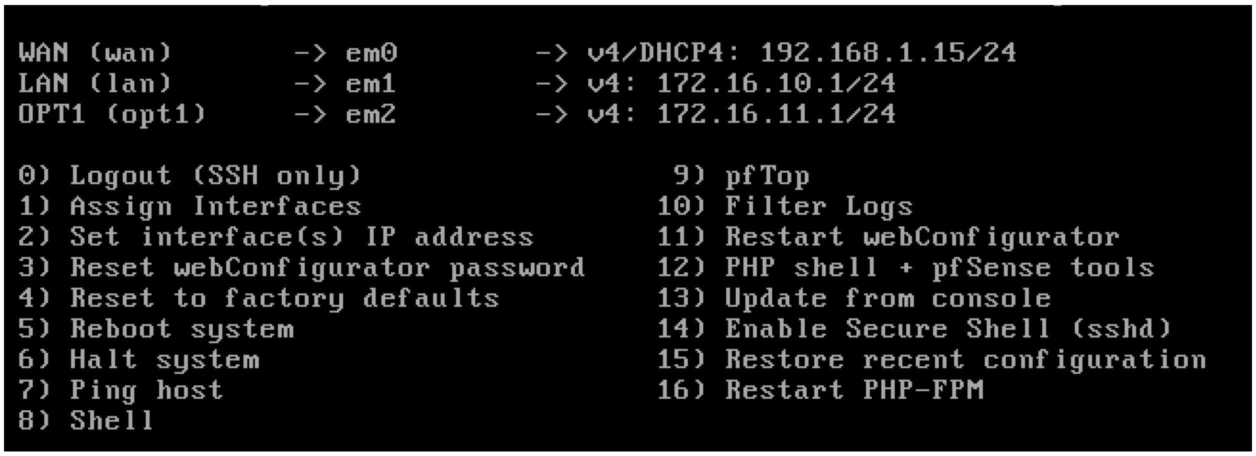
OPT1 IP: **172.16.2.1**
OPT1 Subnet bit count: **24**
OPT1 upstream gateway address: **<empty>**
OPT1 IPv6 address: **<empty>**
Do you want to enable the DHCP server on OPT1? **y**
OPT1 DHCP start address: **172.16.2.10**
OPT1 DHCP end address: **172.16.2.254**
Do you want to revert to HTTP as the webConfigurator protocol? **n**

The reason we choose 1.10 and 2.10 as the DHCP start addresses is to reserve a few IP addresses for static DHCP allocations. Static DHCP allocations allow you to configure a DHCP server to always serve the same IP address when requested from a particular MAC address. We'll talk about this a little bit more later. For now, we are done mucking around in the pfSense CLI. From here on out, we will be using the webConfigurator to manage and configure pfSense.

Note: If you are using 172.16.1.0/24 or 172.16.2.0/24 in your physical network, choose another network range to assign to the LAN and OPT1 interfaces to avoid network conflicts.

Note: When you are configuring the LAN and OPT1 interface IP addresses, the configuration script will ask "Do you want to revert to HTTP as the webConfigurator protocol?" **Always say no to this.** This allows the web UI to accept HTTP logins, and makes your firewall credentials vulnerable to sniffing over the network.

Your pfSense main menu should look something like this when you're all done:



```
WAN (wan)       -> em0         -> v4/DHCP4: 192.168.1.15/24
LAN (lan)       -> em1         -> v4: 172.16.10.1/24
OPT1 (opt1)     -> em2         -> v4: 172.16.11.1/24

0) Logout (SSH only)                  9) pfTop
1) Assign Interfaces                 10) Filter Logs
2) Set interface(s) IP address       11) Restart webConfigurator
3) Reset webConfigurator password    12) PHP shell + pfSense tools
4) Reset to factory defaults         13) Update from console
5) Reboot system                     14) Enable Secure Shell (sshd)
6) Halt system                       15) Restore recent configuration
7) Ping host                         16) Restart PHP-FPM
8) Shell
```

If so, let's move on to setting up pfSense from the webConfigurator. Before doing so however, you must configure the Host-Only adapter (aka the management network) on your host. If you are using Windows as your host, and If you haven't already, you'll want to visit the "Unbinding Network Protocols on Windows Virtual Adapters" at a minimum. This document will guide you through configuring the VMnet1 Adapter on Windows with an IP address as well as unbinding network protocols on this adapter to increase the security of your hypervisor host. You may also want to consider "Using Windows Firewall to Limit Exposure of Windows Hypervisor Hosts" to further enhance the security of the Windows host when interacting with your lab VMs.

Linux users can configure the IP address of their host-only network card by using the command utilities `ifconfig` or `ip addr add`. Try one of the following commands to set the IP address of the vboxnet0 interface:

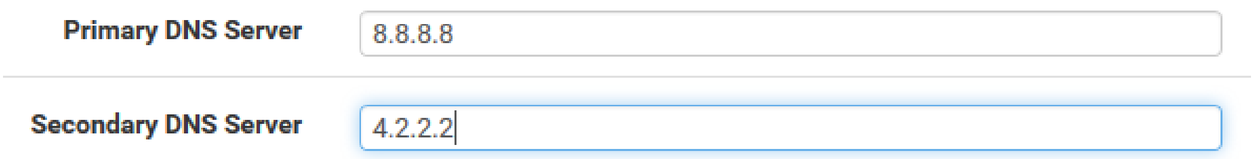
```
ifconfig vmnet1 172.16.1.2 netmask 255.255.255.0
ip addr add 172.16.1.2/24 dev vmnet1
```

Please note that you will need root permissions to run these commands, and that these settings will NOT persist between reboots of your hypervisor host system. Configuring IP address persistence for your host OS is beyond the scope of this guide.

webConfigurator - Initial Setup

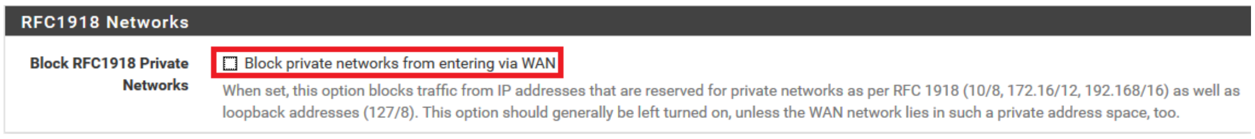
On the host OS, open your favorite web browser (I prefer Firefox - https://www.mozilla.org/en-US/firefox/new/) and navigate to https://172.16.1.1 (or the IP address you assigned to the LAN interface of your pfSense system). **The default credentials for access to the web interface are admin/pfsense.** If this is your first time logging in, pfSense takes you through a nice little setup wizard. I'll highlight some of the important things to take note of, and/or change as necessary:

Set the primary and secondary DNS servers you plan on using. I typically use 8.8.8.8 (Google public DNS) and 4.2.2.2 (Level 3 public DNS). If you're using this lab at work, your workplace may have restrictions and you may have to use their DNS servers instead.



Primary DNS Server	8.8.8.8
Secondary DNS Server	4.2.2.2

Uncheck the checkbox next to "Block private networks from entering via WAN" rule. Uncheck the checkbox next to "Block Bogon Networks" on this page as well.



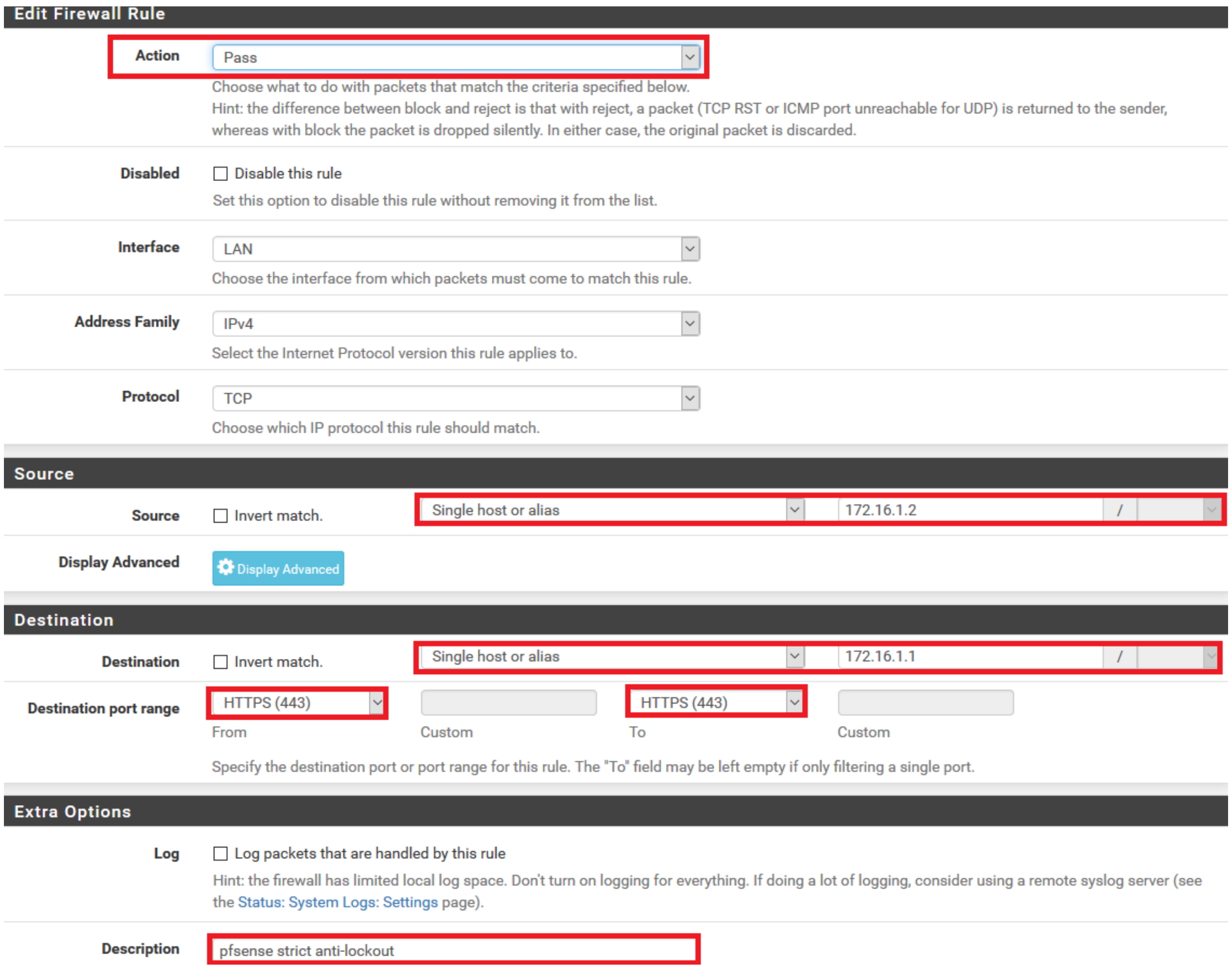
RFC1918 Networks

Block RFC1918 Private Networks ☐ Block private networks from entering via WAN

When set, this option blocks traffic from IP addresses that are reserved for private networks as per RFC 1918 (10/8, 172.16/12, 192.168/16) as well as loopback addresses (127/8). This option should generally be left turned on, unless the WAN network lies in such a private address space, too.

The other default settings on the first time setup wizard should be fine. Please note that you will be asked to change the password for the admin account as a part of the setup wizard. Document the password, and keep it somewhere safe.

Our next order of business is to restrict access to the web interface of the firewall to the machine with the IP address 172.16.1.2. On the menu bar at the top of the page, select *Firewall > Rules*. Click on *LAN* to modify the firewall policy for the *LAN* interface. Click the *Add* button with the arrow facing up. This will add the firewall rule to the top of the firewall policy to where it will evaluated first. The *Edit Firewall Rule* page is fairly straightforward. I've highlighted the options to be aware of.



Edit Firewall Rule

Action	Pass Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.
Disabled	☐ Disable this rule Set this option to disable this rule without removing it from the list.
Interface	LAN Choose the interface from which packets must come to match this rule.
Address Family	IPv4 Select the Internet Protocol version this rule applies to.
Protocol	TCP Choose which IP protocol this rule should match.

Source

Source	☐ Invert match. Single host or alias 172.16.1.2 /
Display Advanced	Display Advanced

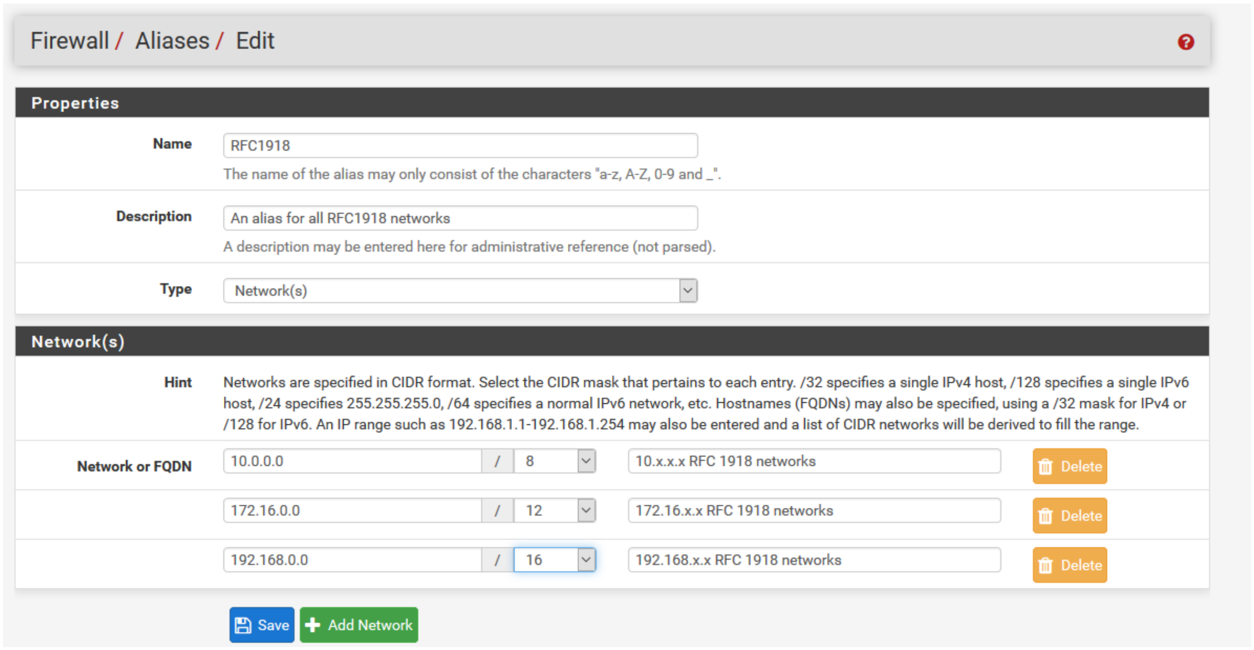
Destination

Destination	☐ Invert match. Single host or alias 172.16.1.1 /
Destination port range	From: HTTPS (443) Custom: To: HTTPS (443) Custom: Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.

Extra Options

Log	☐ Log packets that are handled by this rule Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the Status: System Logs: Settings page).
Description	pfsense strict anti-lockout

When you are done, click the *Save* icon at the bottom of the page. This will take you back to the previous page. A yellow dialogue box with a green button called *Apply Changes* will appear. Click this button to apply this new firewall rule. Next, we want to an alias. Navigate to *Firewall > Aliases*. On the Firewall Aliases page, click on *IP*, then click *Add*. Create an alias with the following settings:



Firewall / Aliases / Edit

Properties

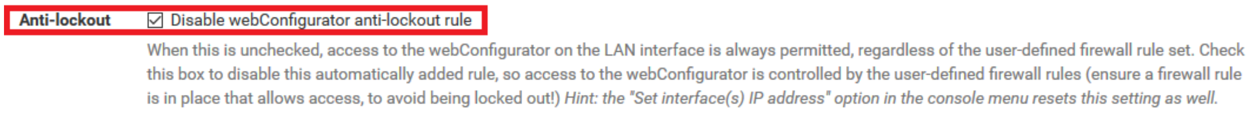
Name	RFC1918 — The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".
Description	An alias for all RFC1918 networks — A description may be entered here for administrative reference (not parsed).
Type	Network(s)

Network(s)

Hint: Networks are specified in CIDR format. Select the CIDR mask that pertains to each entry. /32 specifies a single IPv4 host, /128 specifies a single IPv6 host, /24 specifies 255.255.255.0, /64 specifies a normal IPv6 network, etc. Hostnames (FQDNs) may also be specified, using a /32 mask for IPv4 or /128 for IPv6. An IP range such as 192.168.1.1-192.168.1.254 may also be entered and a list of CIDR networks will be derived to fill the range.

Network or FQDN	Mask	Description
10.0.0.0	8	10.x.x.x RFC 1918 networks
172.16.0.0	12	172.16.x.x RFC 1918 networks
192.168.0.0	16	192.168.x.x RFC 1918 networks

Click Save to be brought back to the previous page, then click *Apply Changes*. We just created an alias for RFC1918 networks (local networks that are not routable through the public internet) This will come in handy later for creating firewall rules around these networks. Our final action for now is to navigate to *System > Advanced*, and click to fill in the checkbox next to the option *Disable webConfigurator anti-lockout rule*, and click *Save* on the bottom of the page. **Be aware that if you did not create the firewall rule above, this will lock you out of the webConfigurator as soon as you click *Save*.**

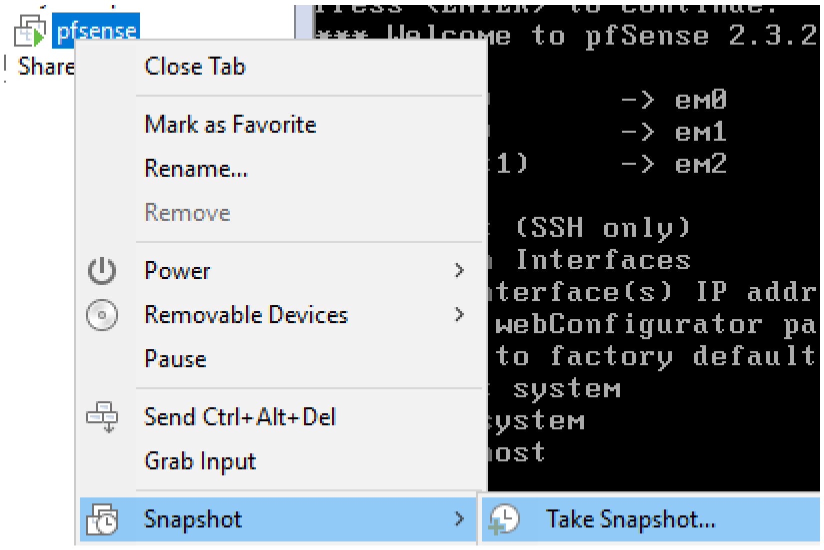


Anti-lockout ☑ Disable webConfigurator anti-lockout rule

When this is unchecked, access to the webConfigurator on the LAN interface is always permitted, regardless of the user-defined firewall rule set. Check this box to disable this automatically added rule, so access to the webConfigurator is controlled by the user-defined firewall rules (ensure a firewall rule is in place that allows access, to avoid being locked out!) *Hint: the "Set interface(s) IP address" option in the console menu resets this setting as well.*

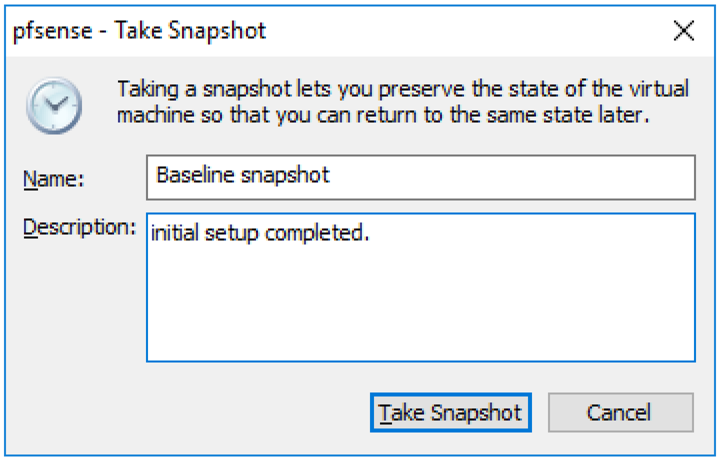
Take a Snapshot

Snapshotting is a VERY important concept with VMs. It lets you restore your virtual machines to its state in the past when the snapshot was taken. This allows you to undo misconfigurations, solve problems or potential issues that might affect the VM, in relatively easily fashion. We're going to take a snapshot of the pfSense virtual machine in its current state. In the VMware Workstation main screen, in the "Library" pane, right click on pfSense, click Snapshot > Take Snapshot…



```
*** Welcome to pfSense 2.3.2
                -> em0
                -> em1
 1)             -> em2

  (SSH only)
  Interfaces
 terface(s) IP addr
 webConfigurator pa
 to factory default
  system
 system
 ost
```

A new window pops up, offering two input boxes: one to name the snapshot, and another to enter a description. I named the snapshot "baseline" and gave a brief description of what the snapshot is for. If you keep multiple snapshots make sure to use good descriptions that include <u>WHEN</u> the snapshot was taken, and <u>WHAT STATE</u> the VM is in when you took the snapshot. After you are finished, click the "Take Snapshot" button. VMware Workstation will begin generating the snapshot for your virtual machine immediately, and should be finished in moments.


pfsense - Take Snapshot

Taking a snapshot lets you preserve the state of the virtual machine so that you can return to the same state later.

Name: Baseline snapshot

Description: initial setup completed.

If you ever have to restore a VM from a snapshot, just right click on the virtual machine in the Library pane and navigate to the Snapshot sub-menu. You will be offered the option to either revert to the most recent snapshot, or to open the Snapshot Manager where you can choose a specific snapshot you want to revert to. The restoration will take a moment or two, and voila, your VM is back in state it was in when the snapshot was taken.

pfSense Summary

At the end of all this, you should have a pfSense VM with the following settings:

- 512MB of RAM
- 5GB of disk
- 1 Virtual CPU
- DVD drive disabled
- Audio disabled
- USB controller disabled
- 3 Network interfaces:
    - 1 Interface connected to the VMnet0 (“Bridged”) network (WAN interface)
    - 1 Interface connected to the VMnet1 (“Management”) network (LAN interface)
    - 1 Interface connected to VMnet2 (“IPS 1”) network (OPT1 interface)

pfSense should be configured as followed:

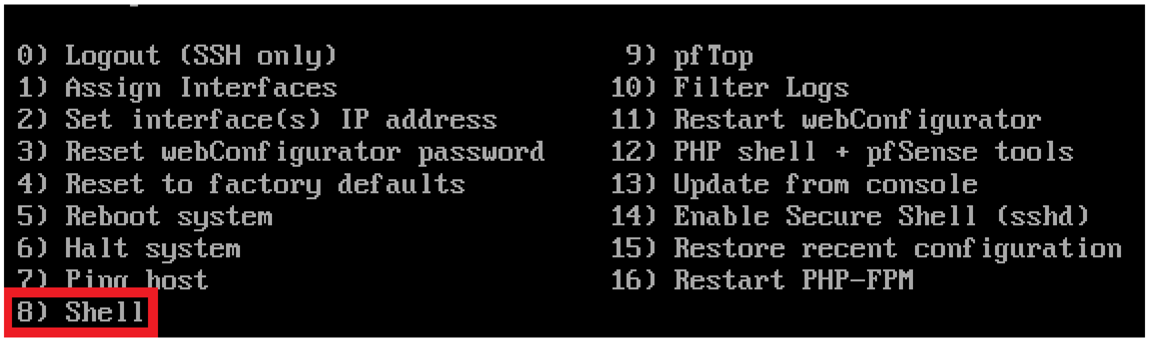
- The WAN interface should be the network adapter connected to the Bridged VMnet. This interface should automatically get an IP address via DHCP. If this isn’t happening, get some help with whoever administers your local network, or start troubleshooting. **Alternatively, configure a static IP address**
- The LAN interface should have an IP address of 172.16.1.1.
- The LAN network should be 172.16.1.0 with a /24 (255.255.255.0) subnet mask
- DHCP should be configured for this network with a scope of 172.16.1.10-172.16.1.254
    - 172.16.1.3-172.16.1.9 are available for setting static DHCP mappings
- The OPT1 interface should have an IP address of 172.16.2.1
- The OPT1 network should be 172.16.2.0 with a /24 (255.255.255.0) subnet mask
- DHCP should be configured for this network with a scope of 172.16.2.10-172.16.2.254
    - 172.16.2.2-172.16.2.9 are available for static DHCP mappings.
- Your host OS accesses pfSense from the Management interface. You should have one virtual adapter connected to VMnet1 (the one we will be using) and VMnet8 (which will NEVER be used. You can delete OR disable the interface in your host OS.). The VMnet1 adapter should have an IP address of 172.16.1.2. We created a firewall rule that serves as our anti-lockout rule for allowing the host OS to access pfSense via the web interface over HTTPS.
- You should have an alias for RFC1918 networks configured for use with the firewall policy.
- You should have at least one good snapshot you can revert to if you need to redo a step or somehow got lost/confused.

What's Next?

Now that you have an operational pfSense VM, your next steps are to read the pfSense Firewall Rules and Network Services Guide. Make sure the firewall policies match those illustrated in the segmentation guide, and make sure that the pfSense VM is hosting all the services mentioned in the Core Network Services guide to make your firewall fully functional and ready to handle your lab network.

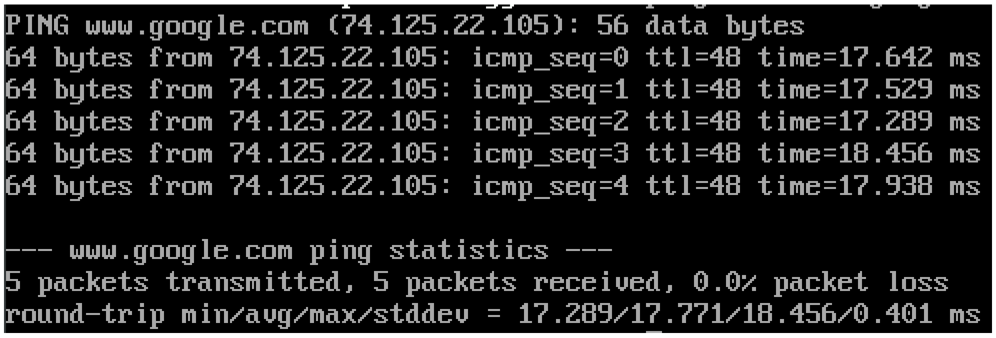
Final Connectivity Checks and Troubleshooting

Once you have finished setting up your pfSense VM according to the instructions laid out, Open a console session to the pfSense VM, and attach your mouse and keyboard by clicking on the console. If the screen is blank, hit the *Enter* key to bring up the pfSense main menu. Select option *8) Shell* to start a shell session on pfSense.



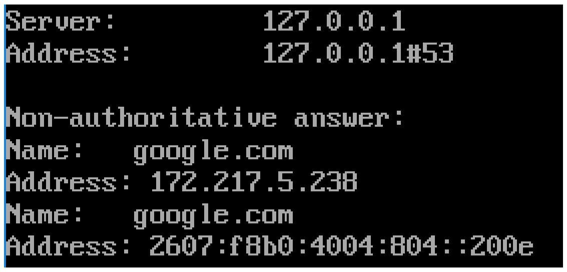
```
0) Logout (SSH only)                  9) pfTop
1) Assign Interfaces                 10) Filter Logs
2) Set interface(s) IP address       11) Restart webConfigurator
3) Reset webConfigurator password    12) PHP shell + pfSense tools
4) Reset to factory defaults         13) Update from console
5) Reboot system                     14) Enable Secure Shell (sshd)
6) Halt system                       15) Restore recent configuration
7) Ping host                         16) Restart PHP-FPM
8) Shell
```

The first, run the command `ping -c 5 www.google.com`



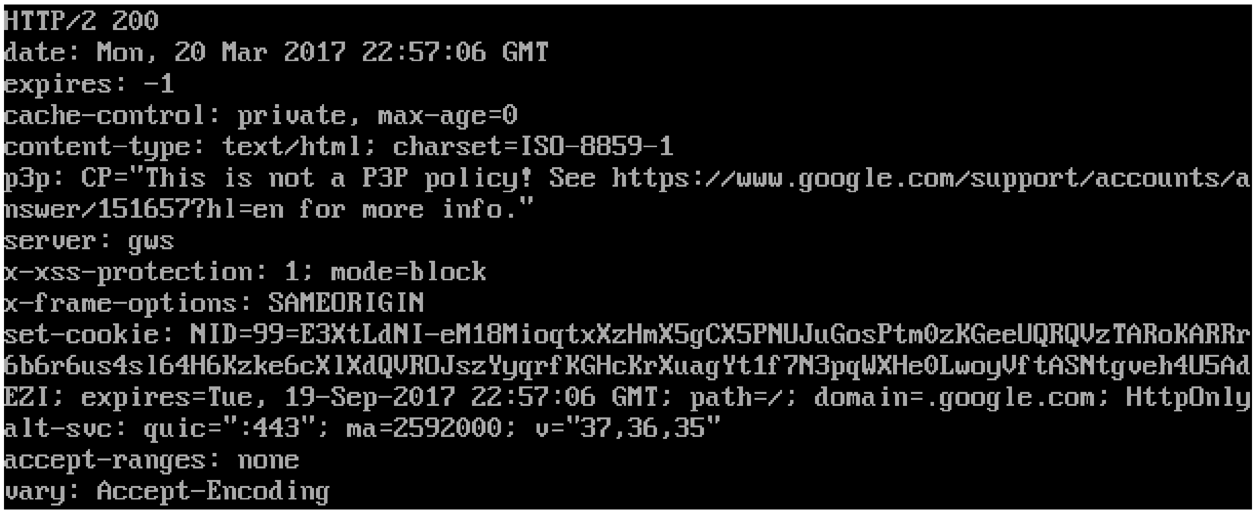
```
PING www.google.com (74.125.22.105): 56 data bytes
64 bytes from 74.125.22.105: icmp_seq=0 ttl=48 time=17.642 ms
64 bytes from 74.125.22.105: icmp_seq=1 ttl=48 time=17.529 ms
64 bytes from 74.125.22.105: icmp_seq=2 ttl=48 time=17.289 ms
64 bytes from 74.125.22.105: icmp_seq=3 ttl=48 time=18.456 ms
64 bytes from 74.125.22.105: icmp_seq=4 ttl=48 time=17.938 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 17.289/17.771/18.456/0.401 ms
```

The output from the `ping` command should look similar to the output above. Next, run the command `nslookup google.com`



```
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.5.238
Name:   google.com
Address: 2607:f8b0:4004:804::200e
```

Again, you should have output similar to the illustration above. Finally, run `curl -I https://www.google.com`



```
HTTP/2 200
date: Mon, 20 Mar 2017 22:57:06 GMT
expires: -1
cache-control: private, max-age=0
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See https://www.google.com/support/accounts/a
nswer/151657?hl=en for more info."
server: gws
x-xss-protection: 1; mode=block
x-frame-options: SAMEORIGIN
set-cookie: NID=99=E3XtLdNI-eM18MioqtxXzHmX5gCX5PNUJuGosPtm0zKGeeUQRQVzTARoKARRr
6b6r6us4sl64H6Kzke6cXlXdQVROJszYyqrfKGHcKrXuagYt1f7N3pqWXHe0LwoyVftASNtgveh4U5Ad
EZI; expires=Tue, 19-Sep-2017 22:57:06 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: quic=":443"; ma=2592000; v="37,36,35"
accept-ranges: none
vary: Accept-Encoding
```

The output of your `curl` command should be similar to the screen capture above. If so, you may type `exit` to leave the shell, and close your connection to the pfSense console. If you had any problems with any of the commands above, you've got troubleshooting to do. When it comes to troubleshooting, start at layer 1 of the OSI model (Physical) and work your way up. Here are some questions to help the troubleshooting process:

- Which of the commands above failed? Did they all fail?
- Have you checked your physical cabling or wireless network connectivity?
- Does your company/LAN have MAC filtering/port security on? You might need to modify switch settings for your drop, or talk to your network administrator.
- Have you checked your upstream gateway/SOHO router? Trying pinging the gateway's IP address to see if your pfSense VM can reach the gateway. Try running the `traceroute` command to google.com to see if and where your ping command is failing.

- Do you have an upstream firewall in place? Is 443/TCP allowed outbound? Is 53/UDP allowed outbound? Is ICMP allowed?
- Did you input the commands correctly? Try entering them again. No, I’m not patronizing you.

Once you have verified that your firewall’s connectivity, policies, and services are properly configured, **SNAPSHOT THE VM <u>BEFORE</u>** moving on and trying your hand at setting up the remaining VMs.

Your Turn

The pfSense virtual machine required a lot of custom configuration, both in VMware Workstation, as well as in the VM itself once the OS was installed. Now that we did that together, I'm going to have you create the rest of the virtual machines yourself -- with the exception of the Metasploitable 2 VM since it's a special case for reasons you will see momentarily. Below are a set of spec sheets to help you create the remaining virtual machines for the lab on your own. Think of them as checklists you should be able to run through unaccompanied.

All of our Linux-based virtual machines are built on Debian Linux based distros (both, Ubuntu and Kali Linux are derived from Debian), so setup and configuration of all of our lab VMs should be very similar.

Kali Linux VM

Kali Linux is a popular penetration testing distro. Popular because hackers and script kiddies are lazy, and practically everything you need for performing a penetration test or joining anonymous is installed by default. Kali is going to be our loud, obnoxious attacker that we're going to use as a noise generator for the express purpose of generating events on our IPS VM. I want you to perform the following tasks:

- Download Kali Linux 64-bit here: https://www.kali.org/downloads/
- Create a VM with the following settings:
    - Configure the VM to install from the Kali Linux ISO you downloaded
    - Set the Guest OS family to "Linux"
    - Set the Guest OS version to "Debian 8.x 64-bit
    - Name the VM "Kali"
    - Allocate 80 GB of space for the disk
        - Choose the "store virtual disk as a single file" option
- Customize the following hardware settings
    - Allocate 4GB (4096MB) of RAM
    - Allocate 1-2 processors, where total number of cores is no more than 2
    - Connect the VM's network adapter to VMnet2
        - Ensure that "Connect at power on" is checked
    - Remove the USB Controller
    - Remove the Sound Card
    - Remove the Printer
- Install Kali Linux
    - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the

installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.2.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy

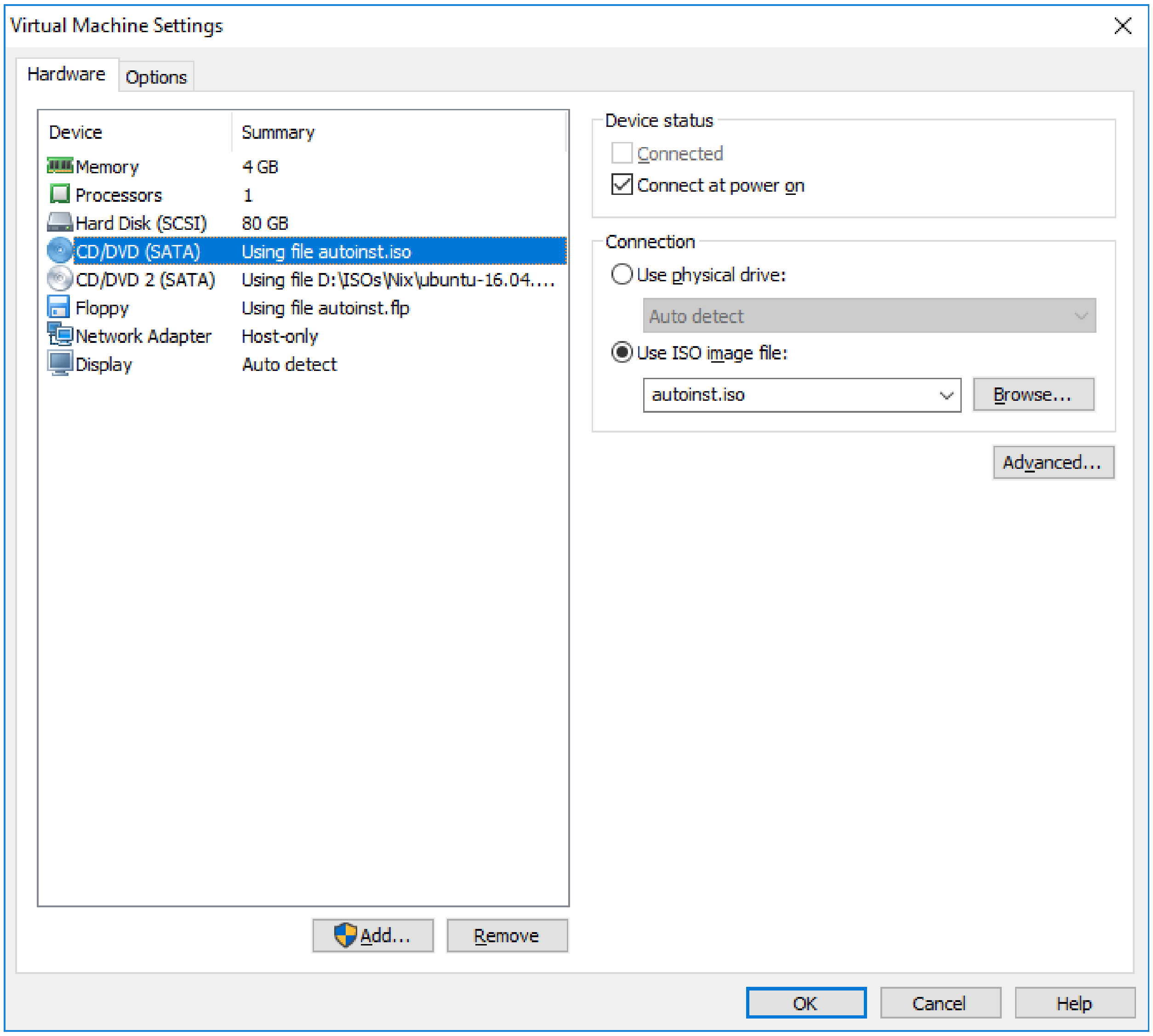
- After the install is completed, power off the VM and adjust the following settings:
  - Remove the CD/DVD drive
  - Make sure to document the MAC address under advanced settings for the network adapter.
- Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the OPT1 interface DHCP, for the MAC address of adapter 1 for the Kali VM; assign it the IP address 172.16.2.2, making sure to enter a description that tells you which VM this static mapping belongs to
- Power on the virtual machine and do the following:
  - Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense
    - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
  - Verify the virtual machine can reach the internet
    - In a terminal/shell run the command `curl -I https://www.google.com`
      - This will confirm DNS can resolve domain names, and that the VM can reach the internet
      - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense
        - **Note:** The `ping` command, in this case, will work from this virtual machine, due to the unique firewall policy on the OPT1 network
  - Update/patch the VM
    - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
      - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
  - Create a snapshot for the VM

Now you should have a functional Kali Linux VM. At this point, the virtual machine should be fully operational, and you should be able to control it from the VM's console. However, if you want to enable and configure SSH access to this host, I have configured a guide for doing so. Jump to "How to Enable SSH on Kali Linux" to enable the ssh service for this VM.

SIEM VM

SIEM is shorthand for Security Intrusion Events Manager. This is fancy security nomenclature for “log aggregator”; we will be having our IPS VM log its events here. We’re going to be running Splunk on this VM. Splunk is a commercial program for managing logs on a pretty large scale. By default, and with no licensing, Splunk only allows you to collect or “index” 500MB worth of logs per day however, there are /ways/ around this that we’ll talk about later. I want you perform the following tasks to set up the SIEM VM:

- Download Ubuntu Server 16.04.1 LTS 64-bit here: https://www.ubuntu.com/download/server
- Create a VM with the following settings:
    - Configure the VM to install from the Ubuntu Server ISO you downloaded
    - Ubuntu supports VMware’s “Easy Install” option, but we’re not going to use it
        - Fill out whatever data you want in the full name, username, password and confirm fields
    - Name the VM “siem”
    - Allocate 80 GB of space for the disk
        - Choose the “store virtual disk as a single file” option
- Customize the following hardware settings
    - Allocate 4GB (4096MB) of RAM
    - Allocate 1-2 processors, where total number of cores is no more than 2
    - Connect the VM’s network adapter to “Host-only” (VMnet1)
        - Ensure that “Connect at power on” is checked
    - Remove the USB Controller
    - Remove the Sound Card
    - Remove the Printer
- **Before powering on the VM to install Ubuntu Server...**
    - Enter the *Virtual Machine Settings* under the *Hardware* tab. Remove the CD/DVD drive with the file named `autoinst.iso`, as well as the Floppy drive with the file named `autoinst.flp`


Virtual Machine Settings

Hardware | Options

Device	Summary
Memory	4 GB
Processors	1
Hard Disk (SCSI)	80 GB
CD/DVD (SATA)	Using file autoinst.iso
CD/DVD 2 (SATA)	Using file D:\ISOs\Nix\ubuntu-16.04....
Floppy	Using file autoinst.flp
Network Adapter	Host-only
Display	Auto detect

Device status
- [ ] Connected
- [x] Connect at power on

Connection
- ○ Use physical drive: Auto detect
- ● Use ISO image file: autoinst.iso

- Install Ubuntu Server
  - Make sure to install “standard system utilities” and “OpenSSH server” during the Software selection phase
  - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.1.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy
- After the install is completed, power off the VM and adjust the following settings:
  - Remove the remaining CD/DVD drive
  - **Make sure to document the MAC address under advanced settings for the *Network Adapter***
- Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static

mapping to the LAN interface DHCP, for the MAC address of network adapter for the SIEM VM; assign it the IP address 172.16.1.3, making sure to enter a description that tells you which VM this static mapping belongs to

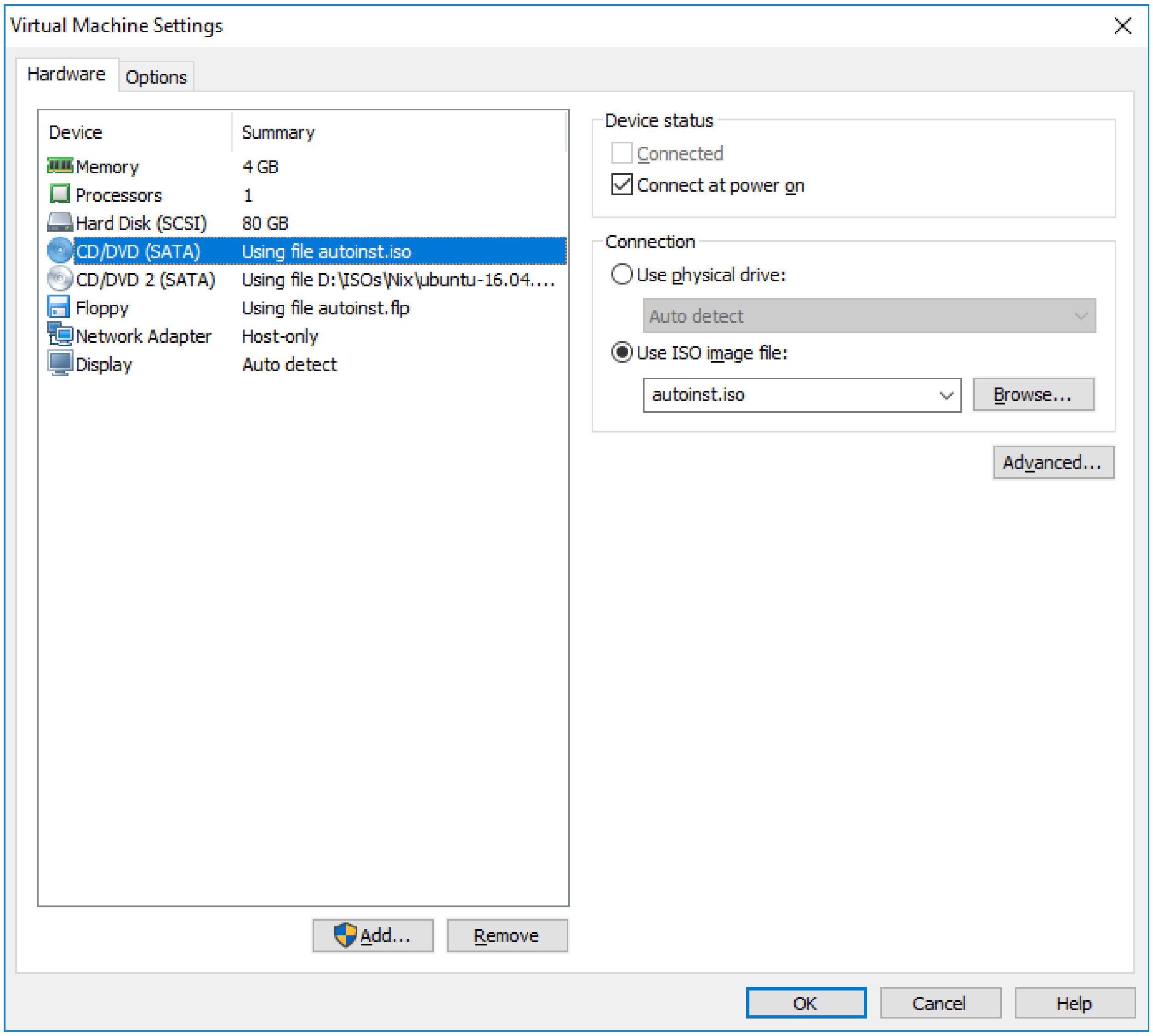
- Power on the virtual machine and do the following:
    - Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense
        - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
    - Verify the virtual machine can reach the internet
        - In a terminal/shell run the command `curl -I https://www.google.com`
            - This will confirm DNS can resolve domain names, and that the VM can reach the internet
            - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense
                - **Note:** The `ping` command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail
    - Update/patch the VM
        - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
            - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
    - Create a snapshot for the VM

IPS VM

This VM is going to be responsible for running the AFPACKET bridge between the IPS 1 (VMnet2) and IPS 2 (VMnet 3) virtual networks. Perform the following tasks to install the IPS VM:

- If you haven't already for the SIEM VM, Download Ubuntu Server 16.04.1 LTS 64-bit here: https://www.ubuntu.com/download/server
- Create a VM with the following settings:
    - Configure the VM to install from the Ubuntu Server ISO you downloaded
    - Ubuntu supports VMware's "Easy Install" option, but we're not going to use it
        - Fill out whatever data you want in the full name, username, password and confirm fields.
    - Name the VM "IPS"
    - Allocate 80 GB of space for the disk
        - Choose the "store virtual disk as a single file" option

- Customize the following hardware settings
    - Allocate 2GB (2048MB) of RAM
    - Allocate 1-2 processors, where total number of cores is no more than 2
    - Connect the VM’s first network adapter to “Host-only” (VMnet1)
        - Ensure that *Connect at power on* is checked
    - Create two additional network adapters.
        - Attach Network Adapter 2 to VMnet2 (IPS 1)
            - Ensure that the *Connect at power on* checkbox is **unchecked**
        - Attach Network Adapter 3 to VMnet3 (IPS 2)
            - Ensure that the *Connect at power on* checkbox is **unchecked**
    - Remove the USB Controller
    - Remove the Sound Card
    - Remove the Printer
- **Before powering on the VM to install Ubuntu Server**
    - Enter the *Virtual Machine Settings* under the *Hardware* tab. Remove the CD/DVD drive with the file named `autoinst.iso`, as well as the Floppy drive with the file named `autoinst.flp`


Virtual Machine Settings

Hardware | Options

Device	Summary
Memory	4 GB
Processors	1
Hard Disk (SCSI)	80 GB
CD/DVD (SATA)	Using file autoinst.iso
CD/DVD 2 (SATA)	Using file D:\ISOs\Nix\ubuntu-16.04....
Floppy	Using file autoinst.flp
Network Adapter	Host-only
Display	Auto detect

Device status

- [ ] Connected
- [x] Connect at power on

Connection

- ○ Use physical drive: Auto detect
- ● Use ISO image file: autoinst.iso

- Install Ubuntu Server
    - Make sure to install “standard system utilities” and “OpenSSH server” during the Software selection phase
    - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.1.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy
- After the install is completed, power off the VM and adjust the following settings:
    - Remove the remaining CD/DVD drive
    - Check the *Connect at power on* checkbox for *Network Adapter 2* and *Network Adapter 3* (connected to *VMnet2* and *VMnet3*)
    - **Make sure to document the MAC address under advanced settings for all**

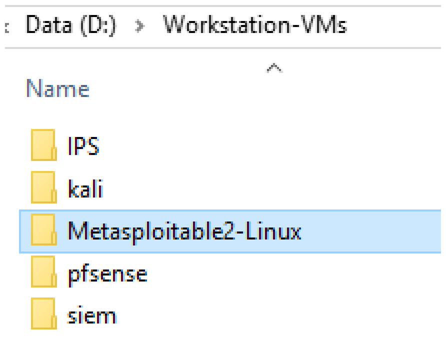
        three network adapters, making sure to note which MAC address pairs up to which network adapter, connected to which VMnet

- Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the LAN interface DHCP, for the MAC address of network adapter connected to VMnet1 (Management network, Host-only VMnet); assign it the IP address 172.16.1.4, making sure to enter a description that tells you which VM this static mapping belongs to
- Power on the virtual machine and do the following:
    - Verify that the static mapping for the VM’s IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter’s name is in the VM) was given the IP address you configured its static mapping for in pfSense
        - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server.
    - Verify the virtual machine can reach the internet.
        - In a terminal/shell run the command `curl -I https://www.google.com`
            - This will confirm DNS can resolve domain names, and that the VM can reach the internet
    - Update/patch the VM
        - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
            - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
    - Create a snapshot for the VM

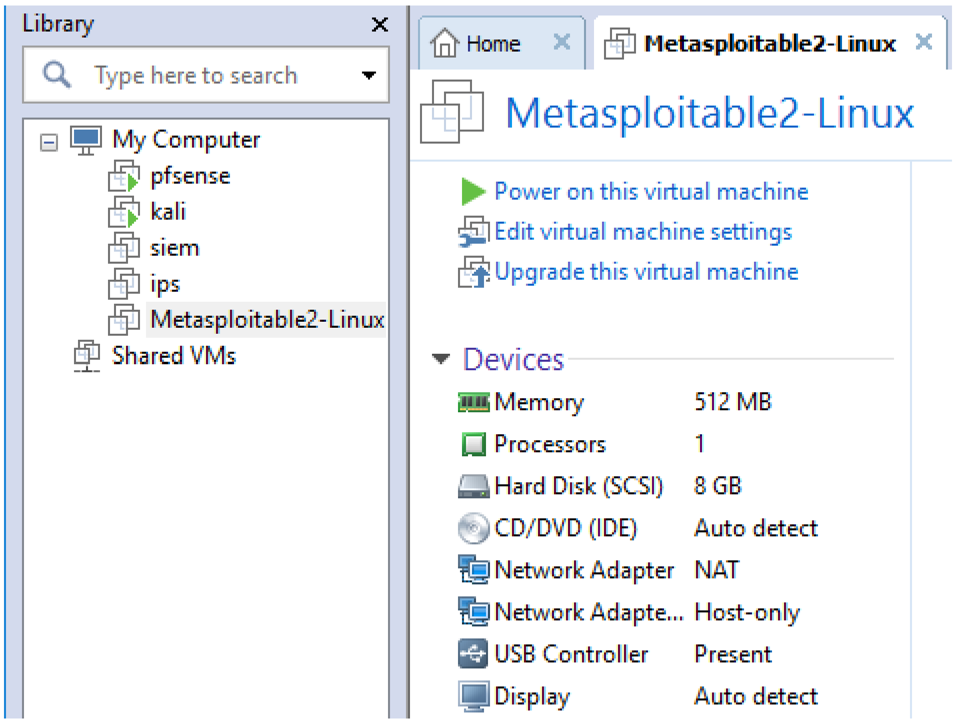
Metasploitable 2

Metasploitable 2 is a little different from the other VMs. It is already pre-created, we just need to import it, and reconfigure some of the virtual hardware. This should be pretty simple to do. First, download a copy of Metasploitable 2 from https://sourceforge.net/projects/metasploitable/files/Metasploitable2/. Metasploitable 2 is distributed as a zip archive, so you will need a compression tool to extract it. On Linux/Unix systems, you can use the command line utility `unzip`. On Windows, I recommend 7-Zip, which can be found at http://www.7-zip.org/download.html.

Download the Metasploitable 2 archive, and decompress it in the download directory. You will be left with a folder labeled `metasploitable-linux-2.0.0`. Inside of that folder is another folder labeled `Metasploitable2-Linux`. Move the entire `Metasploitable2-Linux` directory to the same directory you use to store your other virtual machines. In my case, this is `D:\Workstation-VMs`.



In the VMware Workstation main window, on the menu bar, click *File* > *Open…* and navigate to the folder you just moved the `Metasploitable2-Linux` directory to. For me, that path was `D:\Workstation-Vms\Metasploitable2-Linux`. Select the `Metasploitable.vmx` file in this directory. A new VM named Metasploitable2-Linux should appear in the "Library" pane of VMware Workstation.



Memory	512 MB
Processors	1
Hard Disk (SCSI)	8 GB
CD/DVD (IDE)	Auto detect
Network Adapter	NAT
Network Adapte...	Host-only
USB Controller	Present
Display	Auto detect

There are a few more settings we need to adjust before starting this virtual machine. Using what you know from setting up the previous VMs, perform the following actions in the settings menu, before attempting to boot it:

- Remove the USB Controller
- Remove Network Adapter 2
- Remove the CD/DVD drive
- Reconfigure The Network Adapter from "NAT" to "VMnet3" (IPS 2 Network)
  - Document the MAC address of this network interface
- Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the OPT1 interface DHCP, for the MAC address of network adapter connected to VMnet3 (IPS 1 network); assign it the IP address 172.16.2.3, making sure to enter a description that tells you which VM this static mapping belongs to!

Power on the Metasploitable 2 VM, make sure it is bootable. If so, take a snapshot of the VM, it's ready to go for now. You won't be able to verify whether the DHCP mapping works until AFTER the ips VM is configured.

Next Steps

The VMware Workstation initial setup is all but done at this point. However, you're not quite out of the woods just yet. Here is a checklist of tasks to complete:

- If you haven't completed the section "Defense in Depth for Windows Hosts", and you are using Windows as a hypervisor host, I would very highly suggest doing so.
- While not strictly necessary, you may want to complete "Remote Lab Management" section, for the host OS of your choice (e.g. Windows Remote Access, Linux/Unix Remote Access), Enabling SSH on Kali Linux, and/or if you're lazy like me, the section on Securing root SSH access. This will allow you to remotely manage all of the Linux VMs much more easily than through the VMware Workstation console.
- You have to install IDS/IPS software on the IPS VM. You'll need to complete the "IPS Installation Guide" to learn how to do this with either Snort or Suricata as your IPS software of choice.
- The SIEM VM needs to have Splunk installed and configured. You'll need to complete the "Splunk Installation Guide".
- You may want to consider reading the Automated Patching for Linux Lab VMs chapter, and implementing the `updater.sh` script for your Linux VMs.
- Do you want some ideas on where to take your lab? Check out the chapter "In Your Own Image", for some tips on how to mold your VM lab to better suit your needs.

Setup - VMware vSphere Hypervisor (ESXi)

Note: This guide was made using VMware vSphere Hypervisor 6.0, Update 2 with free licensing, and assumes that users of this guide are this software version or higher.

VMware vSphere Hypervisor (to which from now on I am going to refer to as ESXi, its traditional name) is a bare-metal hypervisor. As opposed to the hosted hypervisors covered in previous sections,. you need dedicated hardware to install ESXi on, and a second workstation or laptop to manage the ESXi server. Configuration, installation and setup are a little bit different too, compared to hosted hypervisors. Read on, pay attention, and you'll come out of this just fine.

Installation

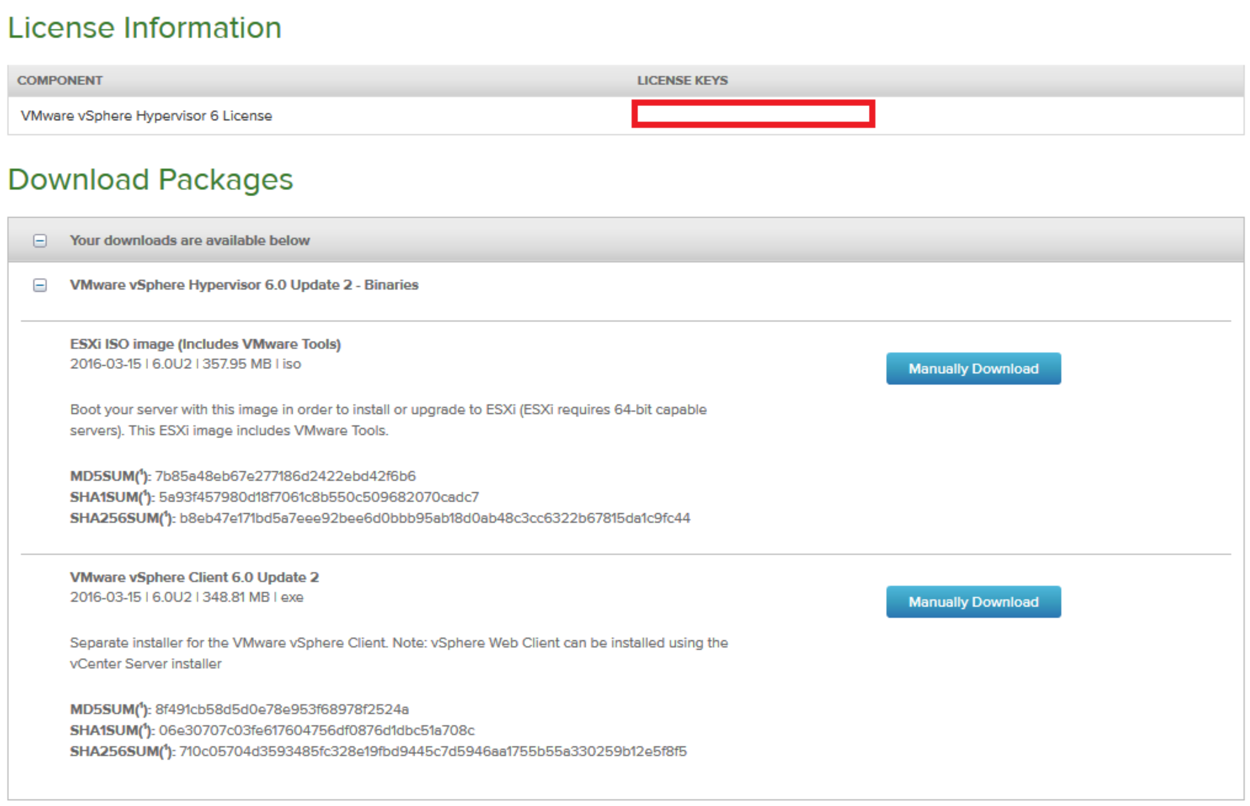
As mentioned in "Hypervisor and Hardware Considerations", ESXi is pretty picky about the hardware it recognizes and installs on. And then, even if it supports your system's components, there may be cases in which not all features of the motherboard/chipset are fully supported. This isn't much of a problem if you are using commercial server solutions from a known hardware vendor (e.g. Dell or HP servers), but if you're trying to build your own system (also known as a *white box* server) you may face some problems. For instance, I built a server based on a Supermicro motherboard that was supported by ESXi, but the Intel RAID controller wasn't. The server would recognize hard drives, but not RAID arrays. Others have told me about experiences where their motherboard was supported, but the integrated ethernet cards were not, so they were required to buy PCI/PCI-e network cards in order to run ESXi.

Some people have managed to install ESXi on some pretty exotic hardware - Shuttle PCs, Intel NUCs, among other things. A lot of these builds require recreating the installation ISO to ensure that drivers for specific chipsets/hardware components are included in the image file. Unfortunately, this is beyond the scope of this guide. I can't guarantee your success if you try to install ESXi on exotic hardware, but if you want to build your own server (instead of buying a new or used commercial system), here are some things to consider:

- ESXi works really well with Intel chipsets and network cards. There's a good chance that if you have an Intel chipset on your motherboard and/or Intel network cards (integrated or PCI cards), that you will do just fine.
- While most prefer Intel chipsets and network cards where possible, the next best solution is to utilize Broadcom network cards. A lot of sysadmins I know dislike Broadcom due to their loose interpretations of various standards, but support for Broadcom NICs in ESXi is pretty decent.
- VMware has a hardware compatibility list you can search here: https://www.VMware.com/resources/compatibility/search.php. Generally, the compatibility list is very cumbersome to use. A better solution would be to Google search for "part/model number" "esxi" to see if there are any results for compatibility problems for the server model or the server components you want to use.

So now that we got hardware compatibility out of the way, you need to actually download ESXi and install it. In order to do so, you need to create an account on VMware's website. After you have signed up and logged in, navigate to Downloads and find "VMware vSphere Hypervisor(ESXi)". VMware changes links around all the time, but for the time being, the page for ESXi can be found here: https://my.VMware.com/en/web/VMware/evalcenter?p=free-esxi6. On the download page, you will be presented with a free product key for the free version of ESXi. While there are some significant differences between the free and the enterprise versions of ESXi, for our purposes, the free version and licensing will work just fine. Record your license key, since we will need it later.

Make sure to download the latest version of ESXi (which as of this writing is 6.5), the latest version of VMware vSphere Client (if you are using Windows as your hypervisor manager, and don't want to use ESXi's web-based UI). vSphere client can be found at https://kb.vmware.com/kb/2089791



License Information

COMPONENT	LICENSE KEYS
VMware vSphere Hypervisor 6 License	

Download Packages

Your downloads are available below

VMware vSphere Hypervisor 6.0 Update 2 - Binaries

ESXi ISO image (Includes VMware Tools)
2016-03-15 | 6.0U2 | 357.95 MB | iso

Boot your server with this image in order to install or upgrade to ESXi (ESXi requires 64-bit capable servers). This ESXi image includes VMware Tools.

MD5SUM(¹): 7b85a48eb67e277186d2422ebd42f6b6
SHA1SUM(¹): 5a93f457980d18f7061c8b550c509682070cadc7
SHA256SUM(¹): b8eb47e171bd5a7eee92bee6d0bbb95ab18d0ab48c3cc6322b67815da1c9fc44

VMware vSphere Client 6.0 Update 2
2016-03-15 | 6.0U2 | 348.81 MB | exe

Separate installer for the VMware vSphere Client. Note: vSphere Web Client can be installed using the vCenter Server installer

MD5SUM(¹): 8f491cb58d5d0e78e953f68978f2524a
SHA1SUM(¹): 06e30707c03fe617604756df0876d1dbc51a708c
SHA256SUM(¹): 710c05704d3593485fc328e19fbd9445c7d5946aa1755b55a330259b12e5f8f5

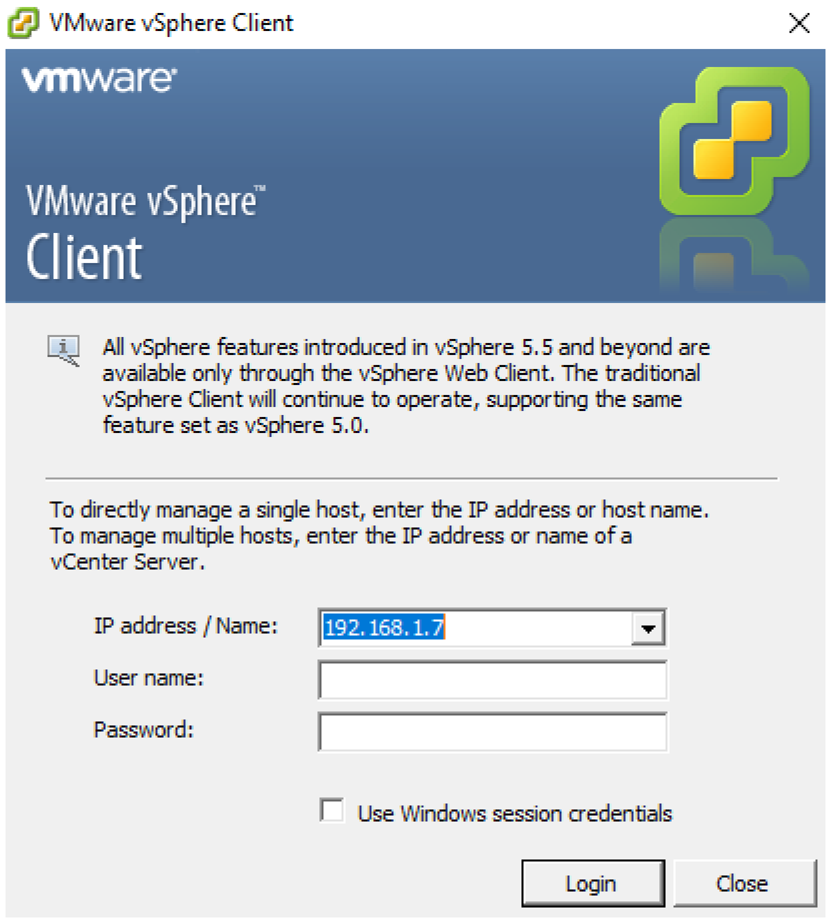
So now that you have an ISO, you need to figure out how you're going to install ESXi on your server. Traditionally, you would use a CD burning utility to get the ISO onto a CD or DVD, and boot from the disk. However, many PCs and servers are lacking optical drives these days The alternative is to install ESXi from a USB thumb drive or other external media. I found a great guide on how to create a USB installer for ESXi here: http://www.virten.net/2014/12/howto-create-a-bootable-esxi-installer-usb-flash-drive/

With a bootable CD/DVD or USB drive prepared, there are multiple different options where ESXi can be installed to. IT professionals have been found to install ESXi on USB drives or even SD cards. The reason for this is that ESXi itself only requires around 4 GB of space to install properly, with most of the components of a running ESXi residing in memory, so a flash drive or SD card more than serves the purpose of booting the system. Installing on external media saves space on the drives you'll be using to store virtual machines, and also ensures that running VMs won't be contending for disk I/O with ESXi itself. If you don't have a spare SD card or USB drive laying around, don't worry about this too much, just bear in mind that this is an installation option available, and that a lot of sysadmins do this. Also remember that **if you do decide to install ESXi to an SD card or USB drive, system logging will be disabled by default!** To enable logging for this installation/boot option anyway, you can adjust the advanced host settings of the ESXi to use one of the datastore disks available (where VMs are stored) to write its logs on, or specify the IP address of a syslog server where the logs should be sent to.

The installation itself is pretty straightforward. During the process, you will be asked to set the root password for the system, create datastores on attached hard drives (for actually storing your virtual machines and their files), as well as configure the management interface (e.g. the network interface and IP address you will be using to connect to this server). If at all possible, you should consider setting a static IP address for this host, or creating a static DHCP mapping to ensure that your ESXi server always has the same IP address in the event the system reboots. Spending a portion of your day to figure out that DHCP has granted your ESXi server a new IP address is pretty annoying.

Accessing ESXi

Once you have ESXi installed, and can confirm that it is reachable over the network (e.g. by using `ping`), you should be able to manage the system remotely. If you're using a Windows host, you have the option of utilizing the VMware vSphere Client to access your ESXi server. Simply plug in the IP address of your ESXi host, enter a username and password (in this case, you want to log in as the root user you created during installation), and click the Login button.



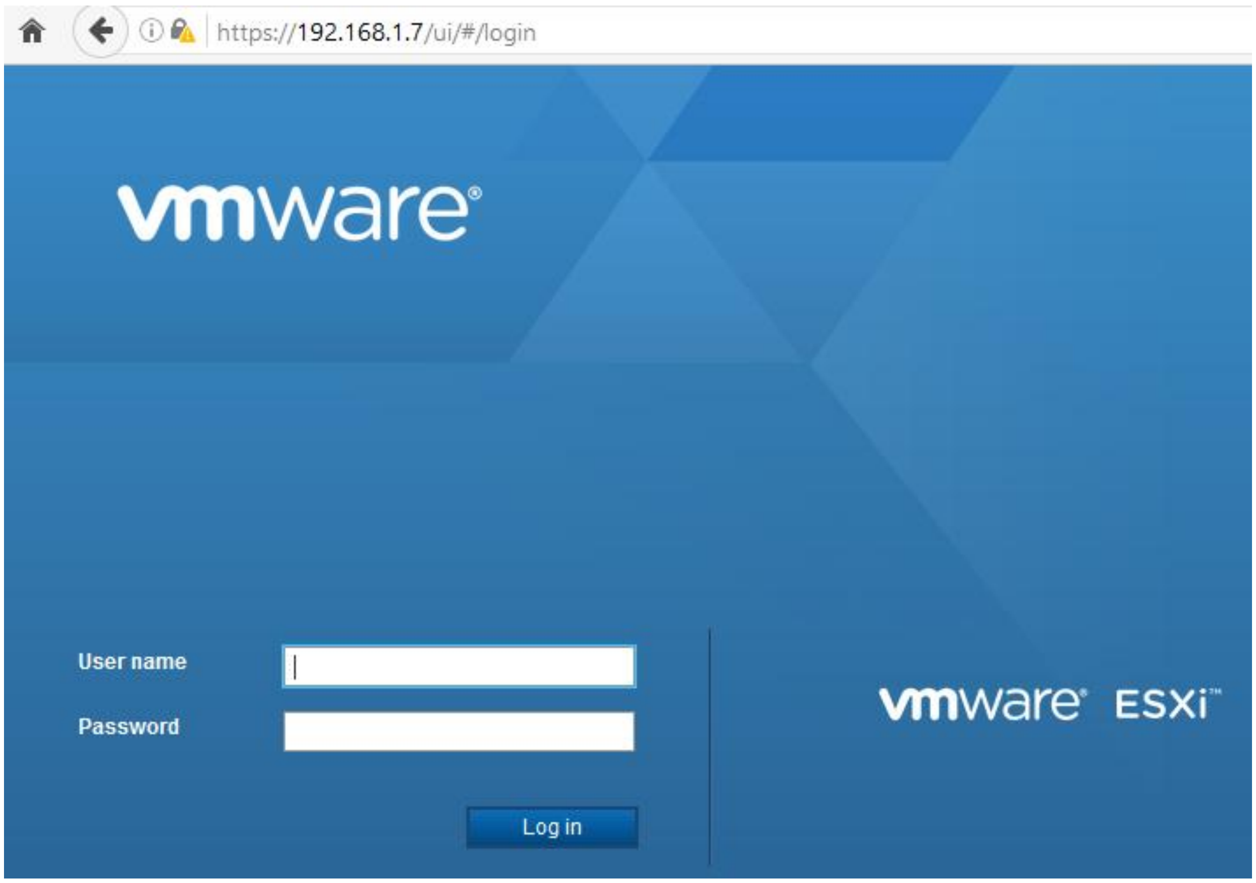
All vSphere features introduced in vSphere 5.5 and beyond are available only through the vSphere Web Client. The traditional vSphere Client will continue to operate, supporting the same feature set as vSphere 5.0.

To directly manage a single host, enter the IP address or host name. To manage multiple hosts, enter the IP address or name of a vCenter Server.

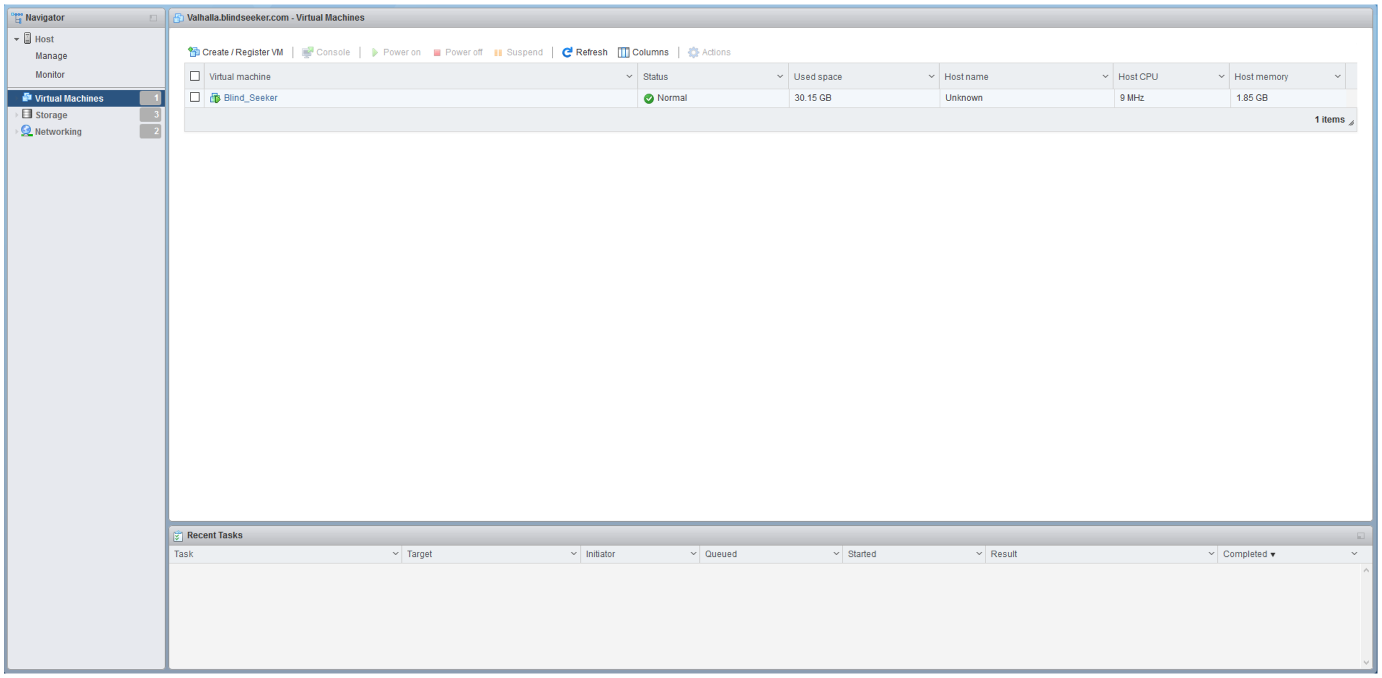
IP address / Name: 192.168.1.7
User name:
Password:

Use Windows session credentials

If you are not using a Windows OS, or you would rather not install/use the vSphere Client, you can point your web browser to https://[server ip]/ui to log in to the VMware ESXi via its HTML5-based web interface. The web UI can be used on Windows, OS X, or Linux, provided you are using a modern web browser that is able to handle HTML5 content. For the sake of simplicity, **we will be using the HTML5 web interface to interact with the ESXi server and perform the configuration tasks, except where noted to do otherwise and/or if workarounds are required.** Please keep in mind that the HTML5 web interface is still relatively new, and prone to bugs here and there.



When you have successfully logged in to the server, you should be greeting with a dashboard that, among other things, shows you the list of VMs available on your ESXi instance. In your case, you shouldn't have any installed yet. In my case, since I'm literally running a web server out of my basement, a virtual machine named BlindSeeker is in the inventory list for this ESXi server.



Virtual machine	Status	Used space	Host name	Host CPU	Host memory
Blind_Seeker	Normal	30.15 GB	Unknown	9 MHz	1.85 GB

1 items

Task	Target	Initiator	Queued	Started	Result	Completed ▾

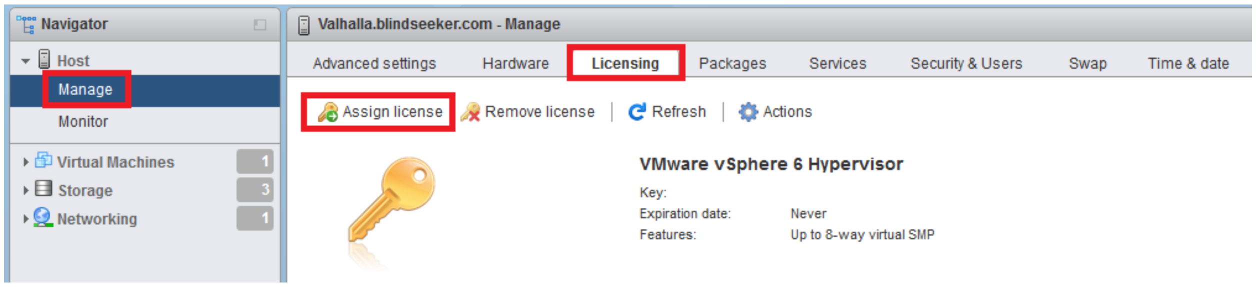
Get used to this interface. Know it and love it, since we'll be spending a lot of time here. The window pane on the left, entitled *Navigator*, is going to be your best friend. As the name implies, it is used to navigate the various settings available for managing the ESXi server. From checking resource utilization, to creating VMs, to enabling/disabling features on the ESXi host, the *Navigator* pane is the key to getting around.

Hypervisor Setup

ESXi is a little bit different from our hosted hypervisors. We need to perform some initial configuration and a couple of post-installation tasks in order to make it ready for us to get started.

Licensing

On the Navigator pane, under *Host*, there is an option called *Manage*. When you click on it, a series of tabs is displayed on top of the middle pane, each related to different aspects of managing the ESXi host. Select the *License* tab, then click *Assign License*. Input the key we got from the VMware website earlier. By default, ESXi will start up in what is called evaluation mode, which offers you the full featureset available for the system (which equals a vSphere Enterprise Plus license) for 60 days. As you need to enter a valid license key at this point at the latest to preserve functionality, it is best to get that taken care of up front, rather then letting it become a problem later. If your key is valid and has been entered successfully, your Licensing page should look like this. Please note that I intentionally blanked out my license key on this illustration.

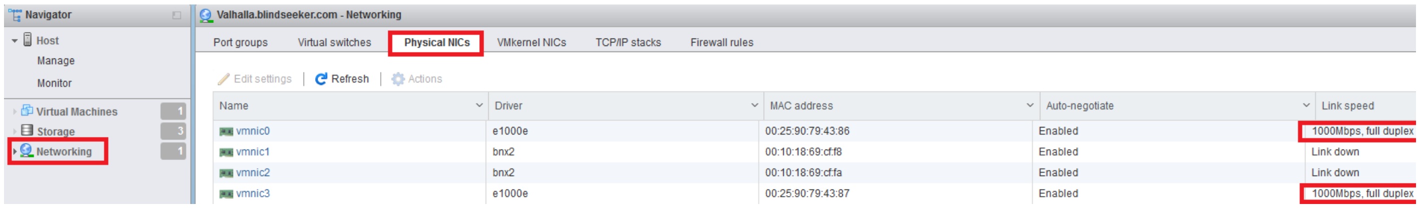


VMware vSphere 6 Hypervisor
Key:
Expiration date: Never
Features: Up to 8-way virtual SMP

Networking and Virtual Switches

ESXi's virtual networking is based off of creating virtual switches, then setting up port groups on these vSwitches, with a VM's virtual network adapter being assigned to a port group. There are no NAT networks, host-only networks, or internal networks here; just virtual switches and port groups. To provide access to a physical network, a virtual switch can be uplinked to one or multiple physical network port(s) on the ESXi host machine to serve the same purpose as a bridged network. Without an uplink, a stand-alone vSwitch operates very similarly to a private or host-only network. To do networking on our ESXi server the *right* way, we need at least two ports available on the server itself that are connected to the physical network. One of the network cards/ports is reserved by the ESXi host's VMkernel port (linked to a device named *vmk0*) for managing the ESXi server itself, while the other network card will be dedicated to network traffic to and from our VMs.

To verify that your network cards/ports have been recognized by ESXi properly, under Navigator in the systems web UI, click *Networking*, then select the *Physical NICs* tab. ESXi will display the status of each recognized network card available to the system, including whether or not the cable is plugged in and/or what link speed the network card has negotiated to. If your network card(s) are NOT showing up here, ESXi does NOT have drivers for them or they are not supported. In my case, I have four network ports (vmnic0-vmnic3), two of which are using the e1000e driver (Intel), and two of which using the bnx (broadcom) driver. Additionally you can see that two of the network ports are physically connected and have a link speed of 1gbps each.

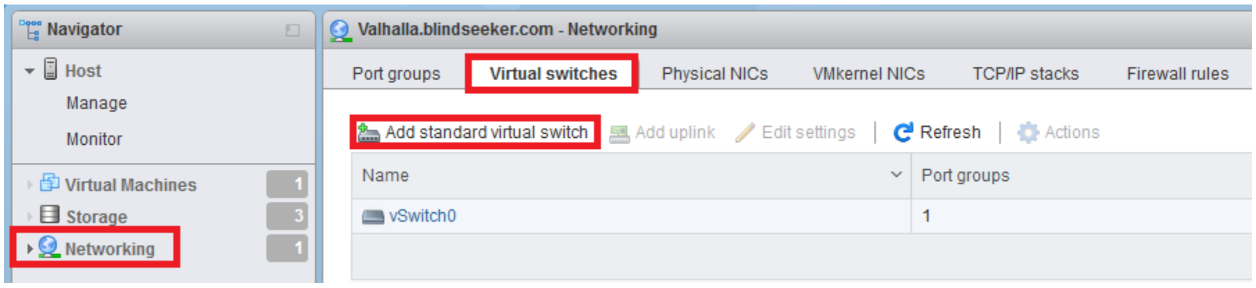


Name	Driver	MAC address	Auto-negotiate	Link speed
vmnic0	e1000e	00:25:90:79:43:86	Enabled	1000Mbps, full duplex
vmnic1	bnx2	00:10:18:69:cf:f8	Enabled	Link down
vmnic2	bnx2	00:10:18:69:cf:fa	Enabled	Link down
vmnic3	e1000e	00:25:90:79:43:87	Enabled	1000Mbps, full duplex

Don’t have two network cards? Don’t fret. There are ways to force the VMkernel port and the vNICs of the virtual machines to use the same network card/port. Though I must state that this goes against best practice, is not technically supported, and may not be an option in future versions of VMware ESXi. I’ll show you how to do it anyhow, because this is a lab environment and you shouldn’t be hosting anything of importance here anyway, right? Right. Now, let’s create some virtual switches.

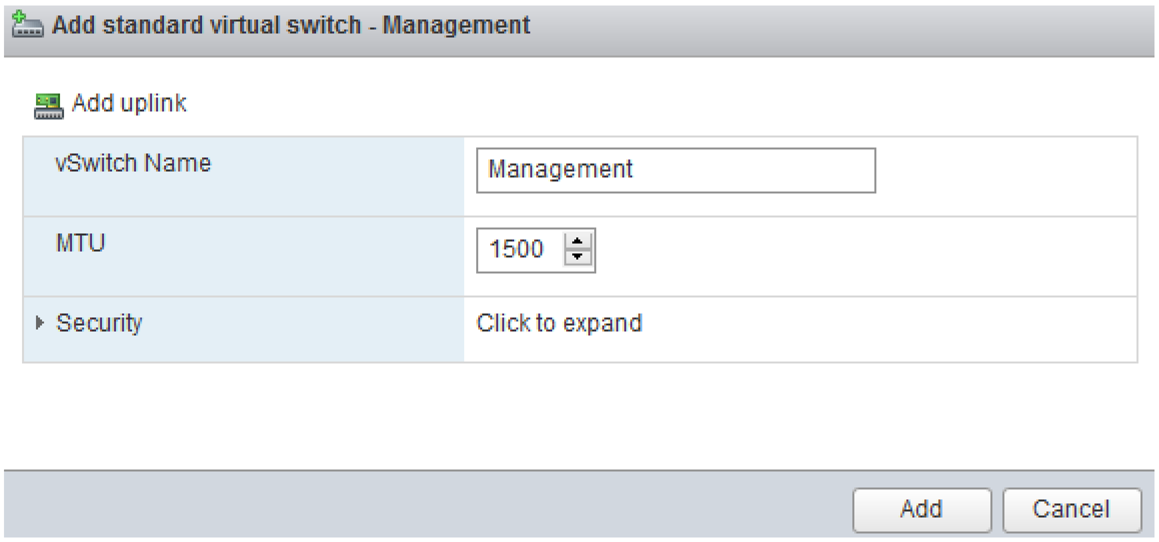
Creating Virtual Switches

Under the *Navigator* pane, click on *Networking*, then click on the *Virtual switches* tab. You’ll notice that a device named *vSwitch0* is already present. This is the virtual switch that ESXi creates by default to attach the VMkernel port to the management interface, and we won’t be using it. If you were to click *Edit Settings* for this vSwitch, you would see that this vSwitch is uplinked to the network card you chose as the management interface when you configured networking during the initial ESXi installation. For now, click the *Add standard virtual switch* option.



Name	Port groups
vSwitch0	1

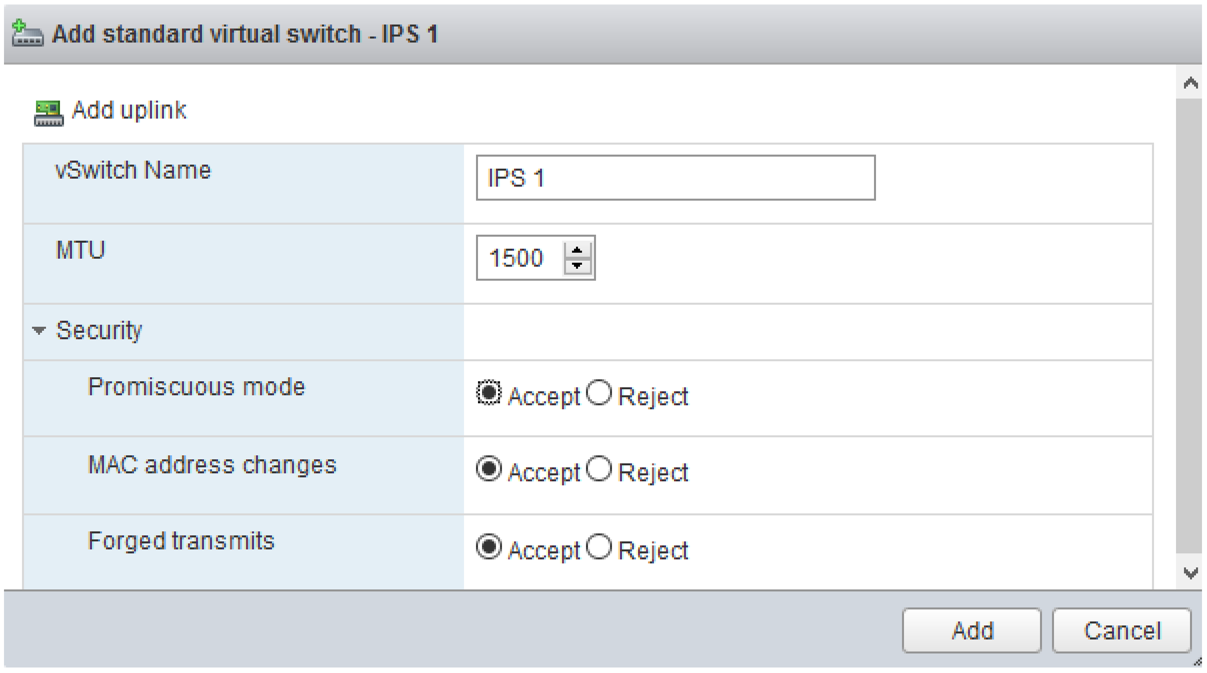
Name this vSwitch “Management”. Click the *X* on the right side of the *Uplink 1* field to remove the drop-down. We do NOT want this virtual switch to have an uplink. Click the Add button to create this vSwitch.


Add standard virtual switch - Management

Add uplink

vSwitch Name	Management
MTU	1500
Security	Click to expand

Next, click the *Add standard virtual switch* option again. This time, name the virtual switch "IPS 1", and as you did with the *Management* virtual switch, remove the *Uplink 1* field by clicking the *X* next to the drop-down. Click the triangle next to *Security* to expand the corresponding settings for the virtual switch. Click the *Accept* radio button for all three settings - *Promiscuous mode*, *MAC address changes*, and *Forged transmits*. To finish, click the *Add* button.


Add standard virtual switch - IPS 1

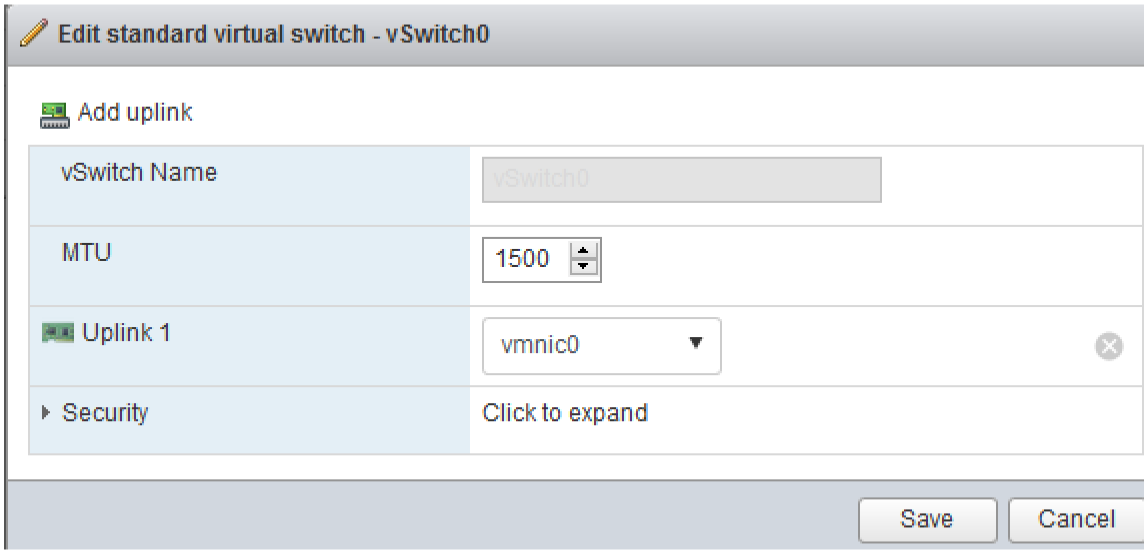
Add uplink

vSwitch Name	IPS 1
MTU	1500
Security	
Promiscuous mode	[✓ Accept] Reject
MAC address changes	[✓ Accept] Reject
Forged transmits	[✓ Accept] Reject

Add another virtual switch with the exact same settings (all of the Security radio buttons set to *Accept*), and name it "IPS 2".

We need to create one more virtual switch. This one is going to serve as the Bridged Network, hence we need to connect it to one of the host's physical network cards/ports. Since one network port is already uplinked to vSwitch0, you need to ensure that you have at least one additional network connection available and physically connected. If this isn't feasible or just not an option for you, please visit the section "Adding a Virtual Machine Port Group to vSwitch0 - No Secondary NIC Available" for further instructions.

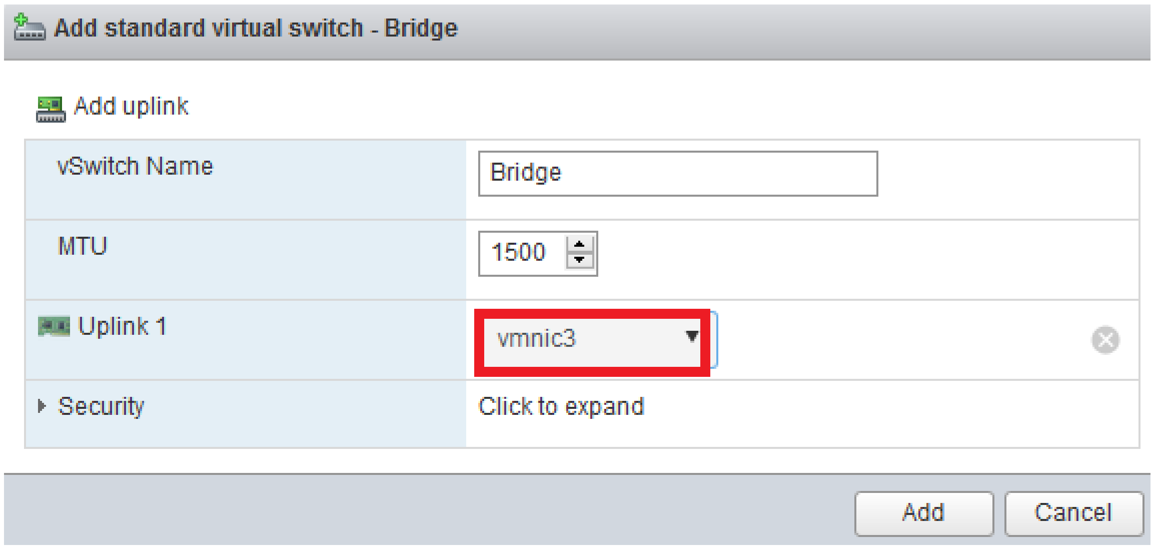
Ahead of creating the vSwitch for our Bridged Network, we need to confirm which network card vSwitch0 is uplinked to. To do so, select the *Virtual switches* tab, then click on *vSwitch0*.. On top of the overview page for this virtual switch, click on *Edit settings*. Take notice of the network interface listed in the *Uplink 1* field. Click cancel to exit.


Edit standard virtual switch - vSwitch0

Add uplink

vSwitch Name	vSwitch0
MTU	1500
Uplink 1	vmnic0
Security	Click to expand

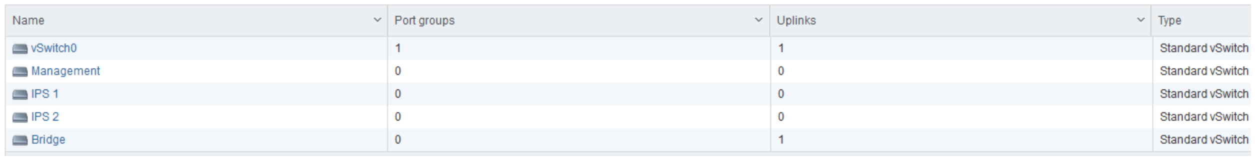
Remember how I had you review which of your server's physical NICs actually were connected under the *Physical NICs* tab? Your listing of physical NICs might look somewhat different, but in my case, vmnic0 and vmnic3 were connected to my physical network. Thanks to the illustration above, we know that vmnic0 is being used as uplink on vSwitch0. Navigate back to *Networking* > *Virtual switches* and click *Add standard virtual switch*. Name this virtual switch "Bridge". This time, we're going to keep "Uplink 1", setting it to a vmnic interface that is physically connected to your physical network, but different from the one used on vSwitch0. In my case, I am using vmnic3. When finished, click the Add button.


Add standard virtual switch - Bridge

Add uplink

vSwitch Name	Bridge
MTU	1500
Uplink 1	vmnic3
Security	Click to expand

Back in the *Virtual switches* tab, you should have a list of vSwitches that looks something like this:



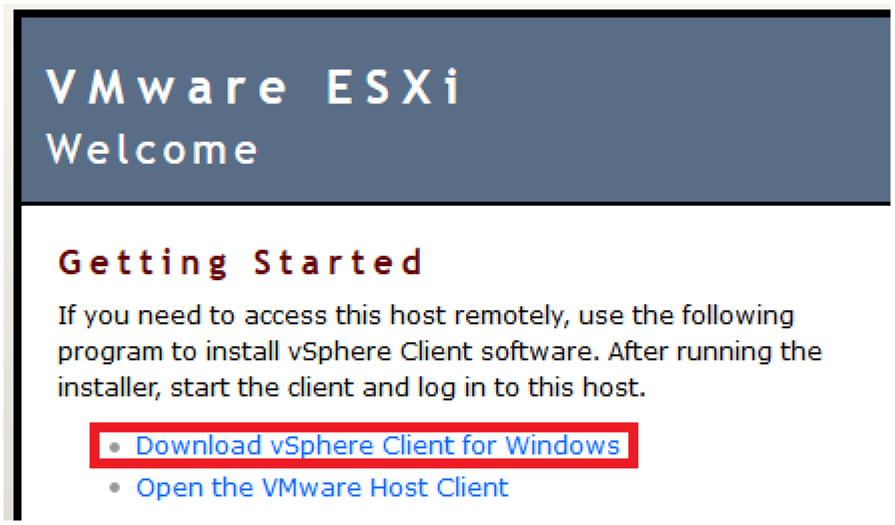
Name	Port groups	Uplinks	Type
vSwitch0	1	1	Standard vSwitch
Management	0	0	Standard vSwitch
IPS 1	0	0	Standard vSwitch
IPS 2	0	0	Standard vSwitch
Bridge	0	1	Standard vSwitch

If you do not have a secondary NIC available, please check out the section "Using the Windows vSphere Client to work around ESXi Web Interface Bugs" to learn how you can work around this.

Port Groups

Port groups are required to allow a VM's network interface(s) to be attached to a virtual switch. Port groups also allow you to logically separate groups of virtual machines connected to the same virtual switch using VLANs. For example, you could have a port group assigned to an "engineering" VLAN, and another port group assigned to a "Human Resources VLAN". They would reside on the same ESXi server, they would use the same vSwitch, but they would be on different VLANs and are thus segmented from each other. There's more to segmentation than that, but that is outside the scope of this document.

Depending on whether or not ESXi's web interface is giving you problems, (See "Resolving Some Interface Bugs") you may need a Windows system with the VMware vSphere client installed to configure port groups and virtual networking.. There is currently a bug where the "Port groups" tab under Networking will not display properly during initial setup, not without at least one virtual machine port group defined. If you did not download vSphere Client from the VMware website, you can get it via the ESXi server by opening up your web browser and pointing it to the management IP address of the host. For example, if your ESXi server's IP address is 192.168.1.20, enter https://192.168.1.20 into your web browser, then click on the *Download vSphere Client for Windows* link to download the software from VMware's site.



VMware ESXi
Welcome

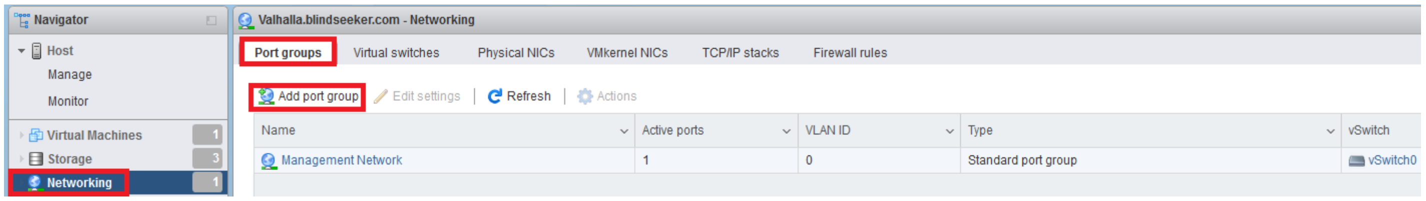
Getting Started

If you need to access this host remotely, use the following program to install vSphere Client software. After running the installer, start the client and log in to this host.

- Download vSphere Client for Windows
- Open the VMware Host Client

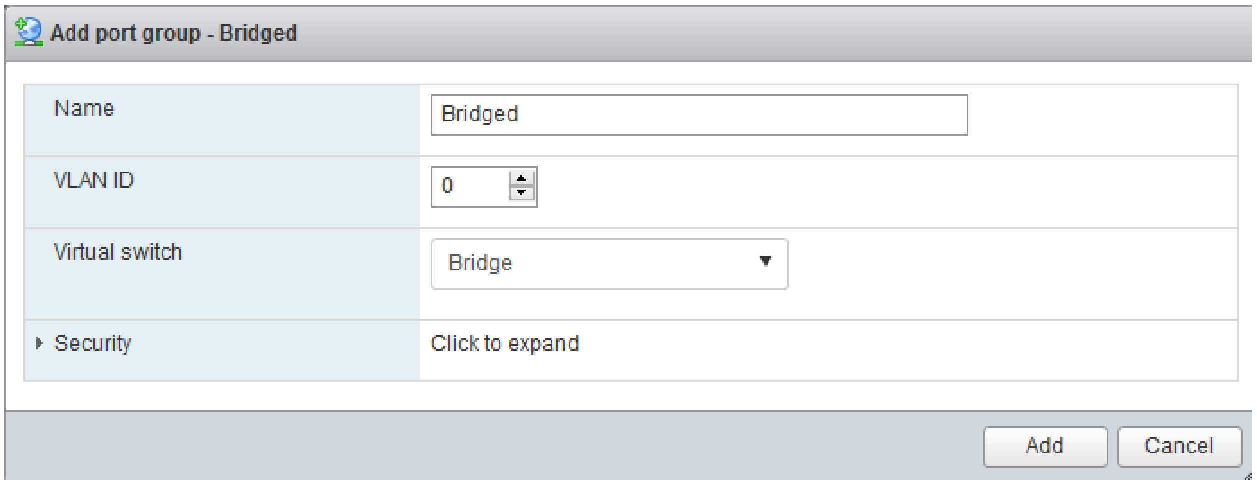
Adding Port Groups via the ESX Web Interface

Under the Navigator pane on the web interface, click on "Networking", select the "Port groups" tab, and click "Add port group".



Name	Active ports	VLAN ID	Type	vSwitch
Management Network	1	0	Standard port group	vSwitch0

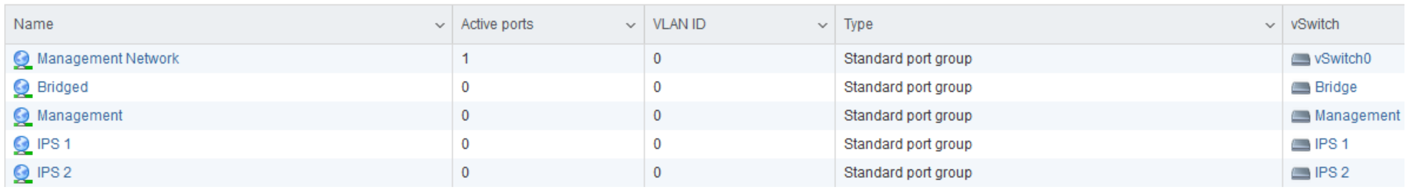
On the window that pops up, name the port group "Bridged" and in the "Virtual switch" drop-down, select "Bridge".When you are done, click the Add button.



Add port group - Bridged

Name	Bridged
VLAN ID	0
Virtual switch	Bridge
Security	Click to expand

You should be back at the “Port groups” tab. Repeat this process three more times, creating a port group for the Management, IPS 1 and IPS 2 vSwitches. Your “Port groups” list should look like this when completed:



Name	Active ports	VLAN ID	Type	vSwitch
Management Network	1	0	Standard port group	vSwitch0
Bridged	0	0	Standard port group	Bridge
Management	0	0	Standard port group	Management
IPS 1	0	0	Standard port group	IPS 1
IPS 2	0	0	Standard port group	IPS 2

Note: If you only have a single network port available to ESXi, you can use vSwitch0 as your switch for the “Bridged” port group. You may need to install the ESX

Resolving Some Web Interface Bugs

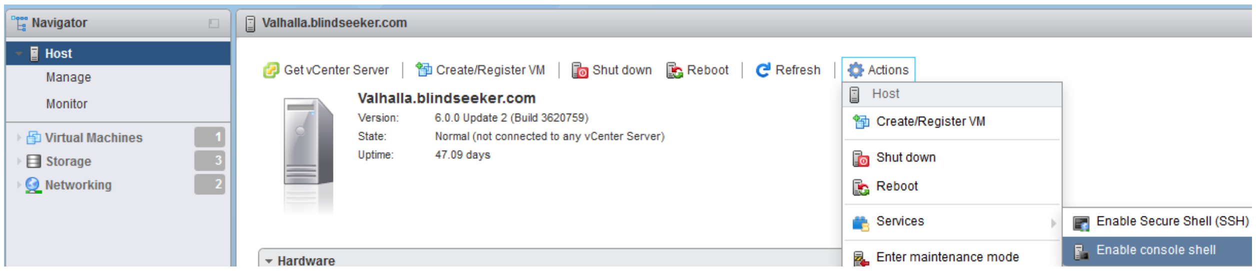
Note: As of ESXi 6.0 Update 3, it appears some of the web interface issues I described below has been resolved. But to the benefit of readers everywhere, I’m including this section here on work-arounds to potential web UI issues, in case you run into new, or similar bugs of your own. **If you’re running ESXi 6.0 update 3 or above, you should be able to skip this section.**

While making this guide, I ran into some web interface bugs related to configuring virtual switches and port groups for some reason, I did not have the option to create port groups This issue appears to affect ESXi 6.0 up to and including ESXi 6.0 Update 2. Fortunately, there are workarounds available, especially if your management workstation runs the Windows OS and you have the vSphere Client installed. For the sake of keeping your options open, I’ll document workarounds to web interface issues I had to use initially, but I’m also going to show you how to update the ESXi embedded web interface using VMware Labs’ Flings to fix the problem. Special thanks to @JohnPittman for pointing me towards this solution.

VMware Flings

Flings are experimental applications and tools made available for system administrators and virtualization enthusiasts to try out before they get accepted as official offerings or mainline additions to VMware products. If you try to download anything on Flings, they make you click a checkbox that says you agree to not install these things on production systems or hold VMware accountable of something terrible happens. In a nutshell, Flings are the equivalent of doing public beta testing of new features and applications. I'm going to walk you through installing the latest "ESXi Embedded Host Client" Fling.

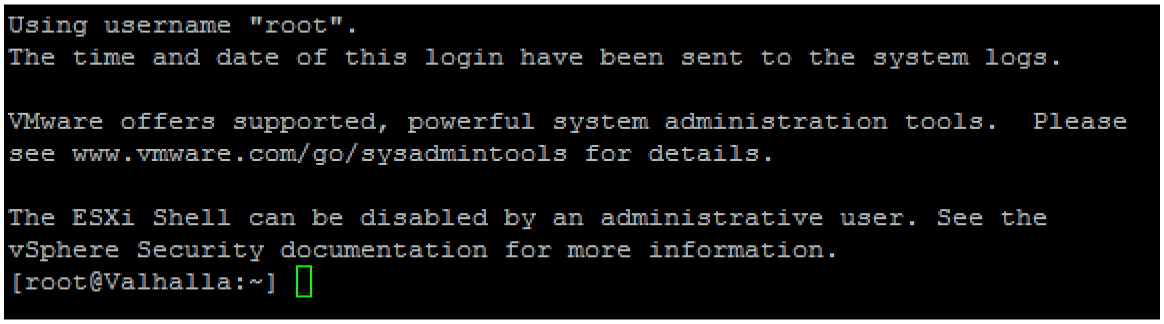
First and foremost, you need to enable access to the ESXi shell. By default, VMware disables access to the command line shell because more often than not, the average user doesn't really have a compelling reason to be mucking around in there, unless they're working with VMware support, or doing strange things to the OS (kinda like what we're about to do). So from a security standpoint, it makes sense. No need to worry however, enabling this access is trivial. In the web interface, click on *Host* under *Navigator*. In the middle pane, click on *Actions*. On the drop-down menu, select *Services*, then click on *Enable console shell*.



Valhalla.blindseeker.com
Version: 6.0.0 Update 2 (Build 3620759)
State: Normal (not connected to any vCenter Server)
Uptime: 47.09 days

This enables access to the shell interface. If you planned on connecting a mouse and keyboard to the machine running the ESXi software, or were going to use a lights-out management system (LOM) or IPMI viewer application of some sort, this is the only service you need to enable.

Apart from that, you may have noticed the menu item *Enable Secure Shell (SSH)* under *Services*. In addition to enabling the console, repeat the steps laid out above once more, only this time this option to start the SSH service on your ESXi host on port 22. From here on, you can use your favorite ssh client, along with the root username and password, to connect to your ESXi server over SSH. If your SSH login is successful, and console access is enabled, you should be greeted with this prompt:

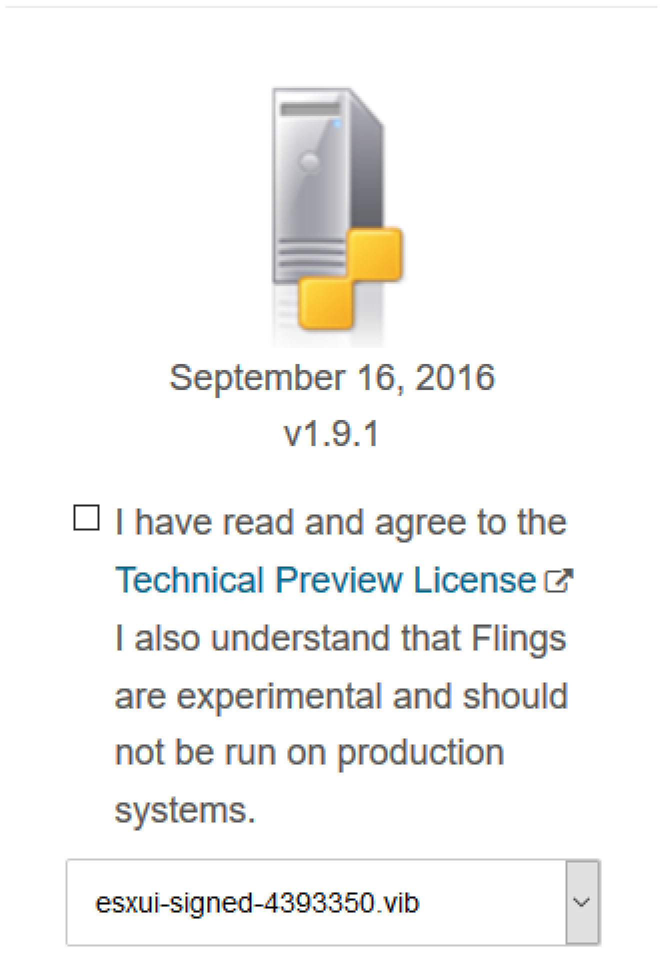


```
Using username "root".
The time and date of this login have been sent to the system logs.

VMware offers supported, powerful system administration tools.  Please
see www.vmware.com/go/sysadmintools for details.

The ESXi Shell can be disabled by an administrative user. See the
vSphere Security documentation for more information.
[root@Valhalla:~]
```

Otherwise, if you prefer using keyboard and mouse, LOM, or IPMI, hit alt+f1 in the ESXi console main menu to gain access to the shell login prompt, and hit alt+f2 to return to the main screen. If console access is enabled, you'll be greeted with a login prompt:

```
Valhalla login: _
```

With CLI access enabled on the ESXi host, go to https://labs.VMware.com/flings/esxi-embedded-host-client and verify the current version and name of the file we will be installing. In my case, the filename is `esxui-signed-4393350.vib`, its version being 1.9.1.

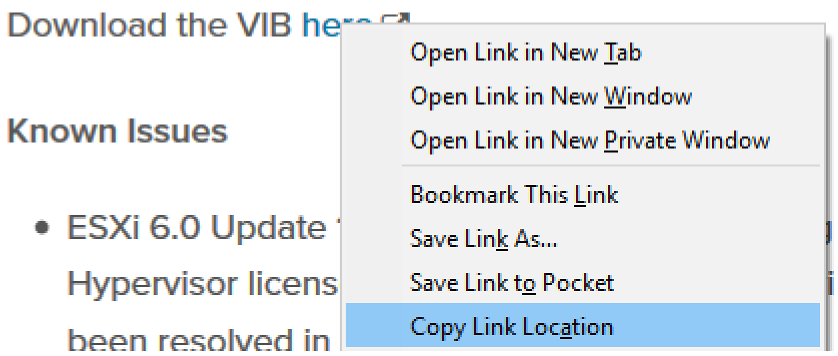

September 16, 2016

v1.9.1

☐ I have read and agree to the Technical Preview License
I also understand that Flings are experimental and should not be run on production systems.

esxui-signed-4393350.vib

Now that you have confirmed that you are going to use the latest version of the esxui package, be sure to copy the download link listed on the page. For instance, http://download3.VMware.com/software/vmw-tools/esxui/esxui-signed-4393350.vib. Copy it to your clipboard or to the text editor of your choice, then return to your ESXi console or SSH session.

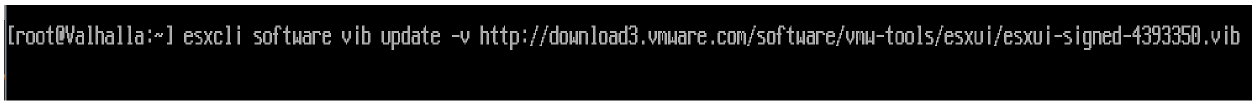


In the CLI, we need to run the following command:

```
esxcli software vib install -v [url here]
```

The complete command I entered looked like this:

```
esxcli software vib update -v http://download3.VMware.com/software/vmw-tools/esxui/esxui-signed-4393350.vib
```

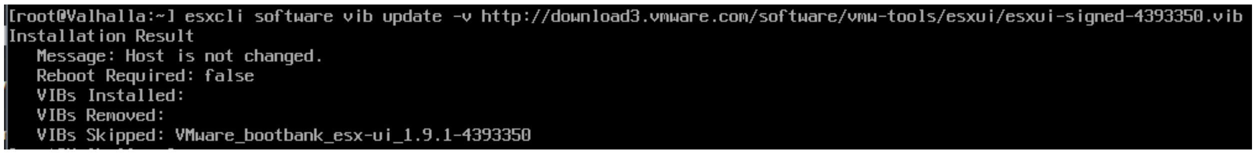


```
[root@Valhalla:~] esxcli software vib update -v http://download3.vmware.com/software/vmw-tools/esxui/esxui-signed-4393350.vib
```

If the command completes successfully, you get a pair of single quotes (‘’) and the prompt returns to you. If you get any errors, following some basic troubleshooting procedures might prove helpful, e.g. making sure that the ESXi server has a DNS server and a default gateway configured, it is allowed to use HTTP outbound, in case of a proxy installed on the network that the ESXi host is configured to use it, etc.

If the system cannot connect to the internet for known or unknown reasons, you need to use the offline installation package and/or employ some method of copying the file to the server. Assuming that you enabled SSH, using SCP would be the easiest way to get the file onto the server. If you need further guidance on installing the esxui package, go here: https://labs.VMware.com/flings/esxi-embedded-host-client#instructions.

If you get an error message stating that VIBs have been skipped, this means that the package has already been installed.

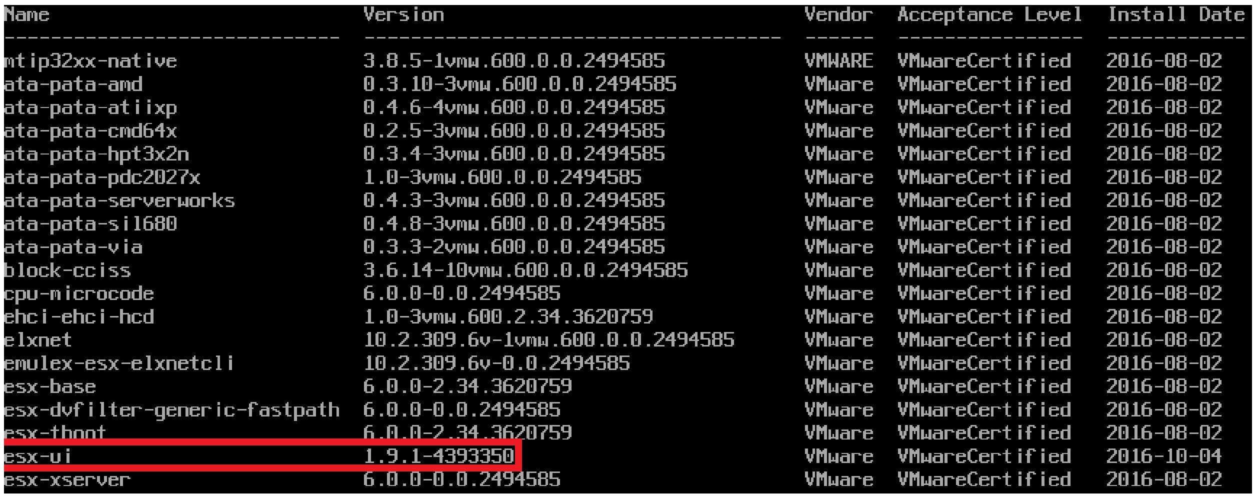


```
[root@Valhalla:~] esxcli software vib update -v http://download3.vmware.com/software/vmw-tools/esxui/esxui-signed-4393350.vib
Installation Result
   Message: Host is not changed.
   Reboot Required: false
   VIBs Installed:
   VIBs Removed:
   VIBs Skipped: VMware_bootbank_esx-ui_1.9.1-4393350
```

You can confirm that the package has been successfully installed by running the following command:

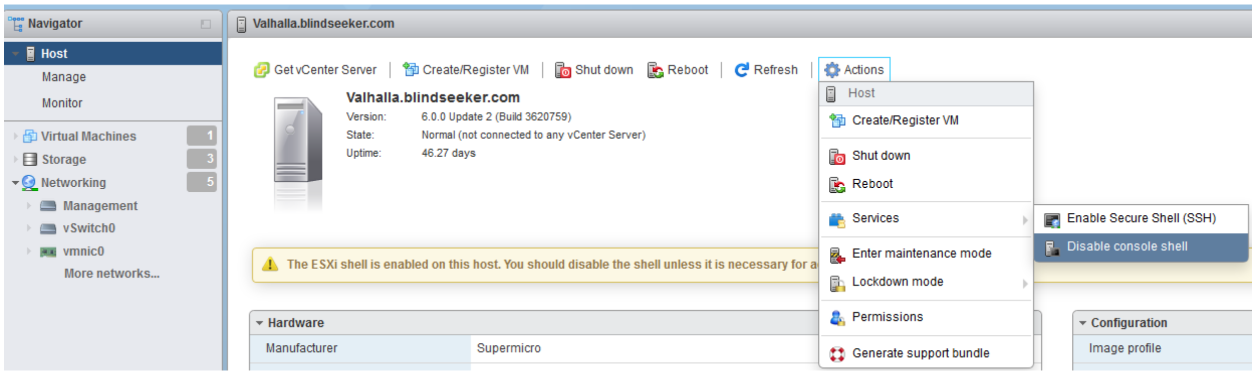
```
esxcli software vib list | less
```

Look for `esx-ui` in the Name column, and verify that the version number displayed matches the one of the file you previously downloaded from the VMware labs site.



```
Name                           Version                              Vendor  Acceptance Level  Install Date
-----------------------------  -----------------------------------  ------  ----------------  ------------
mtip32xx-native                3.8.5-1vmw.600.0.0.2494585           VMWARE  VMwareCertified   2016-08-02
ata-pata-amd                   0.3.10-3vmw.600.0.0.2494585          VMware  VMwareCertified   2016-08-02
ata-pata-atiixp                0.4.6-4vmw.600.0.0.2494585           VMware  VMwareCertified   2016-08-02
ata-pata-cmd64x                0.2.5-3vmw.600.0.0.2494585           VMware  VMwareCertified   2016-08-02
ata-pata-hpt3x2n               0.3.4-3vmw.600.0.0.2494585           VMware  VMwareCertified   2016-08-02
ata-pata-pdc2027x              1.0-3vmw.600.0.0.2494585             VMware  VMwareCertified   2016-08-02
ata-pata-serverworks           0.4.3-3vmw.600.0.0.2494585           VMware  VMwareCertified   2016-08-02
ata-pata-sil680                0.4.8-3vmw.600.0.0.2494585           VMware  VMwareCertified   2016-08-02
ata-pata-via                   0.3.3-2vmw.600.0.0.2494585           VMware  VMwareCertified   2016-08-02
block-cciss                    3.6.14-10vmw.600.0.0.2494585         VMware  VMwareCertified   2016-08-02
cpu-microcode                  6.0.0-0.0.2494585                    VMware  VMwareCertified   2016-08-02
ehci-ehci-hcd                  1.0-3vmw.600.2.34.3620759            VMware  VMwareCertified   2016-08-02
elxnet                         10.2.309.6v-1vmw.600.0.0.2494585     VMware  VMwareCertified   2016-08-02
emulex-esx-elxnetcli           10.2.309.6v-0.0.2494585              VMware  VMwareCertified   2016-08-02
esx-base                       6.0.0-2.34.3620759                   VMware  VMwareCertified   2016-08-02
esx-dvfilter-generic-fastpath  6.0.0-0.0.2494585                    VMware  VMwareCertified   2016-08-02
esx-tboot                      6.0.0-2.34.3620759                   VMware  VMwareCertified   2016-08-02
esx-ui                         1.9.1-4393350                        VMware  VMwareCertified   2016-10-04
esx-xserver                    6.0.0-0.0.2494585                    VMware  VMwareCertified   2016-08-02
```

Once you have verified the package has been installed, exit your console session. You have to logout and log back in to your web UI session as well. After logging back in, navigate back to the *Host* > *Actions* > *Services* and disable the console and/or SSH services as necessary. **The console and/or SSH services should not remain enabled.**



Valhalla.blindseeker.com
Version: 6.0.0 Update 2 (Build 3620759)
State: Normal (not connected to any vCenter Server)
Uptime: 46.27 days

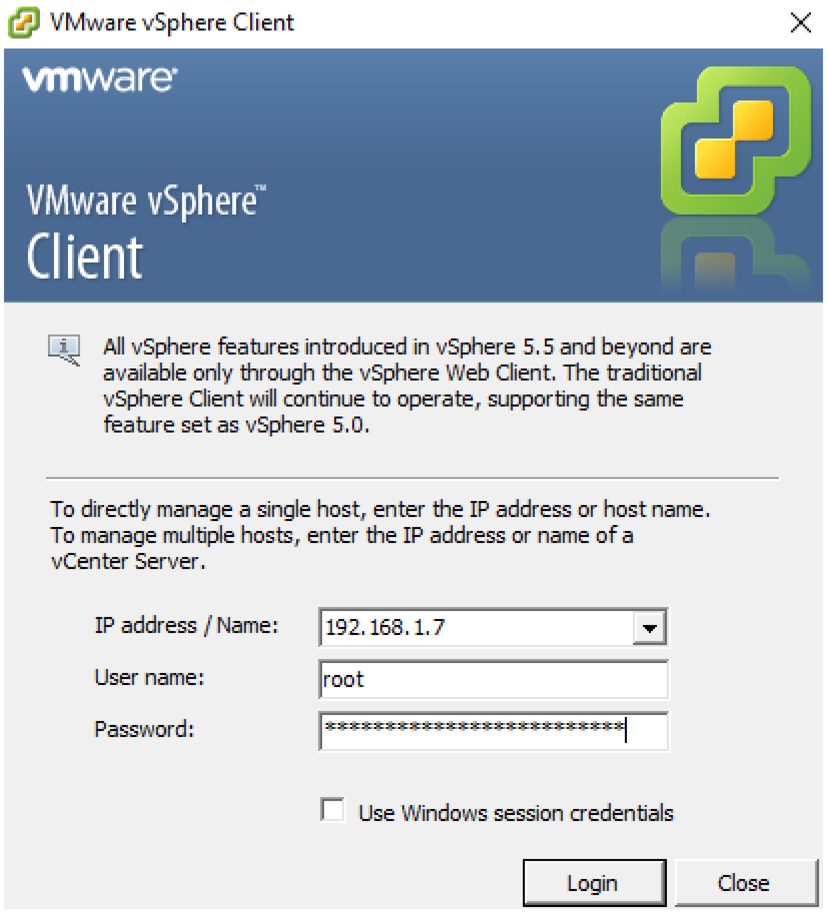
The ESXi shell is enabled on this host. You should disable the shell unless it is necessary for a

Manufacturer Supermicro

What if I don’t want to use experimental software?

Some of you might need to make your lab environment production quality, and unfortunately, most of the time, that means that you can’t install untested/unvetted software upgrades at will, so VMware Flings might be out of the question for you. The only other officially supported fallbacks for administering ESXi servers are using VMware’s vCenter Server product (which requires licensing and costs money), or to use the Windows vSphere Client. We will be focusing on using the Windows vSphere Client in our guide as a fallback.

This guide will teach you how to use Windows system and the vSphere Client to create virtual machine port groups, then return to the ESXi web interface to create the remaining port groups we need for our lab. Open the Windows vSphere Client, input the IP address of your ESXi server, username (root) and password, and click *Login*. Note that the client will likely give you an SSL cert warning if you are using ESXi’s defaults. Accept the warning and allow the connection.



All vSphere features introduced in vSphere 5.5 and beyond are available only through the vSphere Web Client. The traditional vSphere Client will continue to operate, supporting the same feature set as vSphere 5.0.

To directly manage a single host, enter the IP address or host name. To manage multiple hosts, enter the IP address or name of a vCenter Server.

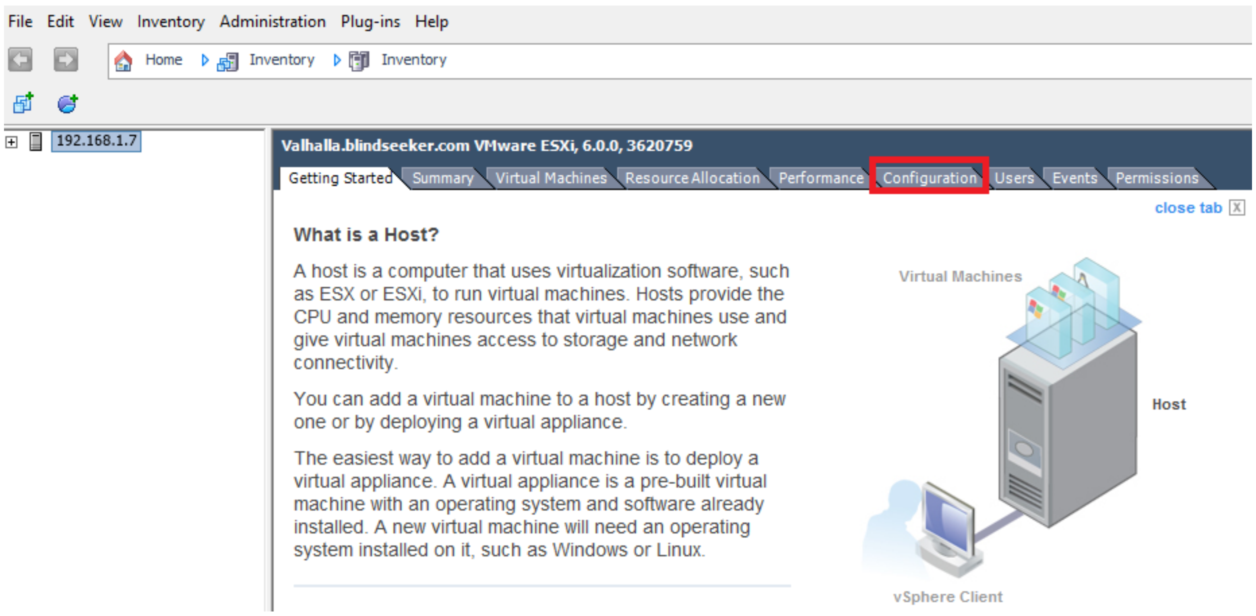
IP address / Name: 192.168.1.7

User name: root

Password: **************************

[ ] Use Windows session credentials

Once you have successfully logged in, the vSphere Client should display an interface with a series of tabs in the center pane for controlling the ESXi host. Select the *Configuration* tab.



Valhalla.blindseeker.com VMware ESXi, 6.0.0, 3620759

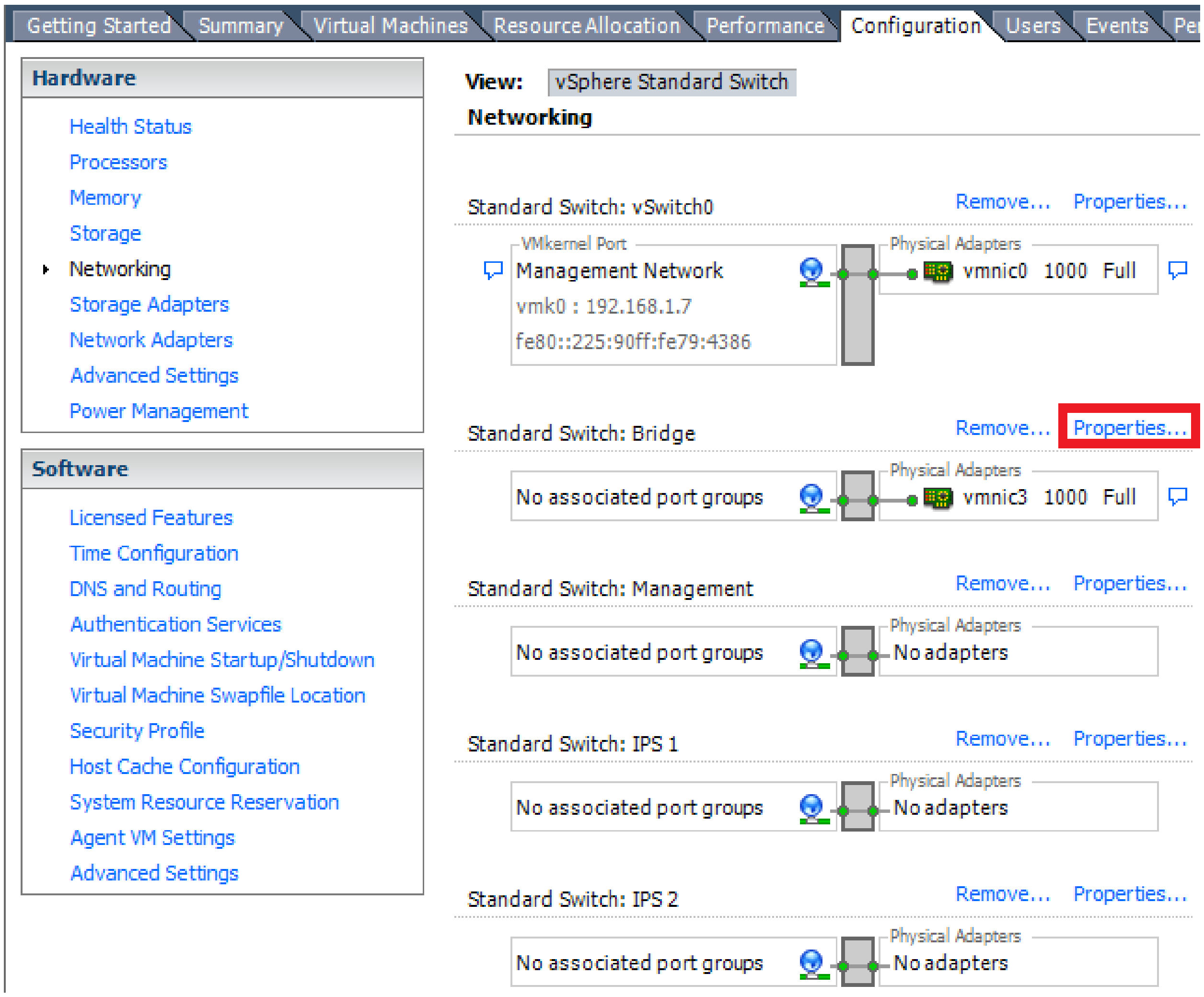
What is a Host?

A host is a computer that uses virtualization software, such as ESX or ESXi, to run virtual machines. Hosts provide the CPU and memory resources that virtual machines use and give virtual machines access to storage and network connectivity.

You can add a virtual machine to a host by creating a new one or by deploying a virtual appliance.

The easiest way to add a virtual machine is to deploy a virtual appliance. A virtual appliance is a pre-built virtual machine with an operating system and software already installed. A new virtual machine will need an operating system installed on it, such as Windows or Linux.

On the left side of the *Configuration* tab, you find a pane labeled *Hardware*. In this pane, click on *Networking*. A list of all the virtual switches and their logical network configuration appears. To start creating our initial port group,click on “Properties…” to the right of “Standard Switch: Bridge”. **If you don’t have a Bridge vSwitch as there is no second physical network card/port available on your server, use vSwitch0 to create your initial port group instead.**



Standard Switch: vSwitch0

VMkernel Port: Management Network, vmk0 : 192.168.1.7, fe80::225:90ff:fe79:4386

Physical Adapters: vmnic0 1000 Full

Standard Switch: Bridge

No associated port groups

Physical Adapters: vmnic3 1000 Full

Standard Switch: Management

No associated port groups

Physical Adapters: No adapters

Standard Switch: IPS 1

No associated port groups

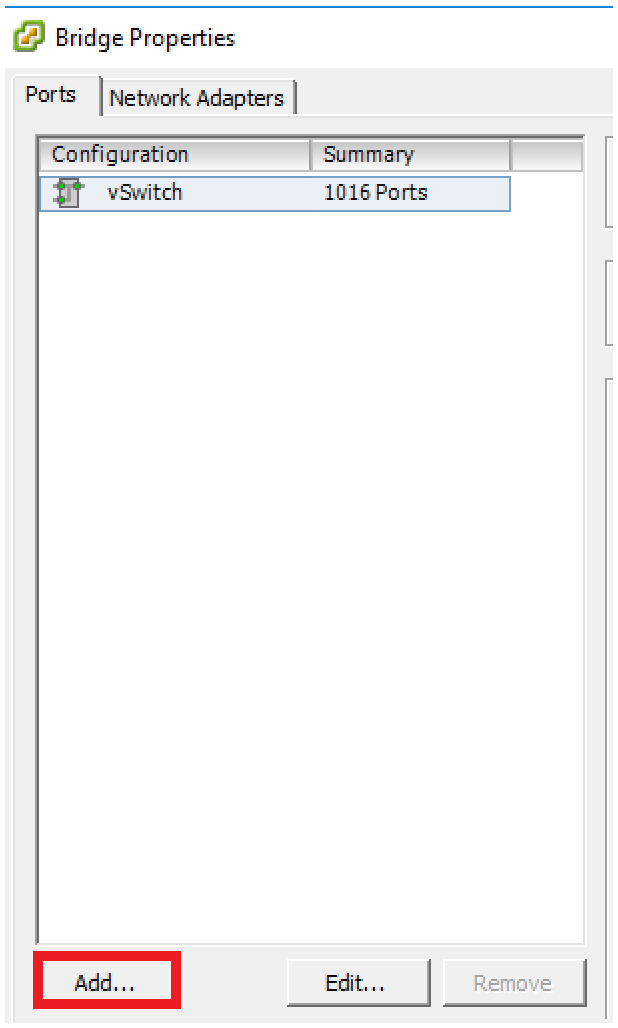
Physical Adapters: No adapters

Standard Switch: IPS 2

No associated port groups

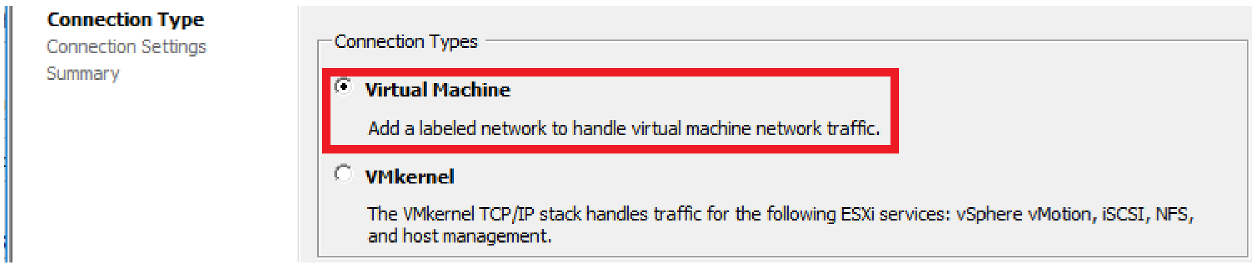
Physical Adapters: No adapters

A new window appears, showing you the properties for the Bridge vSwitch (or for vSwitch0, respectively). Click on the “Add…” button on the bottom left corner to start the Add Network wizard.



Configuration	Summary
vSwitch	1016 Ports

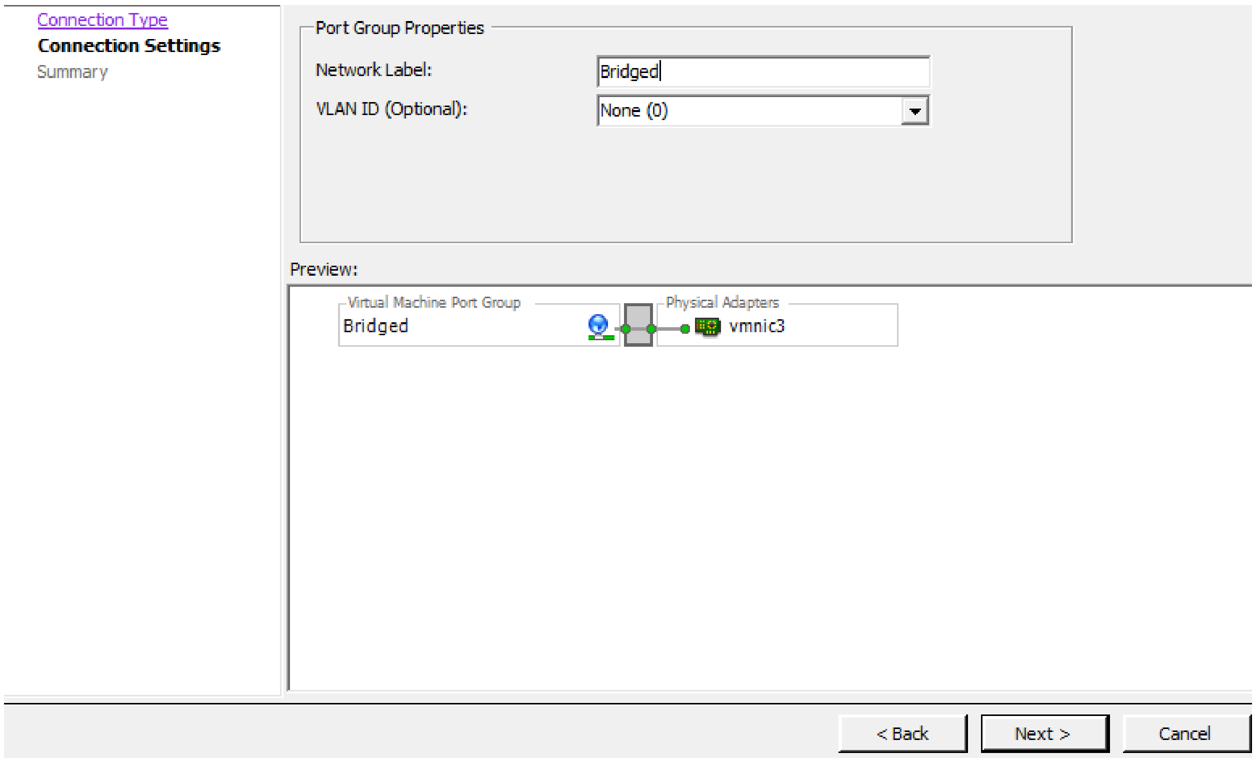
On the first screen, under “Connection Type”, select the radio button in front of *Virtual Machine*, then click Next.



(●) **Virtual Machine**
Add a labeled network to handle virtual machine network traffic.

( ) **VMkernel**
The VMkernel TCP/IP stack handles traffic for the following ESXi services: vSphere vMotion, iSCSI, NFS, and host management.

On the *Connection Settings* screen, in the *Port Group Properties* section, change the *Network Label:* input box to “Bridged”, then click Next again.

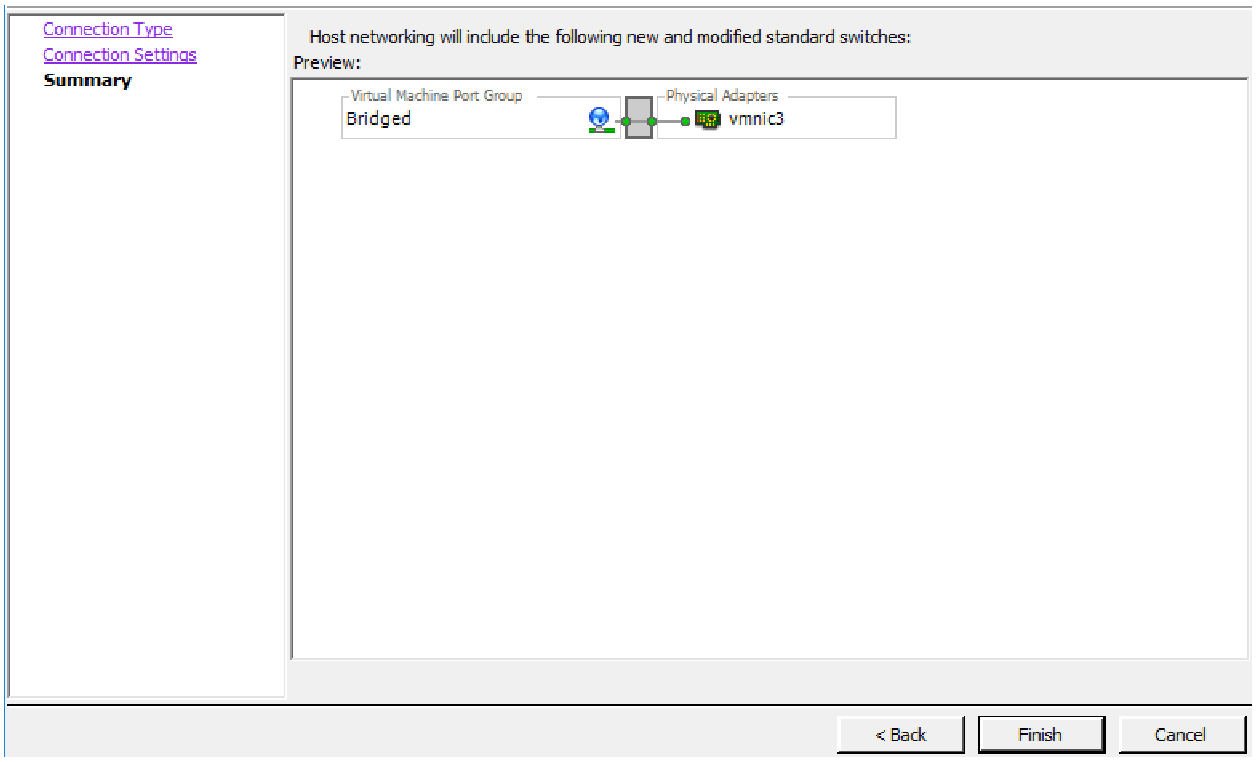


Network Label: Bridged

VLAN ID (Optional): None (0)

Bridged — vmnic3

The final page will confirm your settings. Click Finish.



Host networking will include the following new and modified standard switches:

Preview:

Virtual Machine Port Group: Bridged

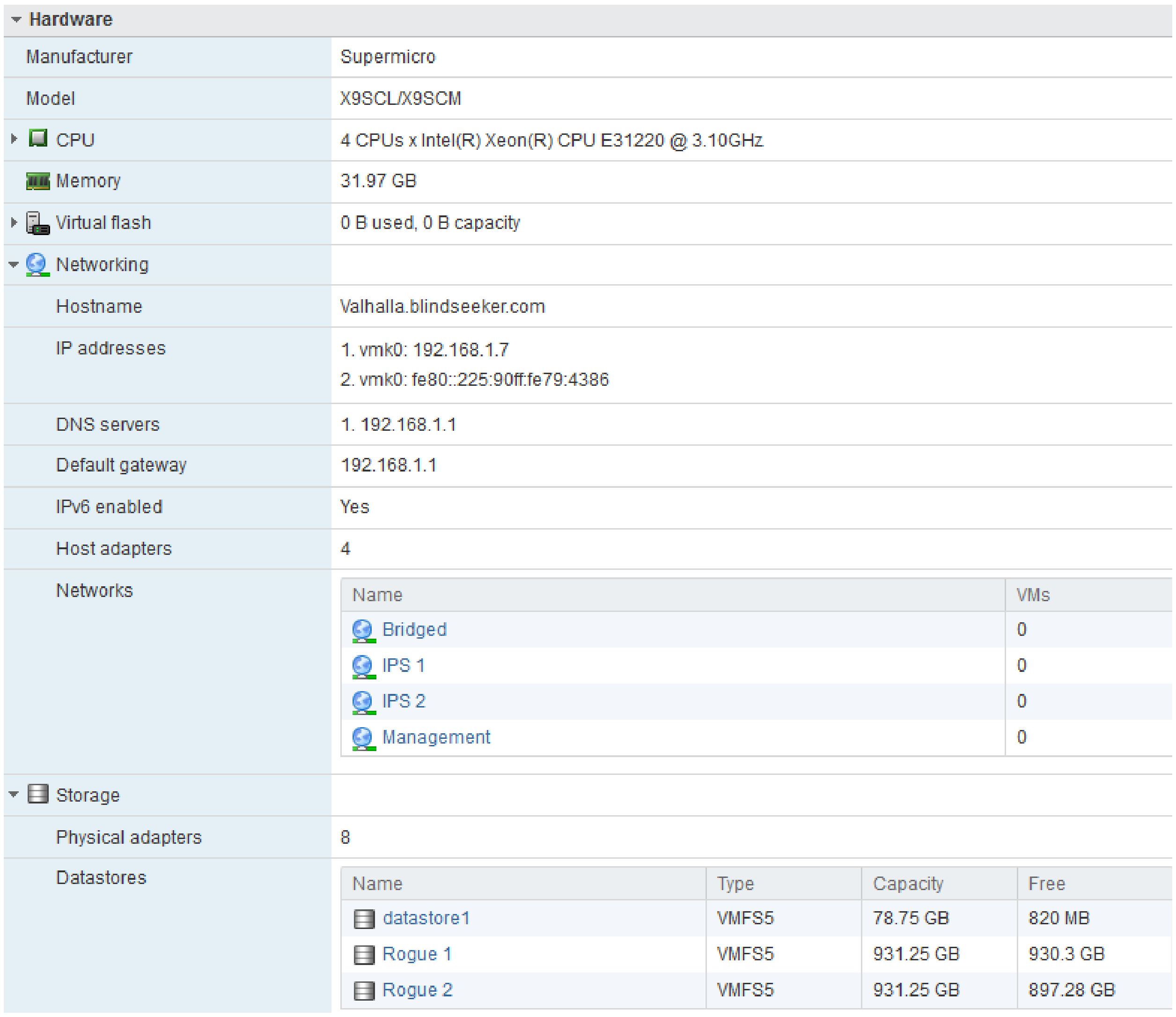
Physical Adapters: vmnic3

Close the Bridge Properties window (or the vSwitch0 Properties window, if that is where you just created the port group on). Back under *Configuration* > *Hardware* > *Networking*, the new port group should appear on the Bridge virtual switch (or on vSwitch0, respectively). If you'd like, you could repeat the process and create the remaining port groups we need on the other virtual switches using the vSphere Client as well, but with the first virtual machine port group generated, this can be done via the ESXi web interface, too.

Final Flight Check

At this point, you should have ESXi installed. You should be able to access the web interface from a management workstation running any operating system, so long as you have a modern web browser. You may or may not be using a Windows system as your "management workstation" for managing and configuring the ESXi host, but if you are, you installed the vSphere Client is installed in case you run into web UI issues.. You should have have 5 virtual switches (four, if you only have a single physical network port available), and at least 5 port groups (including the default "Management Network" port group used for managing ESXi).

You'll want to make sure that ESXi recognizes all the RAM installed in your server, and that all of the disks (or RAID arrays, if you're fancy) attached to your ESXi server are available as well. Quickest way to do this is by checking the host's *Hardware* status section. To get there, fire up the ESXi web interface, log in, and under the Navigator pane, select *Host*. The status report titled *Hardware* can be found in the center pane:



Hardware

Manufacturer	Supermicro
Model	X9SCL/X9SCM
CPU	4 CPUs x Intel(R) Xeon(R) CPU E31220 @ 3.10GHz
Memory	31.97 GB
Virtual flash	0 B used, 0 B capacity
Networking	
Hostname	Valhalla.blindseeker.com
IP addresses	1. vmk0: 192.168.1.7 2. vmk0: fe80::225:90ff:fe79:4386
DNS servers	1. 192.168.1.1
Default gateway	192.168.1.1
IPv6 enabled	Yes
Host adapters	4
Networks	

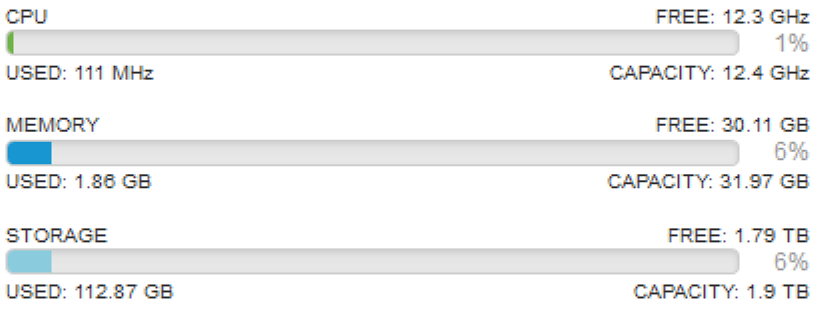
Name	VMs
Bridged	0
IPS 1	0
IPS 2	0
Management	0

Storage	
Physical adapters	8
Datastores	

Name	Type	Capacity	Free
datastore1	VMFS5	78.75 GB	820 MB
Rogue 1	VMFS5	931.25 GB	930.3 GB
Rogue 2	VMFS5	931.25 GB	897.28 GB

The *Hardware* overview provides information on how much RAM, how many network cards/ports, and how many CPU cores were detected by the system. Additionally, it displays the network settings of the actual ESXi host (IP address, gateway, DNS, etc.). Finally, it lists all the configured port groups, and all of the connected data stores (disk drives we can use for storing VMs and other files).

In the upper right corner of the center pane, there are a set of bars that give a quick at a glance view of the resource utilization on the server. For instance, you can see that I have approximately 1.9TB of space available, with a little over 100GB of space used, I have 32GB of RAM in total, with 1.86GB in use, and CPU usage is minimal at 111mhz out of an available 12.4GHz (3.10 Ghz x 4 cores). This system is as ready as it is going to get for hosting VMs.

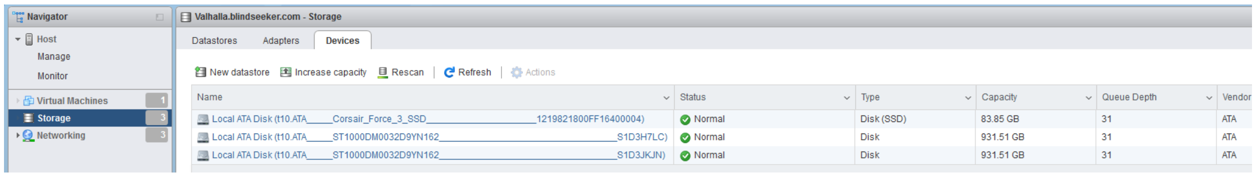


CPU FREE: 12.3 GHz
1%
USED: 111 MHz CAPACITY: 12.4 GHz

MEMORY FREE: 30.11 GB
6%
USED: 1.86 GB CAPACITY: 31.97 GB

STORAGE FREE: 1.79 TB
6%
USED: 112.87 GB CAPACITY: 1.9 TB

If for some reason you think that not all of your server's storage space is detected or utilized, consider visiting the *Storage* interface under the *Navigator* pane. Try to detect if for some reason your array or disks simply weren't formatted or converted to datastores by click the *Devices* tab. If there are some uninitialized disks or disks with free space listed, then format and initialize the disks as needed.



Name	Status	Type	Capacity	Queue Depth	Vendor
Local ATA Disk (t10.ATA_____Corsair_Force_3_SSD____________________1219821800FF16400004)	Normal	Disk (SSD)	83.85 GB	31	ATA
Local ATA Disk (t10.ATA_____ST1000DM0032D9YN162________________________________S1D3H7LC)	Normal	Disk	931.51 GB	31	ATA
Local ATA Disk (t10.ATA_____ST1000DM0032D9YN162________________________________S1D3JKJN)	Normal	Disk	931.51 GB	31	ATA

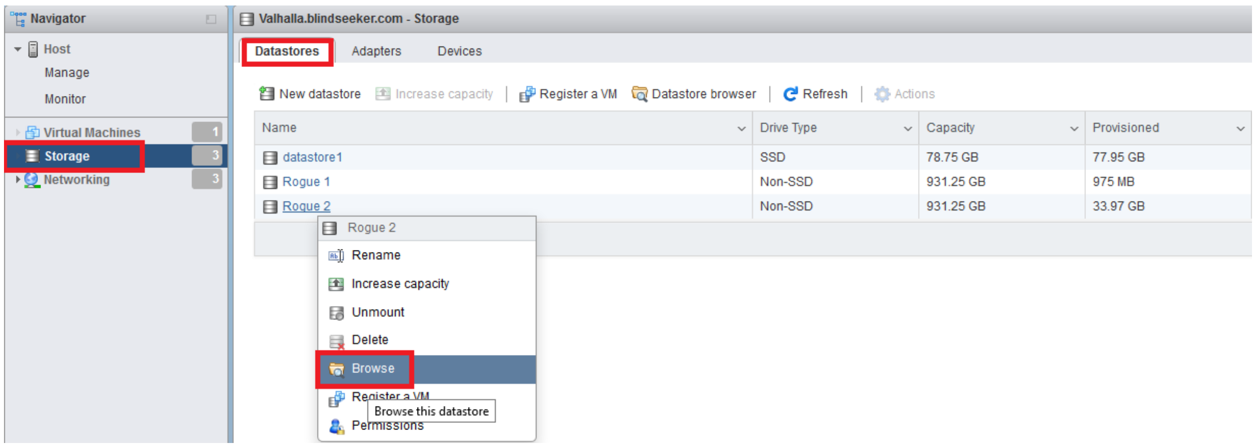
In regards to other hardware-related issues, make sure all your hardware is seated and installed properly. In case your network cards haven't been recognized, you may have to acquire new ones that are supported by ESXi's driver base. If RAM or CPU is not recognized by ESXi, verify that the hardware is detected by the system BIOS; it might turn out to be as simple as a hardware malfunction, or enabling a setting in the BIOS. If everything in your server shows up properly and you have all the resources you need, let's move on.

Creating the First VM, pfSense

pfSense is the keystone holding this entire configuration together. Personally, it's my favorite firewall distro due to ease of use, the amount of functionality it includes out of the box, combined with a plugin/add-on system for additional functionality. If you have the CPU, RAM, and disk, pfSense can easily be converted into a so-called “Next-Generation” firewall.

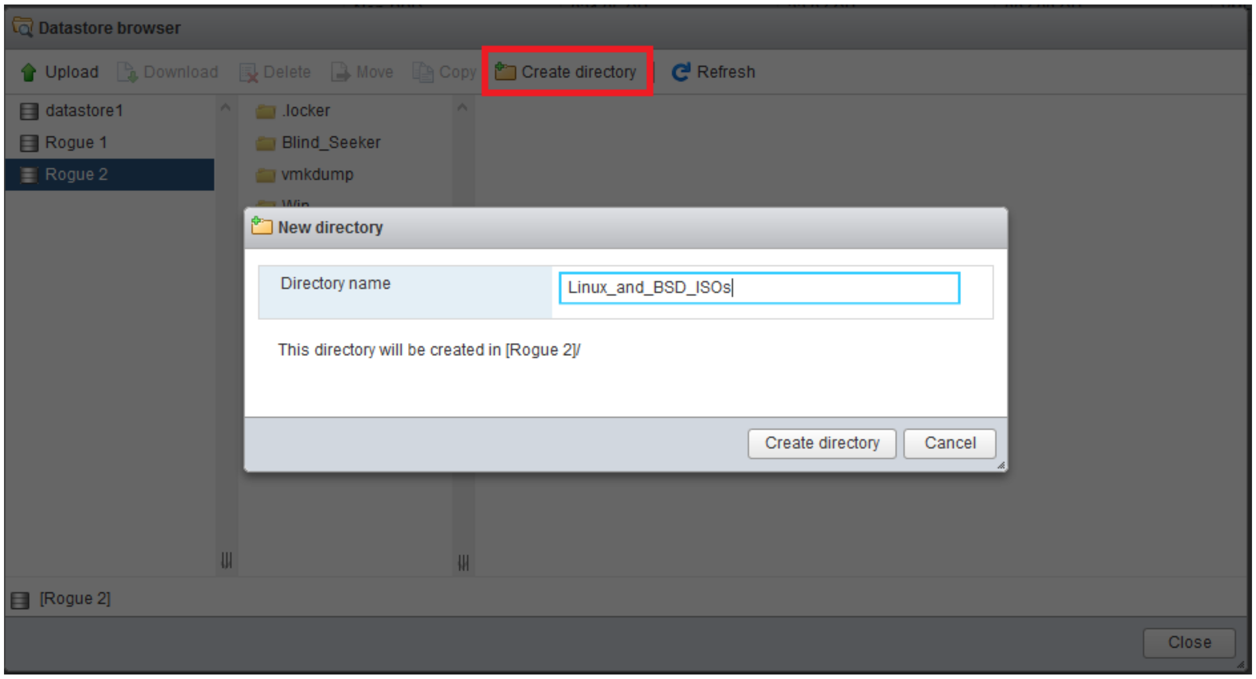
To start, make your way to https://www.pfsense.org/download/. Download the latest installation ISO for the amd64 architecture .Once you have downloaded the ISO, depending on what operating system you are using to manage your ESXi server, you may need to acquire a compression utility, as its maintainers distribute pfSense as an ISO image file compressed with `gzip`. This means that we’ll need to decompress the ISO file, using for example `gunzip` (Mac/Linux) or 7-Zip (Windows).

To make it available for use, we have to upload the decompressed installation image file to one of the datastores of our ESXi server. On the web interface, under the Navigator pane, click on *Storage*, and ensure the *Datastores* tab is selected. In the displayed list, right click on the datastore you wish to store the ISO on, then click *Browse*. In my case, I chose the datastore named “Rogue 2”.



Name	Drive Type	Capacity	Provisioned
datastore1	SSD	78.75 GB	77.95 GB
Rogue 1	Non-SSD	931.25 GB	975 MB
Rogue 2	Non-SSD	931.25 GB	33.97 GB

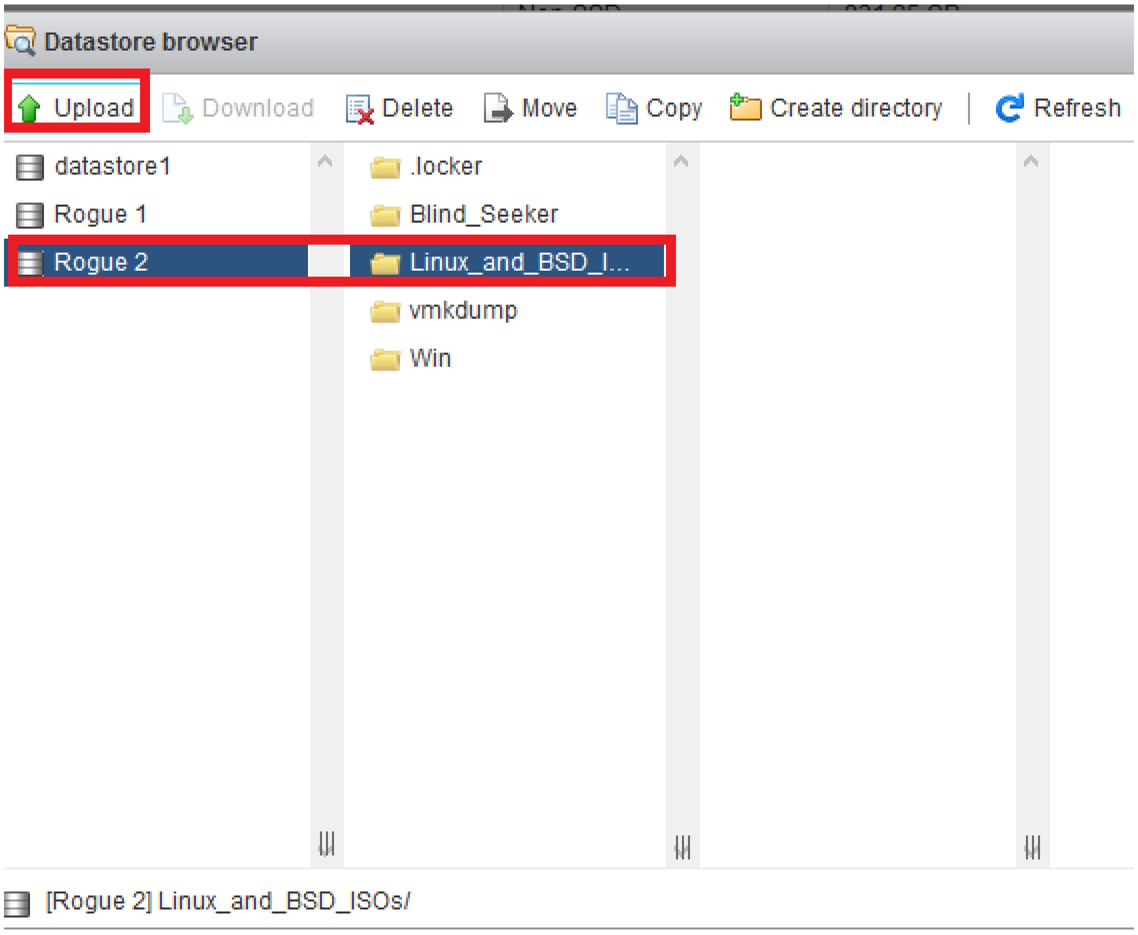
In the Datastore browser window that appeared, click on the *Create directory* option. As we're going to store all of our Linux and BSD ISOs in this directory, let's name it "Linux_and_BSD_ISOs". Click the *Create directory* button.



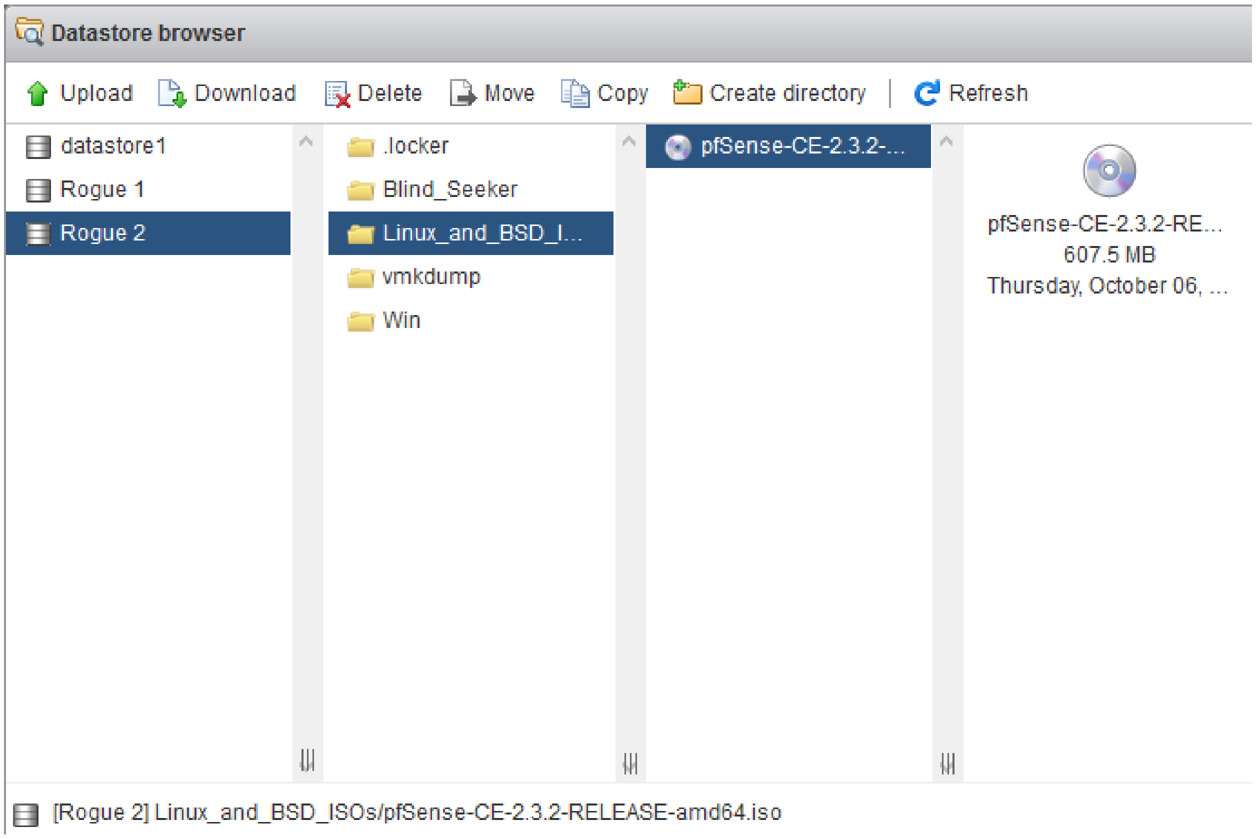
Directory name: Linux_and_BSD_ISOs

This directory will be created in [Rogue 2]/

Back in the Datastore browser, select the newly created directory, then click *Upload* in the upper left portion of the window.



This will open a file browser on your local machine. Navigate to where you downloaded the pfSense ISO. If you haven't already, make sure that the ISO has been decompressed. After selecting the ISO file, the process of uploading it to the ESXi server's datastore is started. On completion, you should be able to select the ISO file in the Datastore browser window to display its file statistics.

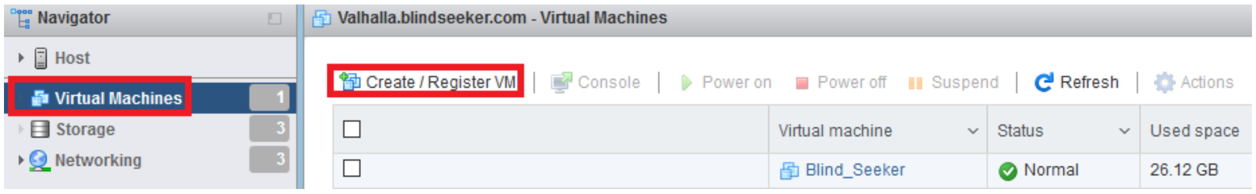


[Rogue 2] Linux_and_BSD_ISOs/pfSense-CE-2.3.2-RELEASE-amd64.iso

After successfully uploading the ISO file to the desired datastore, close the Datastore browser window.

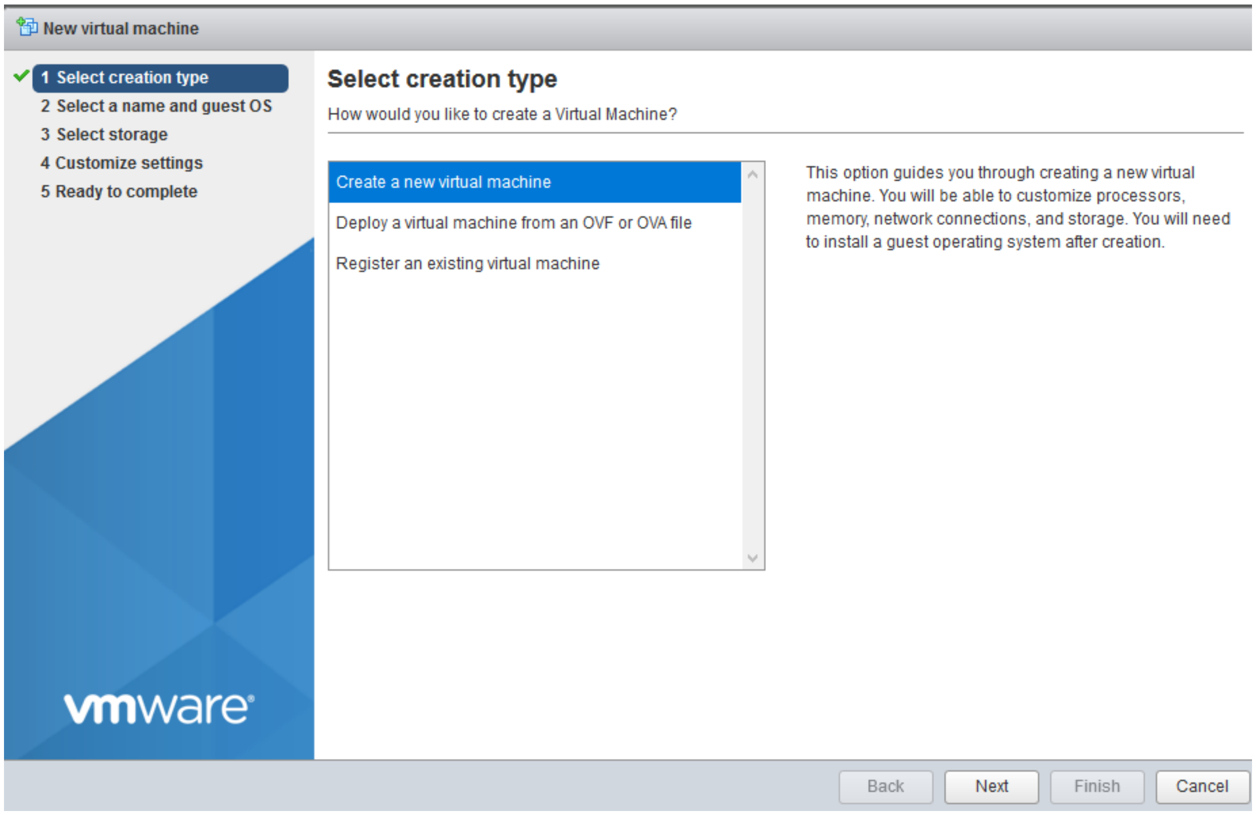
Adding a New VM

Now that we have the ISO image to boot from ready, let's create the VM for pfSense. On the ESXi web interface, click on *Virtual Machines* under the Navigator pane. This brings you to a list of currently registered VMs on the server. On your system, this list should be empty ; in my case, a virtual machine named BlindSeeker (my previously mentioned web server) is already available. Click on *Create/Register VM* to start the *New virtual machine* wizard.



	Virtual machine	Status	Used space
☐	Blind_Seeker	Normal	26.12 GB

On the first screen, select the option *Create a new virtual machine*, then click the *Next* button.



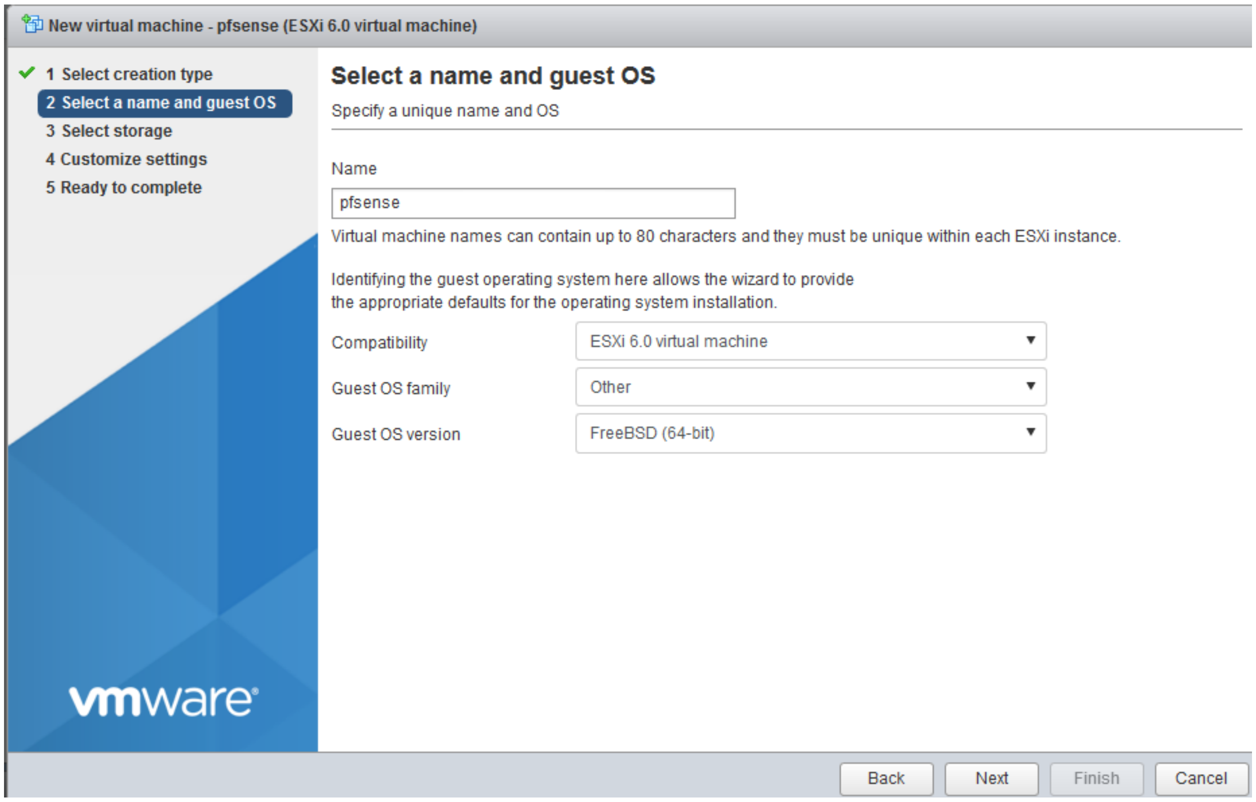
Select creation type

How would you like to create a Virtual Machine?

- Create a new virtual machine
- Deploy a virtual machine from an OVF or OVA file
- Register an existing virtual machine

This option guides you through creating a new virtual machine. You will be able to customize processors, memory, network connections, and storage. You will need to install a guest operating system after creation.

The next page asks to provide a name for the VM, and choose the family and the version of the OS that will be used later. Enter “pfSense” in the *Name* field, choose *Other* from the OS family drop-down, select *FreeBSD (64-bit)* as the OS version , then click *Next*.


New virtual machine - pfsense (ESXi 6.0 virtual machine)

1 Select creation type
2 Select a name and guest OS
3 Select storage
4 Customize settings
5 Ready to complete

Select a name and guest OS

Specify a unique name and OS

Name: pfsense

Virtual machine names can contain up to 80 characters and they must be unique within each ESXi instance.

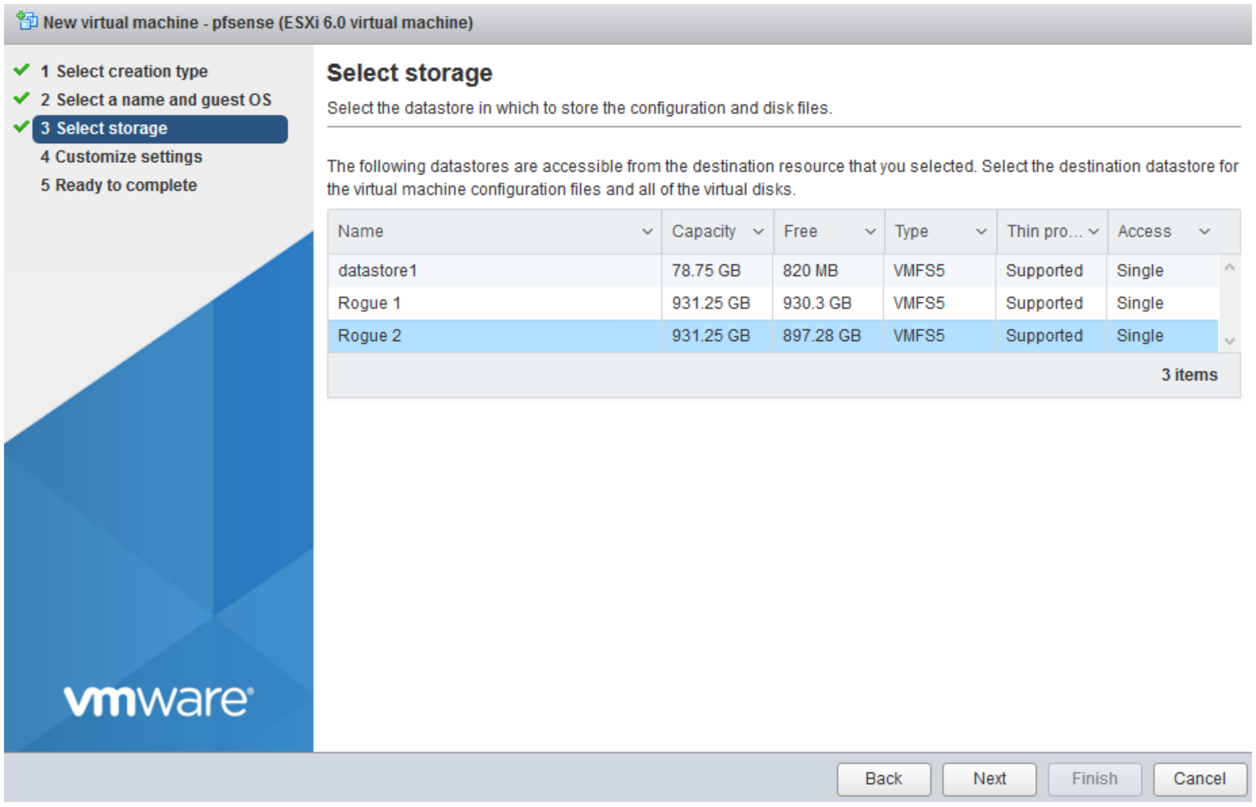
Identifying the guest operating system here allows the wizard to provide the appropriate defaults for the operating system installation.

Compatibility: ESXi 6.0 virtual machine

Guest OS family: Other

Guest OS version: FreeBSD (64-bit)

The next section offers a list of datastores available on the ESXi host. From that list, select the one where the files related to the virtual machine should be stored. Click on the datastore of your choice to select it (if you have more than one datastore), then click *Next* to continue.



Select storage

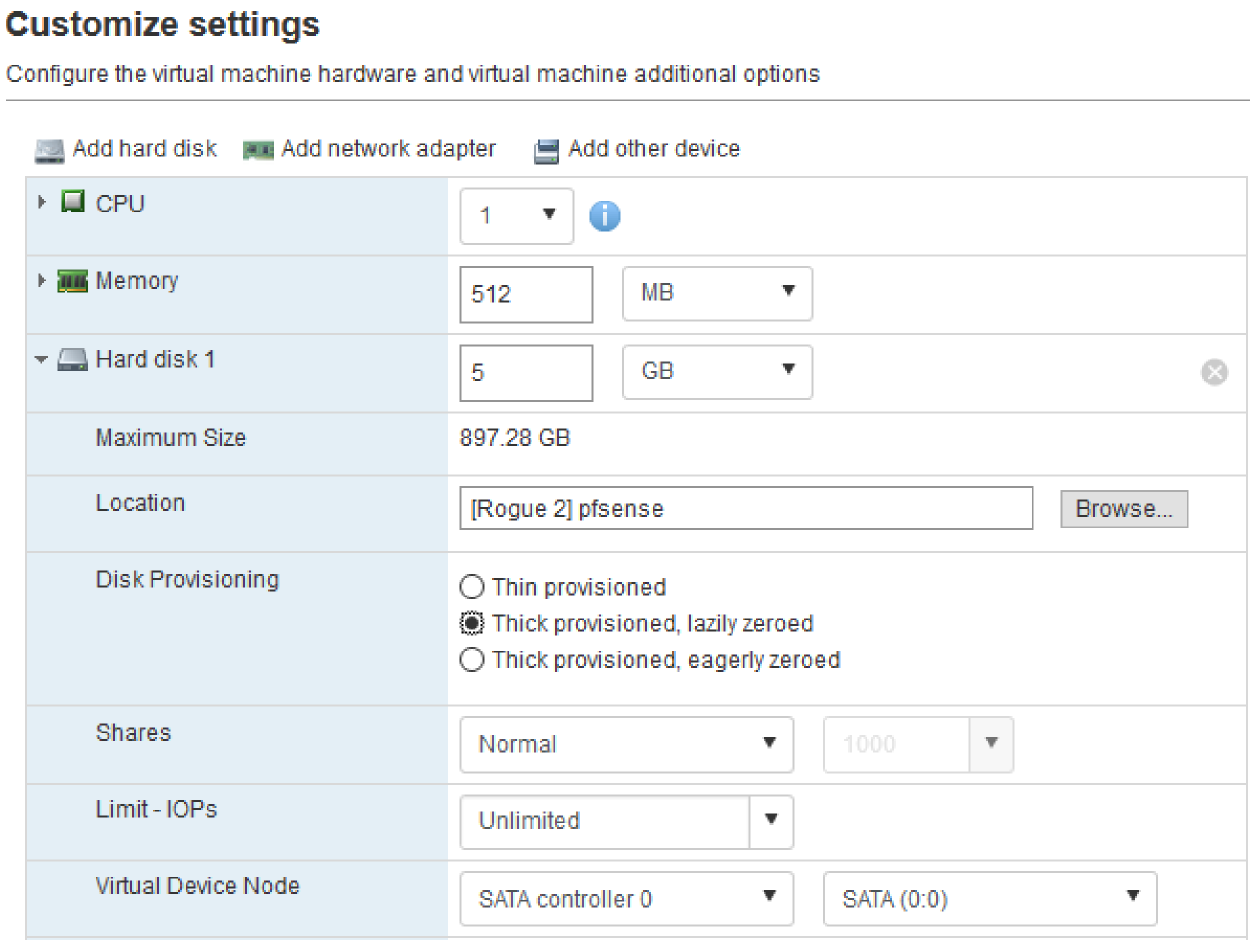
Select the datastore in which to store the configuration and disk files.

The following datastores are accessible from the destination resource that you selected. Select the destination datastore for the virtual machine configuration files and all of the virtual disks.

Name	Capacity	Free	Type	Thin pro...	Access
datastore1	78.75 GB	820 MB	VMFS5	Supported	Single
Rogue 1	931.25 GB	930.3 GB	VMFS5	Supported	Single
Rogue 2	931.25 GB	897.28 GB	VMFS5	Supported	Single

3 items

The next page allows us to make various customizations to our virtual machine's hardware. The first and most obvious settings displayed are CPU, Memory and Hard disk space. For the pfSense VM, adjust the the corresponding values to 1 CPU core, 512 MB of RAM, and 5GB of disk space. As we change the configuration how the disk space is provided to the virtual machine, click the triangle next to *Hard disk 1* to bring up the additional hard disk settings. In the *Disk Provisioning* section, select the radio button next to *Thick provisioned, lazily zeroed*, and in the *Virtual Device Node* section, choose *SATA controller 0* from the drop-down on the left. The one to the right should read *SATA (0:0)*.

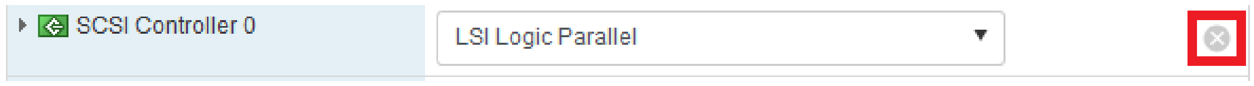


Customize settings

Configure the virtual machine hardware and virtual machine additional options

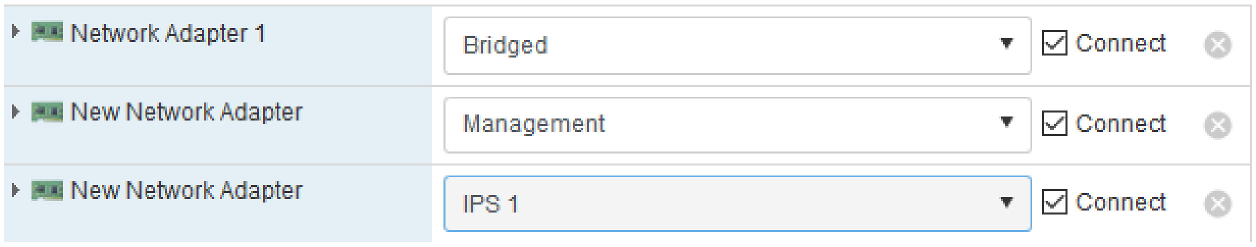
CPU	1	
Memory	512	MB
Hard disk 1	5	GB
Maximum Size	897.28 GB	
Location	[Rogue 2] pfsense	Browse...
Disk Provisioning	○ Thin provisioned ● Thick provisioned, lazily zeroed ○ Thick provisioned, eagerly zeroed	
Shares	Normal	1000
Limit - IOPs	Unlimited	
Virtual Device Node	SATA controller 0	SATA (0:0)

Next, in the *SCSI Controller 0* field, click the X on the right to remove this device entirely as it isn't needed.



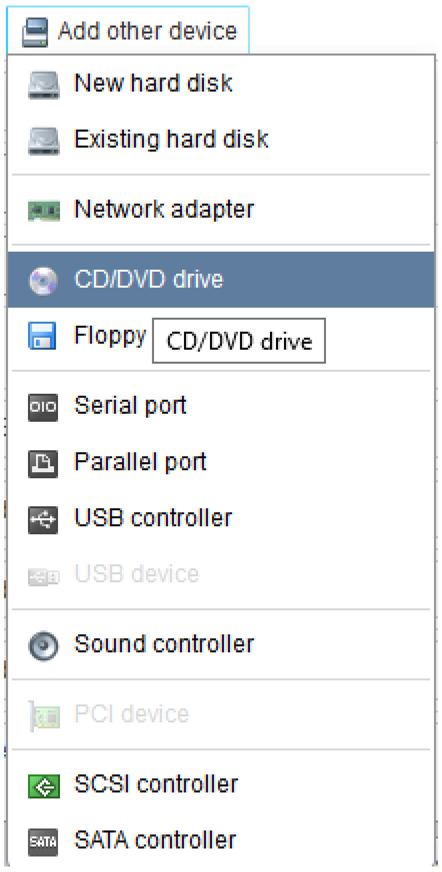
SCSI Controller 0	LSI Logic Parallel

As we need additional NICs on this VM, click twice on the button at the top of the window named *Add network adapter*. Two more network cards appear right below *Network Adapter 1*, each of them labeled *New Network Adapter*. Configure the first network adapter to connect to the *Bridged* port group, the second one to connect to the *Management* port group, and the third NIC to connect to the *IPS 1* port group. Ensure that the checkbox labeled *Connect* is checked for all three adapters.

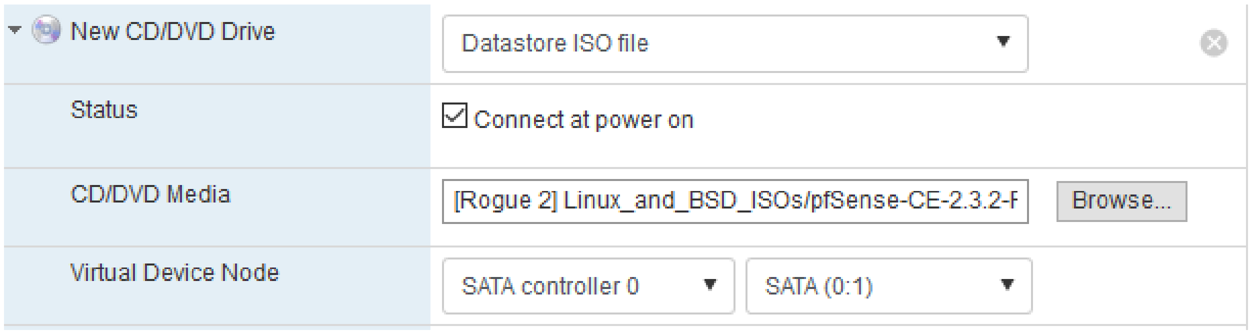


Network Adapter 1	Bridged	✓ Connect
New Network Adapter	Management	✓ Connect
New Network Adapter	IPS 1	✓ Connect

Finally, click the *Add other device* option at the top of the window, and from the appearing drop-down menu, select *CD/DVD drive*.

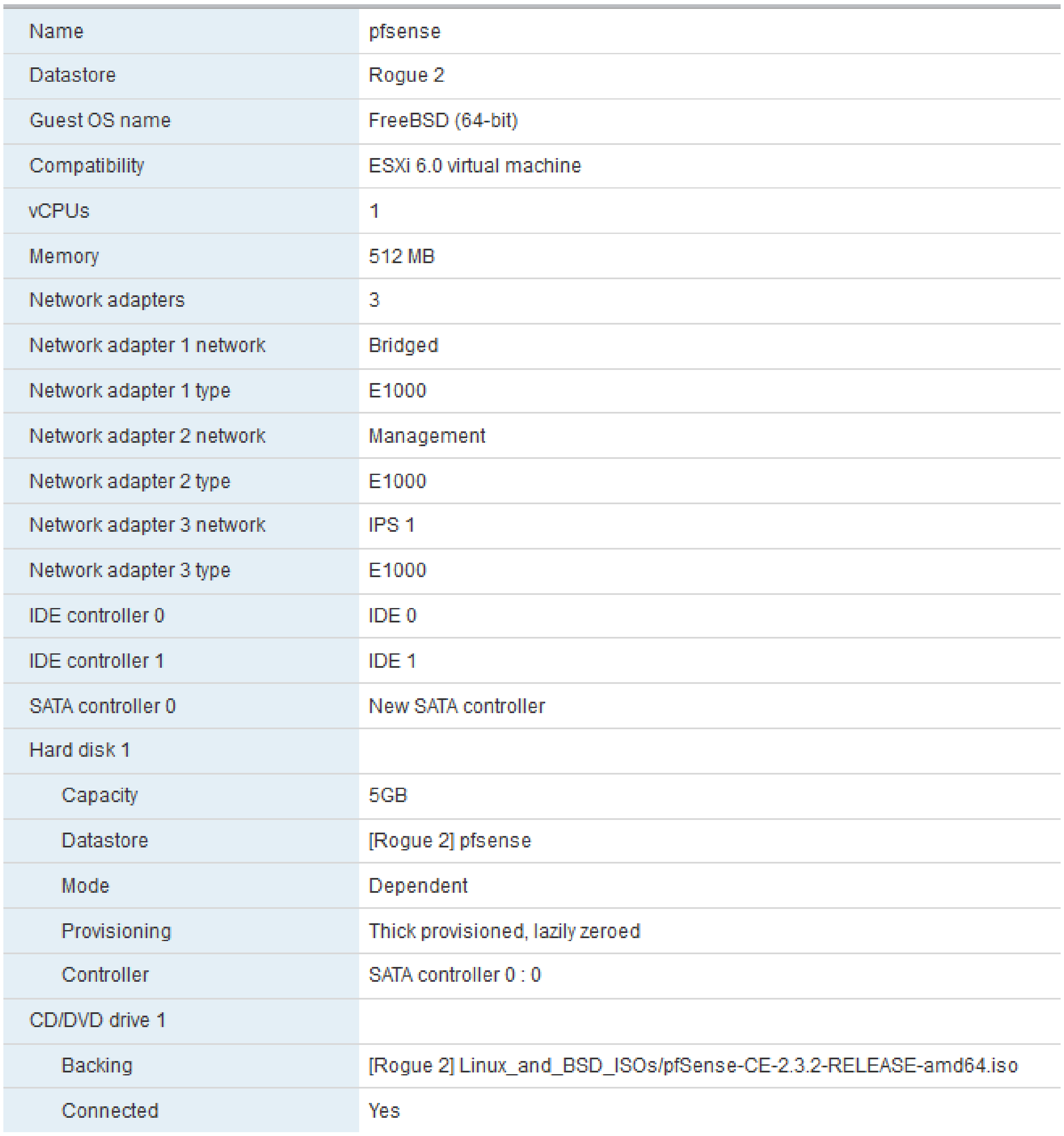


An entry labeled *New CD/DVD Drive* should appear in the overview, immediately underneath the network adapters. Click on the triangle next to this text to bring up the settings for the virtual CD/DVD drive. Ensure that the checkbox *Connect at power on* is checked, then in the drop-down field that is most probably set to *Host device*, change this value to *Datastore ISO file*. When doing so, the *Datastore browser* window should open up. Navigate to the location you uploaded the pfSense ISO to, and select the file.



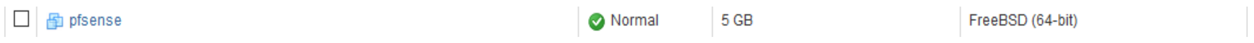
New CD/DVD Drive	Datastore ISO file
Status	✓ Connect at power on
CD/DVD Media	[Rogue 2] Linux_and_BSD_ISOs/pfSense-CE-2.3.2-F
Virtual Device Node	SATA controller 0 / SATA (0:1)

Afterwards, click the Next button to move on to the final screen of the wizard. This is a confirmation page, allowing to review all the settings selected for this VM. Your settings should look similar to this:



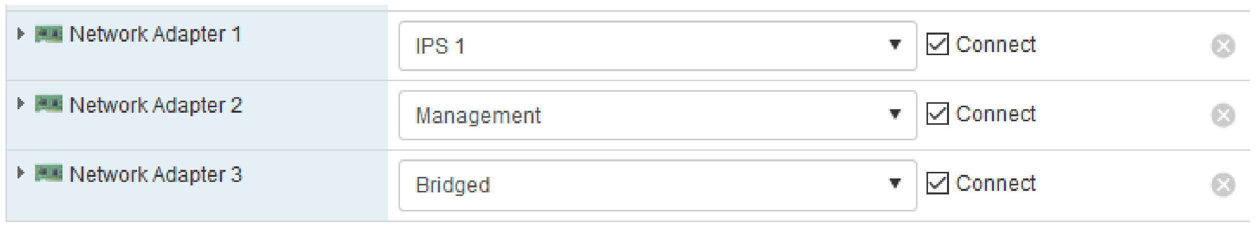
Name	pfsense
Datastore	Rogue 2
Guest OS name	FreeBSD (64-bit)
Compatibility	ESXi 6.0 virtual machine
vCPUs	1
Memory	512 MB
Network adapters	3
Network adapter 1 network	Bridged
Network adapter 1 type	E1000
Network adapter 2 network	Management
Network adapter 2 type	E1000
Network adapter 3 network	IPS 1
Network adapter 3 type	E1000
IDE controller 0	IDE 0
IDE controller 1	IDE 1
SATA controller 0	New SATA controller
Hard disk 1	
Capacity	5GB
Datastore	[Rogue 2] pfsense
Mode	Dependent
Provisioning	Thick provisioned, lazily zeroed
Controller	SATA controller 0 : 0
CD/DVD drive 1	
Backing	[Rogue 2] Linux_and_BSD_ISOs/pfSense-CE-2.3.2-RELEASE-amd64.iso
Connected	Yes

Click *Finish* and allow it some time for ESXi to create the new virtual machine. On completion, you should be returned to the virtual machines listing, and the new VM should be displayed there.



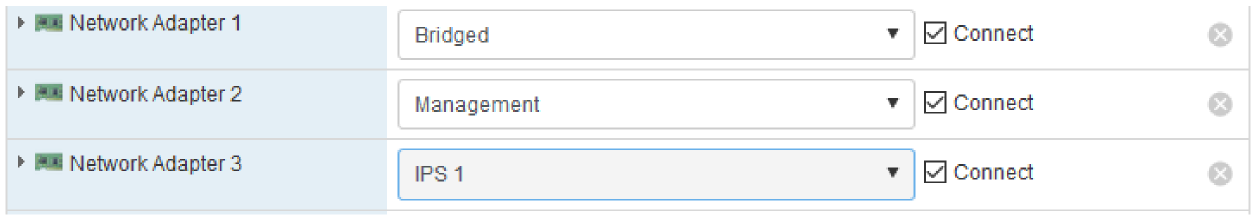
☐	pfsense	Normal	5 GB	FreeBSD (64-bit)

Note: I realized that, for whatever reason, ESXi appears to reverse the order of the network adapters, at least in regard to what one would expect with the hardware overview from the New virtual machine wizard in mind. If you right click on the VM you just created, and select *Edit Settings*, you may find that *Network Adapter 1* is set to *IPS 1* as its port group, *Network Adapter 2* is set to *Management* and *Network Adapter 3* is set to *Bridged*.



Network Adapter 1	IPS 1	☑ Connect
Network Adapter 2	Management	☑ Connect
Network Adapter 3	Bridged	☑ Connect

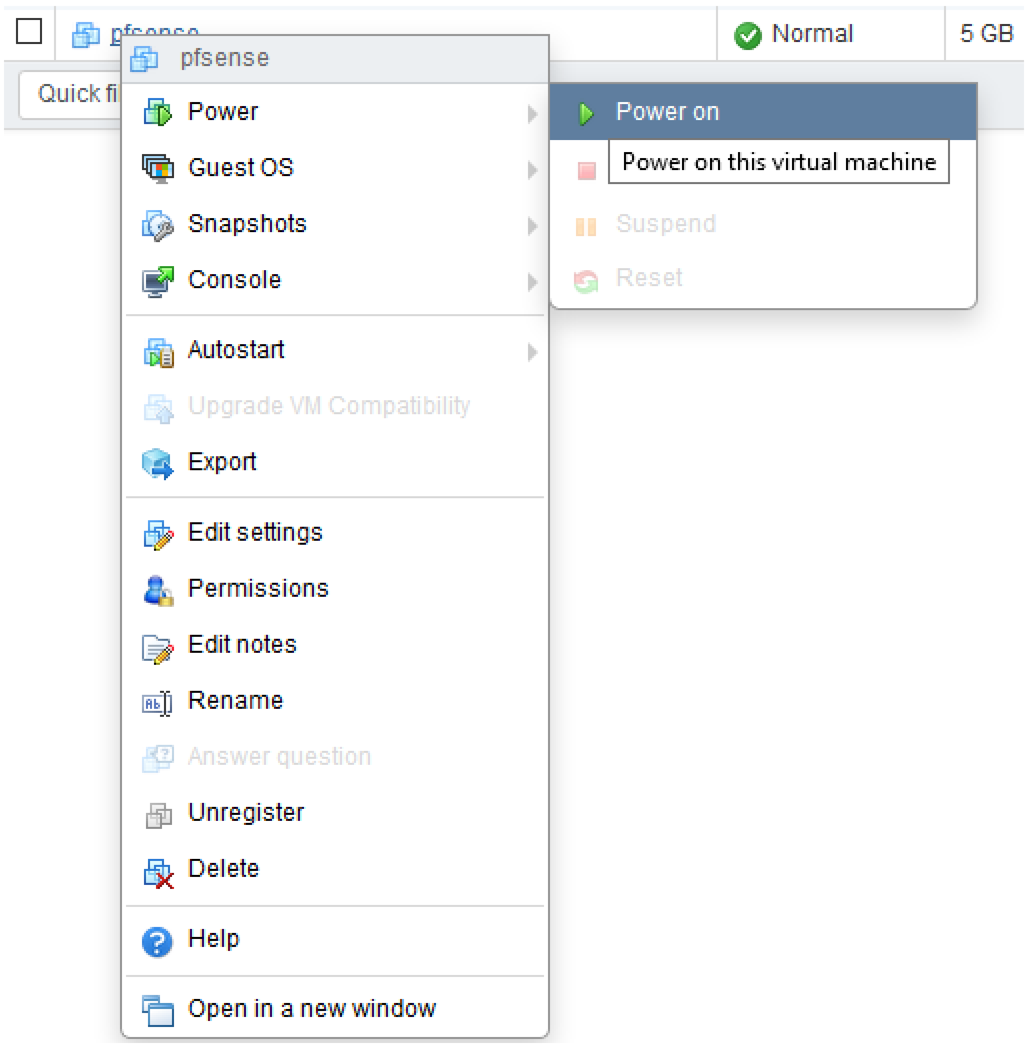
In fact, it appears as though ESXi considers the most recently created network adapter to be *Network Adapter 1*, then 2, and so on. For the setup of the virtual machine, this ordering doesn't really matter. To fix this, simply swap the port group settings for Adapter 1 and Adapter 3, then click the Save button. The only other VM with multiple network adapters that we are going to create, is the IPS virtual machine. So don't worry too much about it, just keep this in mind.



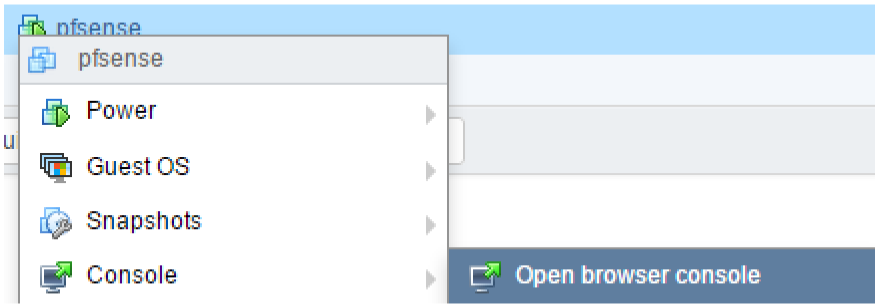
Network Adapter 1	Bridged	☑ Connect
Network Adapter 2	Management	☑ Connect
Network Adapter 3	IPS 1	☑ Connect

Installing pfSense

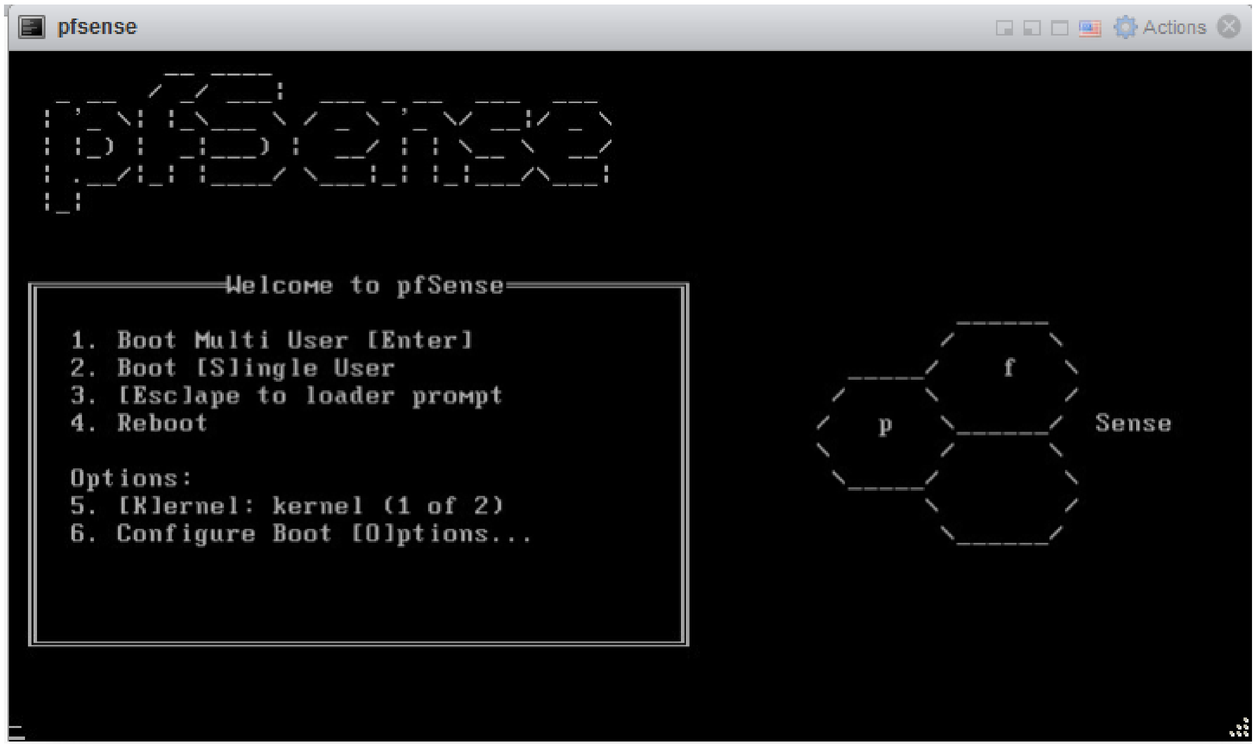
pfSense is really easy to install. Start the virtual machine by navigating to *Virtual Machines* in the *Navigator* pane, then either right click pfSense and select *Power > Power on* from the menu, or click to highlight pfSense, and click *Power on* on the option bar above the VM list.



In either case, the virtual machine should start up. Open a console window to interact with the VM. To do so, right click on pfSense in the virtual machines list again. This time, select *Console > Browser Console* to open a pop-up window that will act as a console to interact with the system running in the virtual machine. Click on this console lets you interact with the VM with your keyboard and mouse. To release the keyboard and mouse, use the key combination Ctrl + Alt.

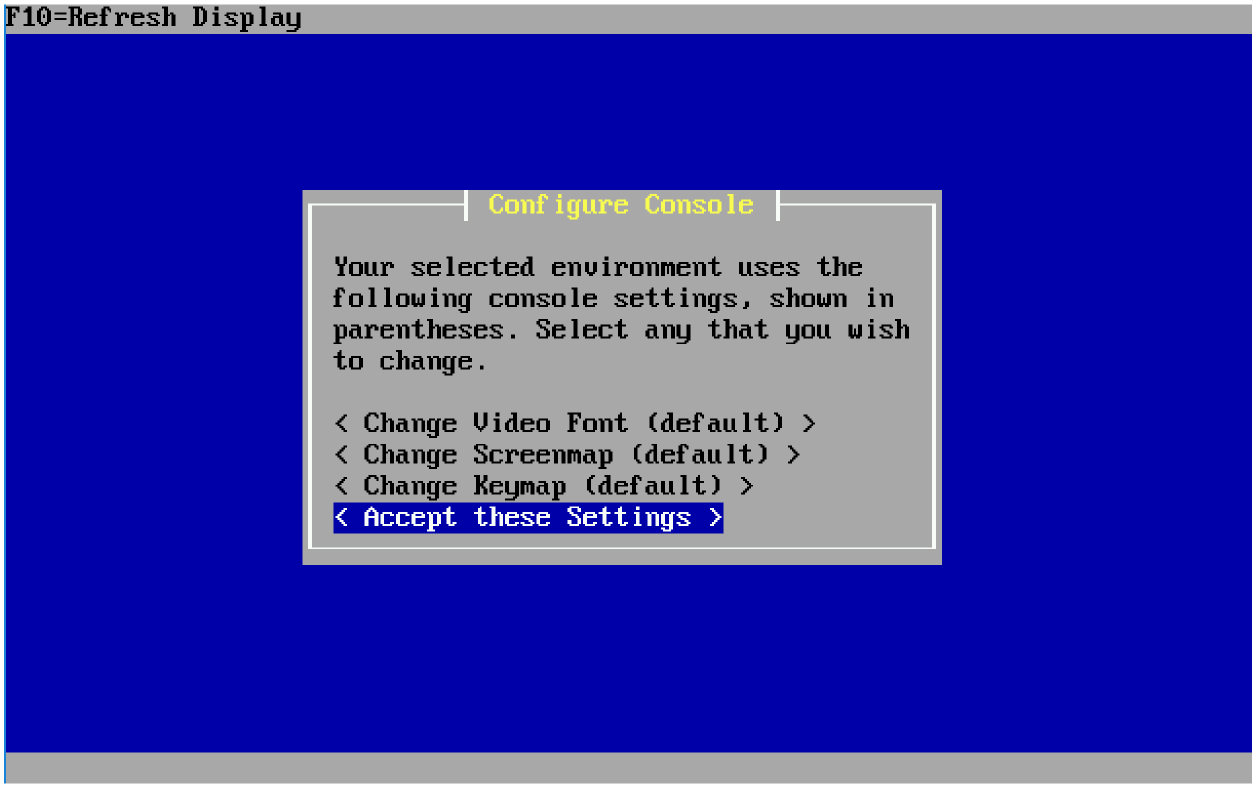


After booting the VM and attaching to the virtual machine's console, select option 1, *Boot Multi User [Enter]* by hitting the enter key.



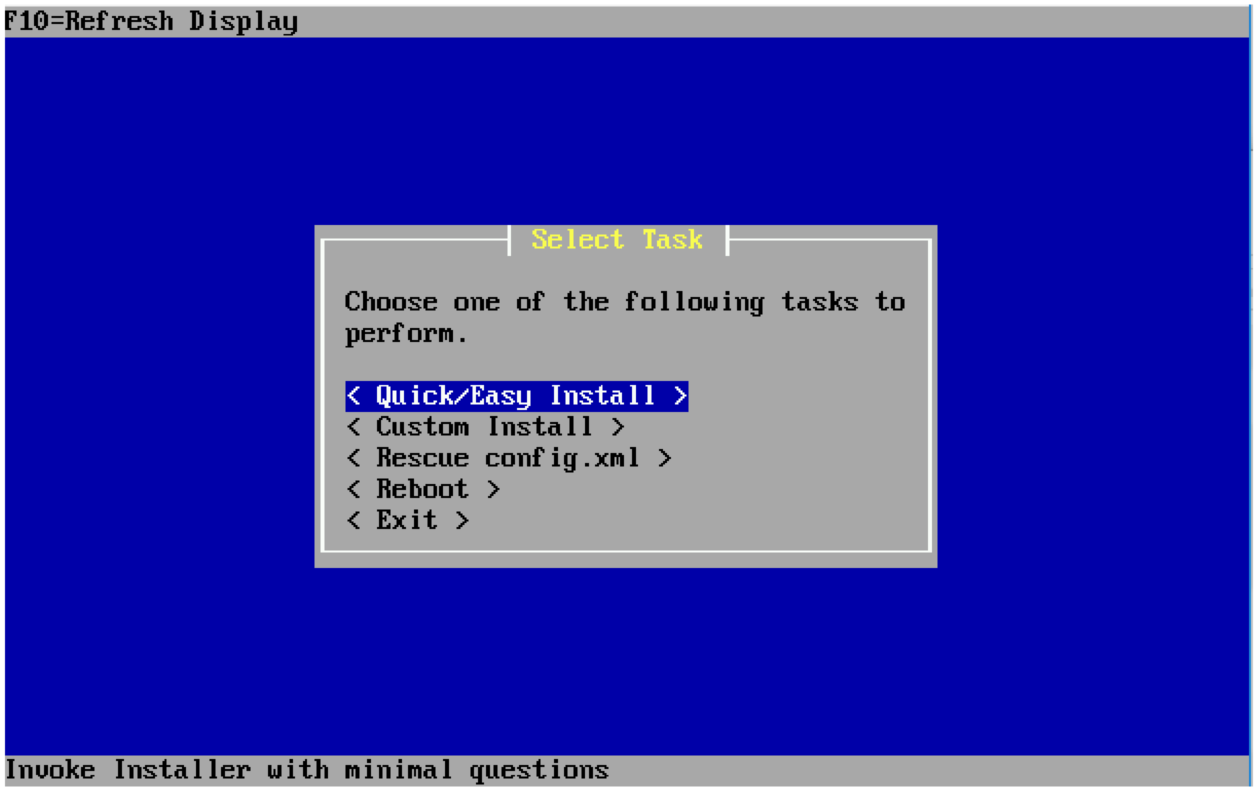
```
Welcome to pfSense

  1. Boot Multi User [Enter]
  2. Boot [S]ingle User
  3. [Esc]ape to loader prompt
  4. Reboot

  Options:
  5. [K]ernel: kernel (1 of 2)
  6. Configure Boot [O]ptions...
```

Let pfSense boot up, and the system should automatically run the installer. Adjust your video, screenmap, and keymap settings as necessary, then select `< Accept these Settings >`.



```
F10=Refresh Display

Configure Console

Your selected environment uses the
following console settings, shown in
parentheses. Select any that you wish
to change.

< Change Video Font (default) >
< Change Screenmap (default) >
< Change Keymap (default) >
< Accept these Settings >
```

On the next screen, select `< Quick/Easy Install >` and let pfSense do all the heavy lifting. The next screen will inform you that the install will erase the contents of the hard disk. Since our virtual disk is already empty, this doesn't matter in the least. Select OK, and let the installer run.

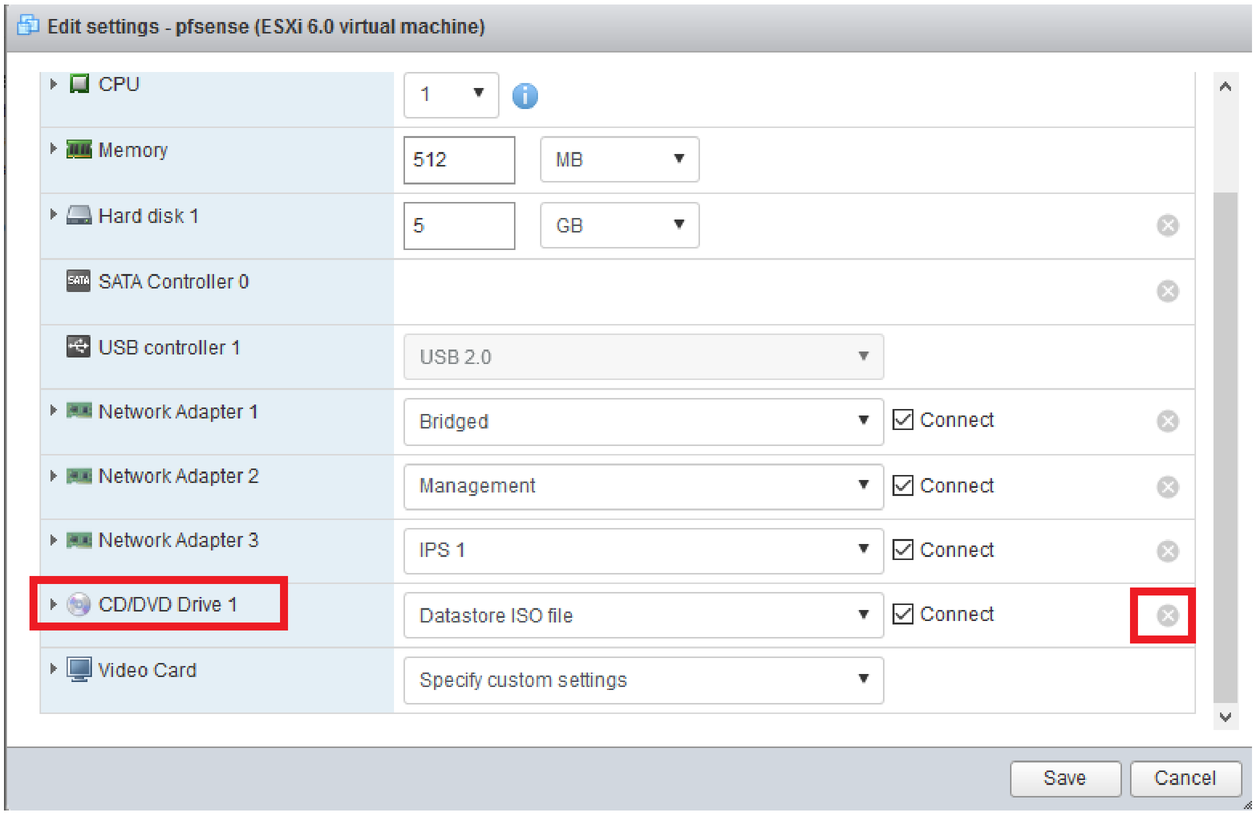


```
F10=Refresh Display

                 Select Task

     Choose one of the following tasks to
     perform.

     < Quick/Easy Install >
     < Custom Install >
     < Rescue config.xml >
     < Reboot >
     < Exit >

Invoke Installer with minimal questions
```

The installer will go through and install pfSense to the virtual hard disk. At a certain point you will be asked whether the standard kernel or the embedded one should be used. Make sure to select `< Standard Kernel >`. Finally, the installer informs you to reboot the machine to boot from the hard drive. Allow the virtual machine to reboot in order to finish the installation procedure. After system has rebooted, in the ESXi web interface, close the console, right click on pfSense, and select *Power* > *Power off* to turn off the VM.

Final VM Settings

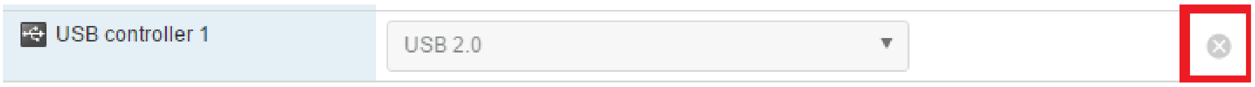
In the *Virtual Machines* pane under Navigator, right click on pfSense and select *Edit settings* from the menu. In the section labeled *CD/DVD drive*, click the X to the right to remove the CD/DVD drive; we won't need it anymore.



Edit settings - pfsense (ESXi 6.0 virtual machine)

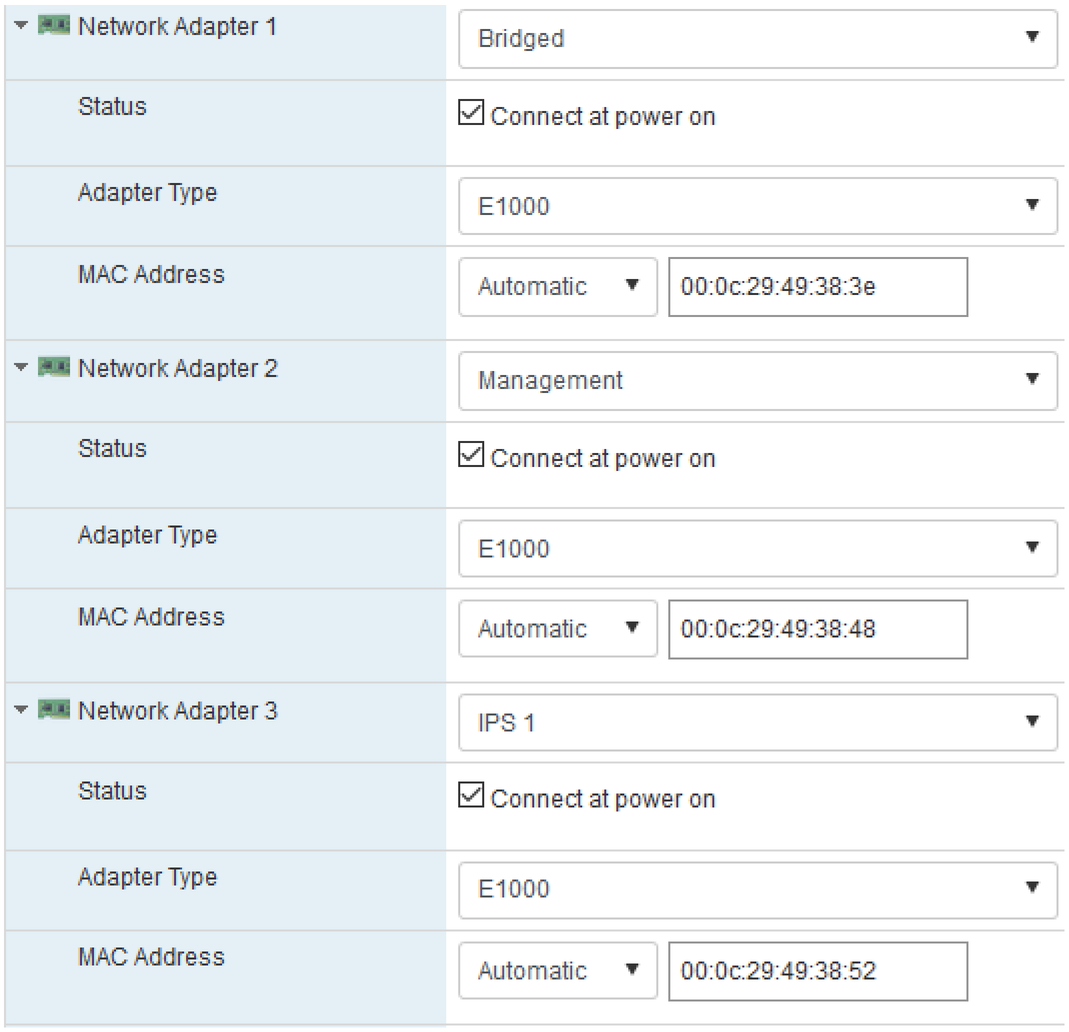
CPU	1	
Memory	512 MB	
Hard disk 1	5 GB	
SATA Controller 0		
USB controller 1	USB 2.0	
Network Adapter 1	Bridged	☑ Connect
Network Adapter 2	Management	☑ Connect
Network Adapter 3	IPS 1	☑ Connect
CD/DVD Drive 1	Datastore ISO file	☑ Connect
Video Card	Specify custom settings	

Follow the instructions as described above for the CD/DVD drive to remove *USB controller 1* as well:



USB controller 1 — USB 2.0

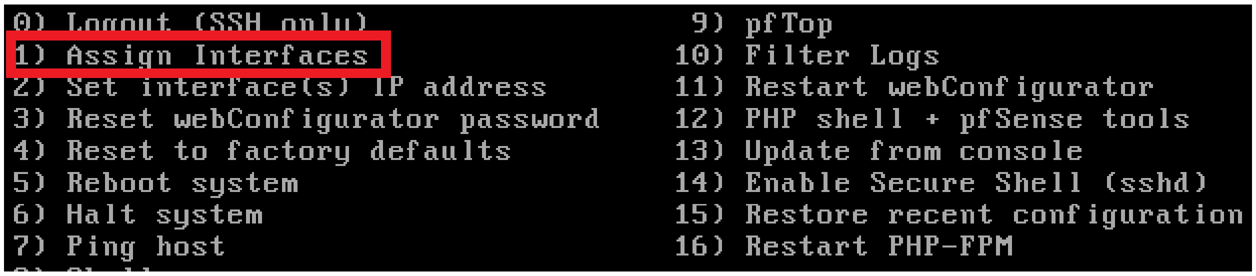
Afterwards, click the triangle next to *Network Adapter 1*, *Network Adapter 2*, and *Network Adapter 3* to show the additional information available for these devices. Be sure to document the contents of the *MAC address* field, and which MAC address is connected to which port group. We will need this information in the next section to perform the network configuration portion of the pfSense installation. When finished, click the *Save* button to close the window.



Network Adapter 1	Bridged
Status	✓ Connect at power on
Adapter Type	E1000
MAC Address	Automatic 00:0c:29:49:38:3e
Network Adapter 2	Management
Status	✓ Connect at power on
Adapter Type	E1000
MAC Address	Automatic 00:0c:29:49:38:48
Network Adapter 3	IPS 1
Status	✓ Connect at power on
Adapter Type	E1000
MAC Address	Automatic 00:0c:29:49:38:52

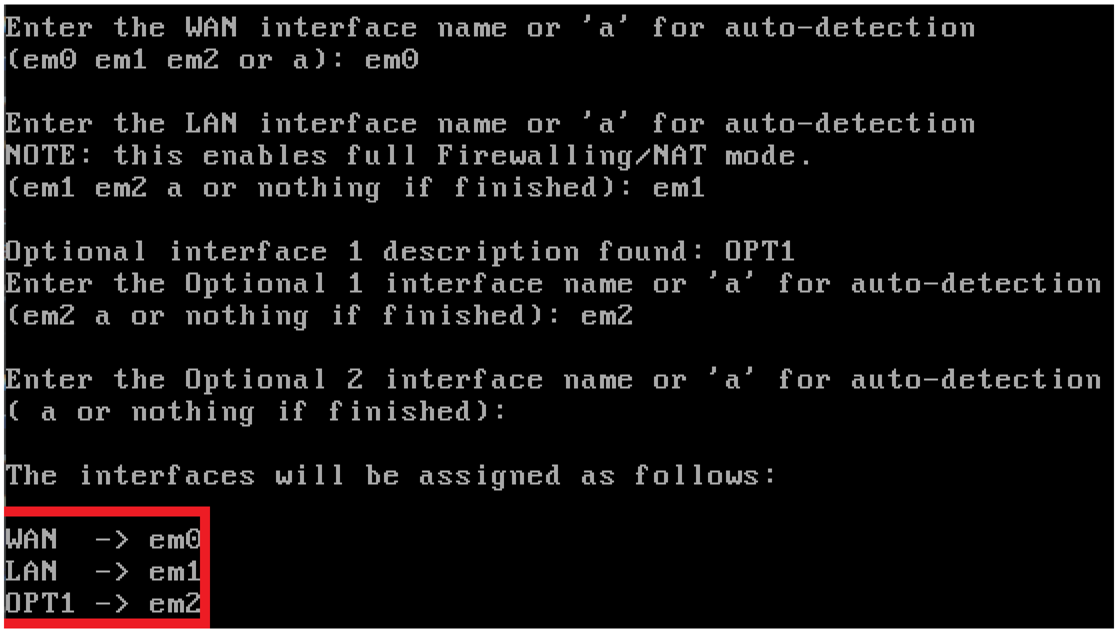
Network Configuration

Power on the pfSense VM, and allow it to boot up. Eventually you'll be greeted with the main pfSense menu on the console with 16 options. Select option 1, *Assign Interfaces* to run the interface assignment wizard. To manage its network services and firewall policies, pfSense uses internal network interface descriptors (*WAN, LAN, OPT1*, etc.). Running the wizard and assigning the interfaces defines which of them *(em0, em1, em2*, etc.) corresponds to which of pfSense's internal descriptors.



```
0) Logout (SSH only)                  9) pfTop
1) Assign Interfaces                 10) Filter Logs
2) Set interface(s) IP address       11) Restart webConfigurator
3) Reset webConfigurator password    12) PHP shell + pfSense tools
4) Reset to factory defaults         13) Update from console
5) Reboot system                     14) Enable Secure Shell (sshd)
6) Halt system                       15) Restore recent configuration
7) Ping host                         16) Restart PHP-FPM
```

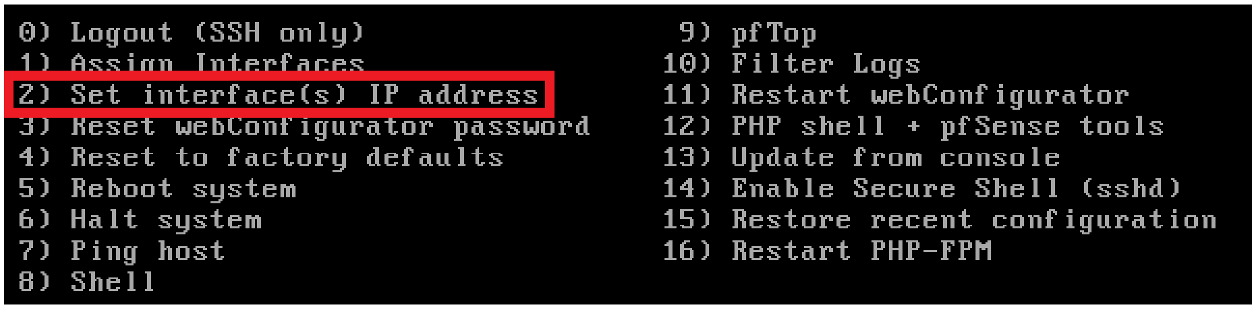
In the "Final VM Settings" section, we documented the MAC address of each network adapter and the respective ESXi port group it is connected to. With this information at hand, it should be easy to identify each of them on the list of valid interfaces the wizard displays on startup. After figuring out which pfSense interface corresponds to which ESXi network adapter(and port group), we need to assign em0, em1 and em2 to the WAN, LAN, and OPT1 descriptor accordingly. The WAN interface needs to be assigned to the network adapter connected to the Bridged port group, The LAN interface needs to be assigned to the network adapter connected to the Management port group, and finally, OPT1 needs to be assigned to the network adapter connected to the IPS 1 port group. In my case, em0 became the WAN interface, em1 became the LAN interface, and em2 became the OPT1 interface. After confirming that the settings are correct, you should automatically be returned to the pfSense main menu.



```
Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): em0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): em1

Optional interface 1 description found: OPT1
Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished): em2

Enter the Optional 2 interface name or 'a' for auto-detection
( a or nothing if finished):

The interfaces will be assigned as follows:

WAN  -> em0
LAN  -> em1
OPT1 -> em2
```

Now that we have assigned the network interfaces, we have to configure their IP addresses. **I very highly recommend configuring a static IP address for the pfSense WAN interface (using the Set interface(s) IP address menu option and configuring an IP address, subnet mask and gateway appropriate to your physical network) or creating a static DHCP mapping for the pfSense WAN interface on the device providing DHCP services to your physical network.** We will be using the pfSense VM's WAN IP address to route traffic from our management workstation (or "jumpbox") to the lab network. In order to do this, the WAN interface's IP address cannot change.

As for the LAN and OPT1 networks, we will have to manually set the IP address, subnet mask, and DHCP scopes for the networks. Select option 2 in the pfSense main menu to get started.

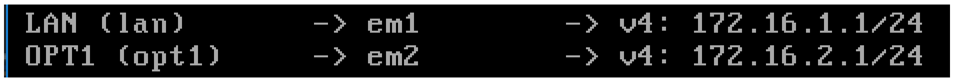


```
0) Logout (SSH only)                  9) pfTop
1) Assign Interfaces                 10) Filter Logs
2) Set interface(s) IP address       11) Restart webConfigurator
3) Reset webConfigurator password    12) PHP shell + pfSense tools
4) Reset to factory defaults         13) Update from console
5) Reboot system                     14) Enable Secure Shell (sshd)
6) Halt system                       15) Restore recent configuration
7) Ping host                         16) Restart PHP-FPM
8) Shell
```

The configuration wizard starts by asking for the interface the settings should be changed for. We have to go through the process two times, for the LAN and the OPT1 interface individually. Here are the settings I recommend:

LAN IP: **172.16.1.1**
LAN Subnet bit count: **24**
LAN upstream gateway address: **<empty>**
LAN IPv6 address: **<empty>**
Do you want to enable the DHCP server on LAN? **y**
LAN DHCP start address: **172.16.1.10**
LAN DHCP end address: **172.16.1.254**
Do you want to revert to HTTP as the webConfigurator protocol? **N**

OPT1 IP: **172.16.2.1**
OPT1 Subnet bit count: **24**
OPT1 upstream gateway address: **<empty>**
OPT1 IPv6 address: **<empty>**
Do you want to enable the DHCP server on OPT1? **y**
OPT1 DHCP start address: **172.16.2.10**
OPT1 DHCP end address: **172.16.2.254**
Do you want to revert to HTTP as the webConfigurator protocol? **n**



```
LAN (lan)       -> em1        -> v4: 172.16.1.1/24
OPT1 (opt1)     -> em2        -> v4: 172.16.2.1/24
```

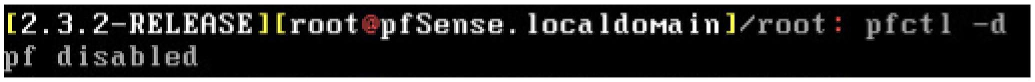
The reason we choose 1.10 and 2.10 as the DHCP start addresses is to reserve a few IP addresses for static DHCP allocations. Static DHCP allocations allow you to configure a DHCP server to always serve the same IP address when requested from a particular MAC address. We'll talk about this a little bit more later.

Note: If you are using 172.16.1.0/24 or 172.16.2.0/24 in your physical network, choose another network range to assign to the LAN and OPT1 interfaces to avoid network conflicts.

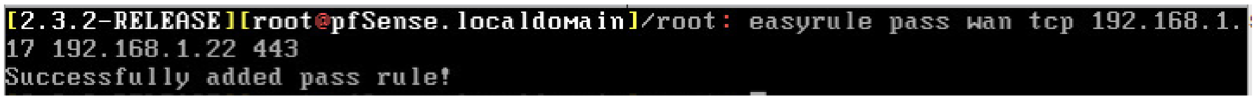
Note: When you are configuring the LAN and OPT1 interface IP addresses, the configuration script will ask "Do you want to revert to HTTP as the webConfigurator protocol?" **Always say no to this.** This allows the web UI to accept HTTP logins, and makes your firewall credentials vulnerable to sniffing over the network.

Before we're done using the command line, there is one more task we need to perform to access the pfSense web interface. Select option 8 from the pfSense menu to open up a shell. Once you get a shell prompt, run the following commands:

```
pfctl -d
easyrule pass wan tcp 192.168.1.17 192.168.1.22 443
exit
```



```
[2.3.2-RELEASE][root@pfSense.localdomain]/root: pfctl -d
pf disabled
```



```
[2.3.2-RELEASE][root@pfSense.localdomain]/root: easyrule pass wan tcp 192.168.1.
17 192.168.1.22 443
Successfully added pass rule!
```

The first command, `pfctl -d`, temporarily disables the firewall on pfSense. This is necessary because by default, the firewall drops ALL connections from RFC1918 addresses on the WAN interface. More than likely, your pfSense vm's WAN interface, and management workstation are going to be connected to a private network using RFC1918 addresses. For example, my home network uses the 192.168.1.0/24 private address range. pfSense assumes that the WAN interface will be exposed to the internet, so by default all RFC1918 addresses attempting to connect via the WAN interface are dropped (RFC1918 address ranges are NOT routable on the public internet). The management workstation you use will have to communicate with the pfSense VM through its WAN interface. Both of these network devices are on an RFC1918 network. Can you see where the problem is now?

The first command we entered, `pfctl -d`, disables the firewall entirely. This is a temporary solution to allow us to connect to the firewall, through explicitly disabling the setting that blocks ALL RFC1918 addresses on the WAN interface.

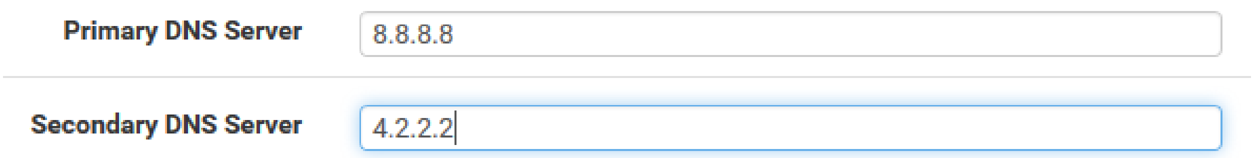
The second command, `easyrule pass wan tcp 192.168.1.17 192.168.1.22 443`, adds a firewall rule to the WAN interface of pfSense that allows 192.168.1.17 (the IP address of my management workstation) to access 192.168.1.22 (the IP address of WAN interface of the pfSense VM) on port 443 (https). Adjust the IP addresses of the management workstation, and the pfSense WAN interface in this rule as required for your network at work or at home. After you are done, type exit to leave the shell.

Note: You will need to run both of the commands above before moving on to the next section.

webConfigurator - Initial Setup

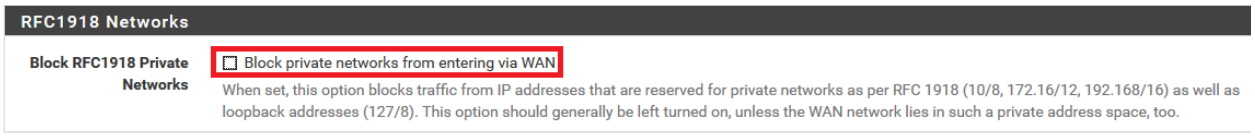
On your management workstation, open a web browser and navigate to the WAN address of the pfSense firewall (in my case, this is https://192.168.1.22). **The default credentials for access to the web interface are admin/pfsense.** If this is your first time logging in, pfSense takes you through a nice little setup wizard. I'll highlight some of the important things to take note of, and/or change as necessary:

Set the primary and secondary DNS servers you plan on using. I typically use 8.8.8.8 (Google public DNS) and 4.2.2.2 (Level 3 public DNS).



Primary DNS Server	8.8.8.8
Secondary DNS Server	4.2.2.2

Be sure to uncheck the *Block private networks from entering via WAN* rule. **This setting must be unchecked in order for your management system to be able to access the pfSense web interface** (For a more detailed explanation, see the paragraphs at the end of the previous section, "Network Configuration"). You may also elect to uncheck the *Block bogon networks* checkbox (I typically uncheck this option since pfSense blocks traffic that isn't explicitly allowed with a firewall rule).



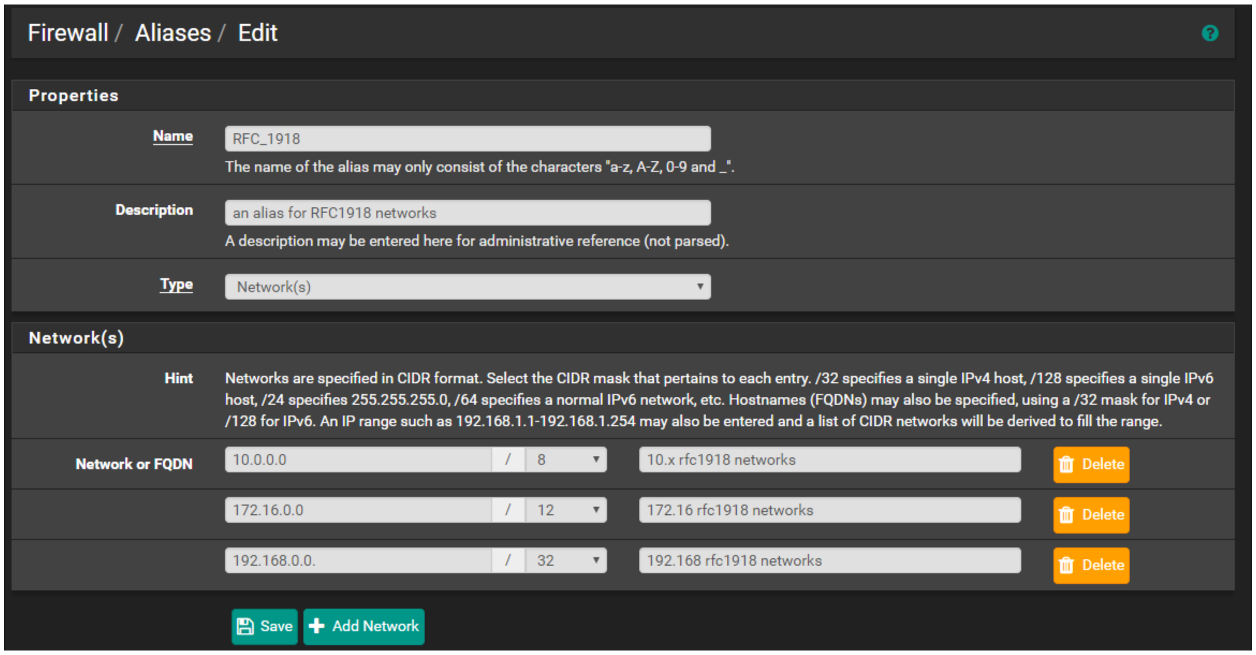
RFC1918 Networks

Block RFC1918 Private Networks — ☐ Block private networks from entering via WAN

When set, this option blocks traffic from IP addresses that are reserved for private networks as per RFC 1918 (10/8, 172.16/12, 192.168/16) as well as loopback addresses (127/8). This option should generally be left turned on, unless the WAN network lies in such a private address space, too.

The other default settings on the first time setup wizard should be fine. Please note that you will be asked to change the password for the admin account as a part of the setup wizard. Document the password and keep it somewhere safe. Use a password manager if necessary.

Next, on the menu bar at the top of the page, navigate to *Firewall* > *Aliases*. On the Firewall Aliases page, click on *IP*, then click *Add*. Create an alias with the following settings:


Firewall / Aliases / Edit

Properties

Name	RFC_1918 The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and _".
Description	an alias for RFC1918 networks A description may be entered here for administrative reference (not parsed).
Type	Network(s)

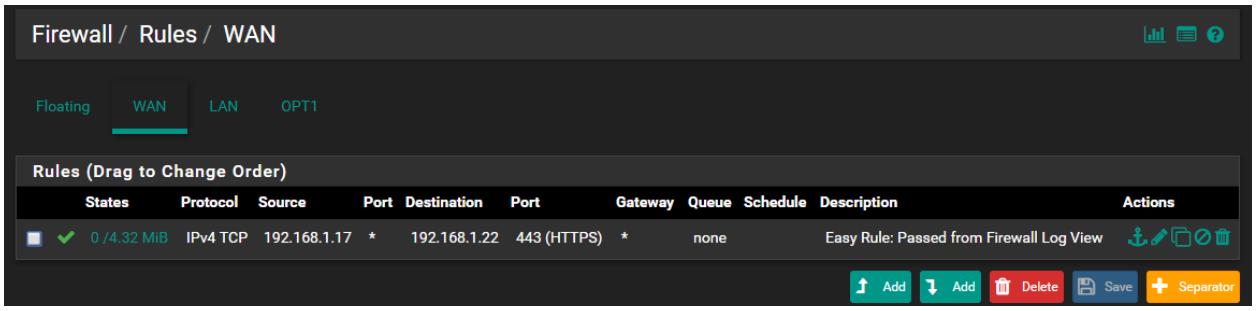
Network(s)

Hint: Networks are specified in CIDR format. Select the CIDR mask that pertains to each entry. /32 specifies a single IPv4 host, /128 specifies a single IPv6 host, /24 specifies 255.255.255.0, /64 specifies a normal IPv6 network, etc. Hostnames (FQDNs) may also be specified, using a /32 mask for IPv4 or /128 for IPv6. An IP range such as 192.168.1.1-192.168.1.254 may also be entered and a list of CIDR networks will be derived to fill the range.

Network or FQDN	10.0.0.0	/ 8	10.x rfc1918 networks	Delete
	172.16.0.0	/ 12	172.16 rfc1918 networks	Delete
	192.168.0.0.	/ 32	192.168 rfc1918 networks	Delete

Click Save to be brought back to the previous page, then click Apply Changes. We just created an alias for RFC1918 networks (local networks that are not routable through the public internet) This will come in handy later for creating firewall rules around these networks.

Next, navigate to Firewall > Rules, and click on WAN, if it isn't already selected. Your firewall rules for the WAN interface should look like this:


Firewall / Rules / WAN

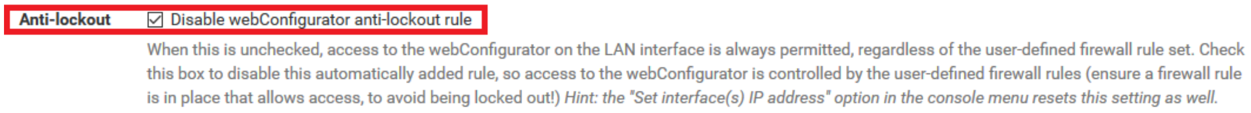
Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
✓	0 /4.32 MiB	IPv4 TCP	192.168.1.17	*	192.168.1.22	443 (HTTPS)	*	none		Easy Rule: Passed from Firewall Log View	

That lone firewall rule? <u>Never remove it.</u> It is the one created by using the `easyrule` command, as described at the end of the previous section, "Network Configuration". **This rule is absolutely mandatory.** Make sure to never create and place any other firewall rule for the WAN interface above this one (besides the *Block bogon networks* rule, if you left it enabled).

This rule is the only thing keeping you from being locked out of the firewall web interface. Once you are more familiar with the firewall rules UI, you may choose to edit this rule and change the description under the *Extra Options* section of this rule.

Next, navigate to System > Advanced. Click to fill in the checkbox next to the option “Disable webConfigurator anti-lockout rule”, and click Save on the bottom of the page. This rule is meant to allow any machine connected to the LAN network to access the firewall’s web interface. Since we are going to manage pfSense by connecting to the WAN interface, and we just made sure that the necessary firewall rule is in place, disabling the anti-lockout rule is safe and reduces unnecessary exposure.



Anti-lockout ☑ Disable webConfigurator anti-lockout rule

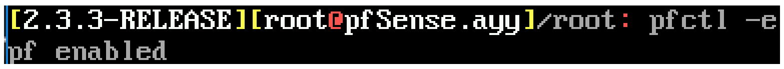
When this is unchecked, access to the webConfigurator on the LAN interface is always permitted, regardless of the user-defined firewall rule set. Check this box to disable this automatically added rule, so access to the webConfigurator is controlled by the user-defined firewall rules (ensure a firewall rule is in place that allows access, to avoid being locked out!) *Hint: the "Set interface(s) IP address" option in the console menu resets this setting as well.*

One final task before moving on to the next section, we are going to make sure that the firewall is re-enabled. If no longer available, open a new console session to the pfSense VM in the ESXi web UI, and select option 8 from the firewall’s main menu to open a shell session. Enter the following command to re-enable the firewall:

```
pfctl -e
```

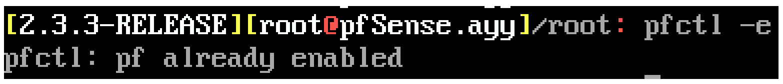
As a part of the initial webConfigurator setup, pfSense reloads the firewall config. While this SHOULD re-enable the firewall, running the command above ensures this has happened and that the firewall is operating normally.

In case the firewall still was disabled, the console output will look like the one in the picture below::



```
[2.3.3-RELEASE][root@pfSense.ayy]/root: pfctl -e
pf enabled
```

This is the console output that is returned if the firewall has already been re-enabled:



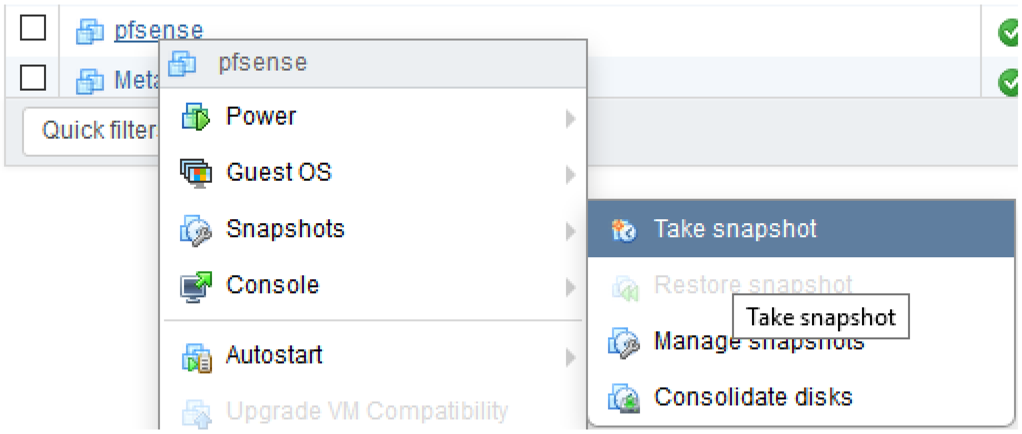
```
[2.3.3-RELEASE][root@pfSense.ayy]/root: pfctl -e
pfctl: pf already enabled
```

__**Make sure to verify that the firewall is re-enabled, and that you are still able to access the webConfigurator before moving on to the next section.**__

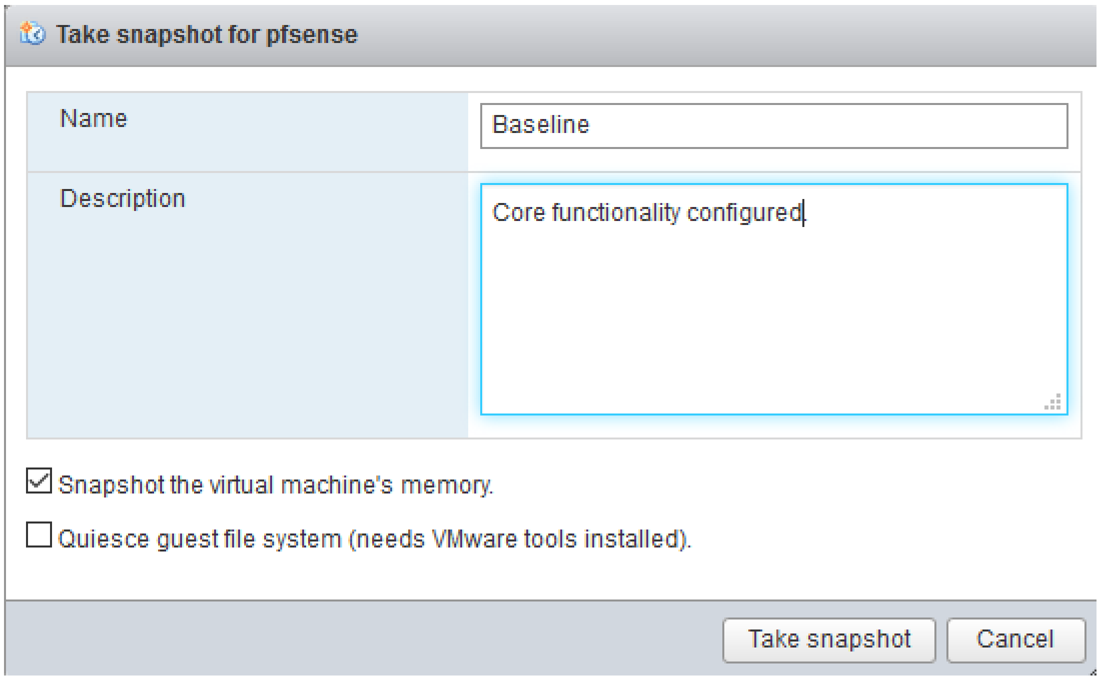
Take a Snapshot

Snapshotting is a VERY important concept with VMs. It lets you restore your virtual machines to its state in the past when the snapshot was taken. This allows you to undo misconfigurations, solve problems or potential issues that might affect the VM, in relatively easily fashion. We're going to take a snapshot of the pfSense virtual machine in its current state.

On the ESXi web interface, click on *Virtual Machines* under the Navigator pane. In the Virtual machine list, right click on pfSense, then select *Snapshots* > *Take Snapshot* from the menu.



A new window pops up, offering two input boxes: one to name the snapshot, and another to enter a description. I named the snapshot “Baseline” and gave a brief description of what the snapshot is for. If you keep multiple snapshots, make sure to use good descriptions that include WHEN the snapshot was taken, and WHAT STATE the VM is in when you took the snapshot. After you are done, click the Take Snapshot button. The ESXi server will begin generating the snapshot for your virtual machine immediately. You can monitor the progress in the Recent tasks pane at the bottom of the ESXi web UI window.

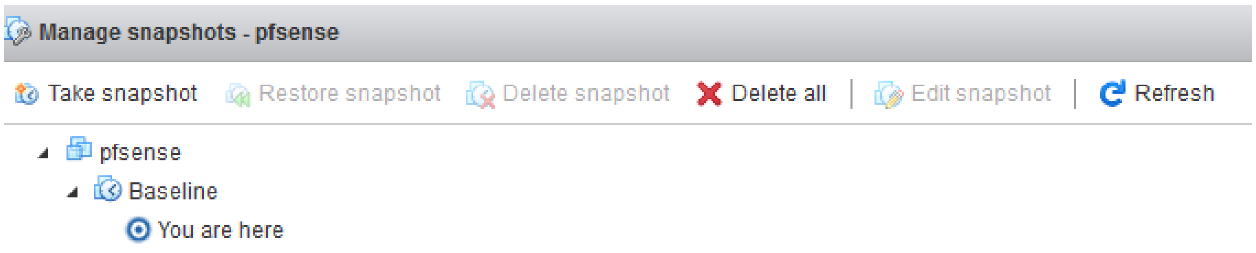


Name	Baseline
Description	Core functionality configured

☑ Snapshot the virtual machine's memory.

☐ Quiesce guest file system (needs VMware tools installed).

To access previous snapshots, right click on the VM, and choose *Snapshots* > *Manage snapshots*. This opens a new window, displaying a tree view of snapshots available, and offers options for restoring from previous snapshots, deleting snapshots, etc.



pfSense Summary

At the end of all this, you should have a pfSense VM with the following settings:

- 512MB of RAM
- 5GB of disk
- 1 Virtual CPU
- 3 Network interfaces:
    - 1 Interface connected to the Bridged port group (WAN interface)
    - 1 Interface connected to the Management port group (LAN interface)
    - 1 Interface connected to the IPS1 port group (OPT1 interface)

pfSense should be configured as followed:

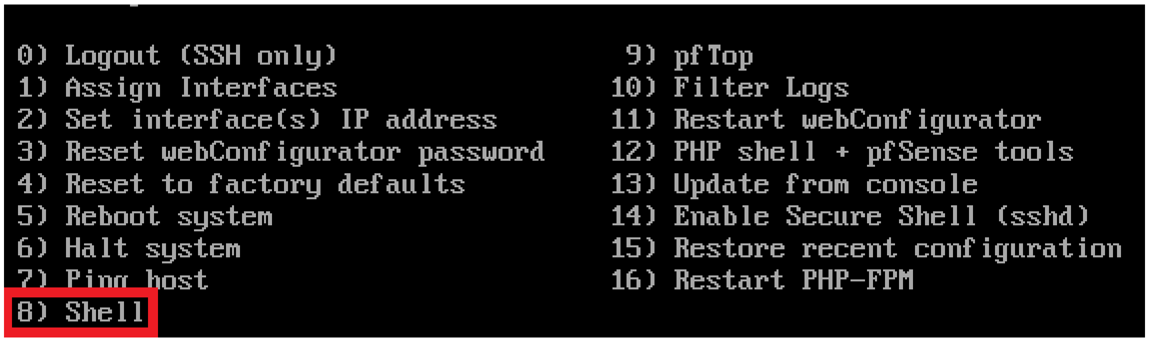
- The WAN interface should be the ESXi network adapter connected to the Bridged port group, meaning it should have an IP address on your physical network. **Configure a static IP address or a static DHCP mapping for this interface to ensure that the address does not change. <u>This is extremely important.</u>**
- The LAN interface should have an IP address of 172.16.1.1
- The LAN network should be 172.16.1.0 with a /24 (255.255.255.0) subnet mask
- DHCP should be configured for this network with a scope of 172.16.1.10-172.16.1.254
    - 172.16.1.2-172.16.1.9 are available for setting static DHCP mappings
- The OPT1 interface should have an IP address of 172.16.2.1
- The OPT1 network should be 172.16.2.0 with a /24 (255.255.255.0) subnet mask
- DHCP should be configured for this network with a scope of 172.16.2.10-172.16.2.254
    - 172.16.2.2-172.16.2.9 are available for static DHCP mappings
- Your management workstation should ble capable of accessing the pfSense webConfigurator on the WAN interface over HTTPS. We created a firewall rule that serves as anti-lockout rule for the webConfigurator on this interface.
    - Be aware that if the IP address of the management workstation or of the WAN interface changes, this firewall rule will need to be adjusted via the ESXi web UI, or in the console for the pfSense VM, using the `easyrule` utility
- You should have an alias for RFC1918 networks configured for use with the firewall policy.
- You should have at least one good snapshot you can revert to if you need to redo a step or somehow got lost/confused.

What's Next?

Now that you have an operational pfSense VM, your next steps are to read the pfSense Firewall Rules and Network Services Guide. Make sure the firewall policies match those illustrated in the segmentation guide, and make sure that the pfSense VM is hosting all the services mentioned in the Core Network Services guide to make your firewall fully functional and ready to handle your lab network.

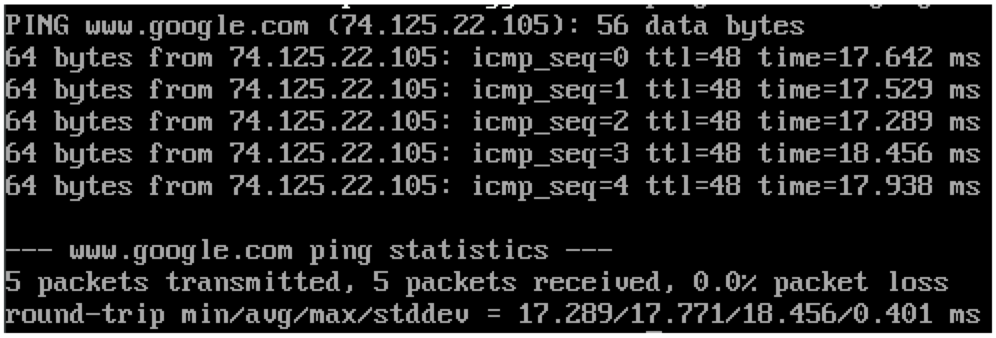
Final Connectivity Checks and Troubleshooting

Once you have finished setting up your pfSense VM according to the instructions laid out, Open a console session to the pfSense VM, and attach your mouse and keyboard by clicking on the console. If the screen is blank, hit the *Enter* key to bring up the pfSense main menu. Select option *8) Shell* to start a shell session on pfSense.



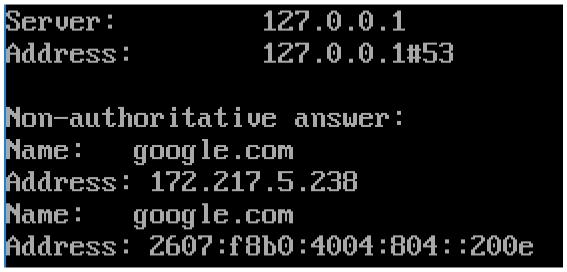
```
0) Logout (SSH only)                  9) pfTop
1) Assign Interfaces                 10) Filter Logs
2) Set interface(s) IP address       11) Restart webConfigurator
3) Reset webConfigurator password    12) PHP shell + pfSense tools
4) Reset to factory defaults         13) Update from console
5) Reboot system                     14) Enable Secure Shell (sshd)
6) Halt system                       15) Restore recent configuration
7) Ping host                         16) Restart PHP-FPM
8) Shell
```

The first, run the command `ping -c 5 www.google.com`



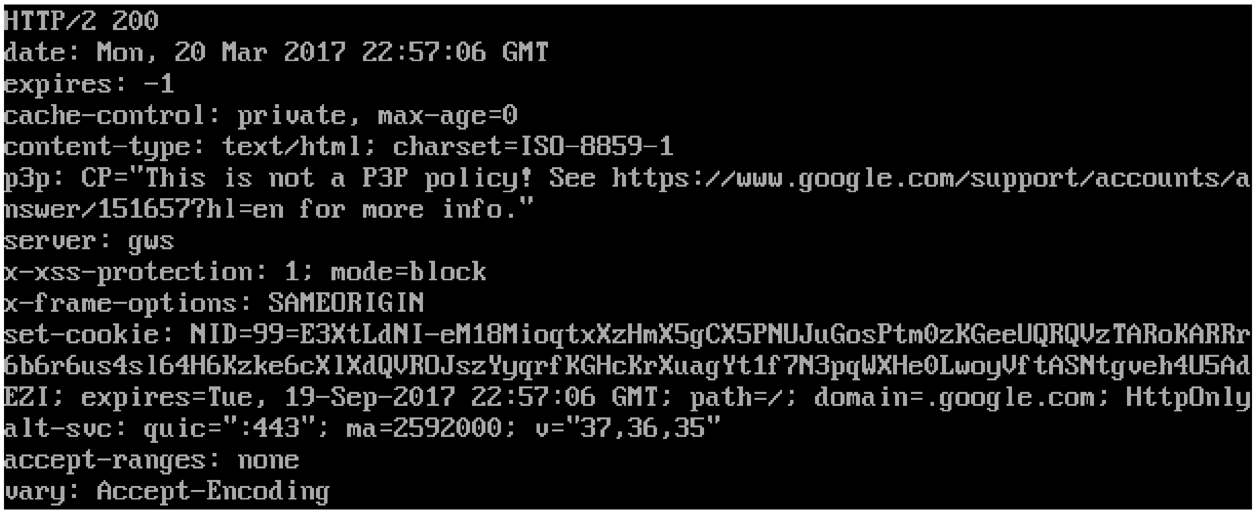
```
PING www.google.com (74.125.22.105): 56 data bytes
64 bytes from 74.125.22.105: icmp_seq=0 ttl=48 time=17.642 ms
64 bytes from 74.125.22.105: icmp_seq=1 ttl=48 time=17.529 ms
64 bytes from 74.125.22.105: icmp_seq=2 ttl=48 time=17.289 ms
64 bytes from 74.125.22.105: icmp_seq=3 ttl=48 time=18.456 ms
64 bytes from 74.125.22.105: icmp_seq=4 ttl=48 time=17.938 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 17.289/17.771/18.456/0.401 ms
```

The output from the `ping` command should look similar to the output above. Next, run the command `nslookup google.com`



```
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.5.238
Name:   google.com
Address: 2607:f8b0:4004:804::200e
```

Again, you should have output similar to the illustration above. Finally, run `curl -I https://www.google.com`



```
HTTP/2 200
date: Mon, 20 Mar 2017 22:57:06 GMT
expires: -1
cache-control: private, max-age=0
content-type: text/html; charset=ISO-8859-1
p3p: CP="This is not a P3P policy! See https://www.google.com/support/accounts/a
nswer/151657?hl=en for more info."
server: gws
x-xss-protection: 1; mode=block
x-frame-options: SAMEORIGIN
set-cookie: NID=99=E3XtLdNI-eM18MioqtxXzHmX5gCX5PNUJuGosPtm0zKGeeUQRQVzTARoKARRr
6b6r6us4sl64H6Kzke6cXlXdQVROJszYyqrfKGHcKrXuagYt1f7N3pqWXHe0LwoyVftASNtgveh4U5Ad
EZI; expires=Tue, 19-Sep-2017 22:57:06 GMT; path=/; domain=.google.com; HttpOnly
alt-svc: quic=":443"; ma=2592000; v="37,36,35"
accept-ranges: none
vary: Accept-Encoding
```

The output of your `curl` command should be similar to the screen capture above. If so, you may type `exit` to leave the shell, and close your connection to the pfSense console. If you had any problems with any of the commands above, you've got troubleshooting to do. When it comes to troubleshooting, start at layer 1 of the OSI model (Physical) and work your way up. Here are some questions to help the troubleshooting process:

- Which of the commands above failed? Did they all fail?
- Have you checked your physical cabling or wireless network connectivity?
- Does your company/LAN have MAC filtering/port security on? You might need to modify switch settings for your drop, or talk to your network administrator.
- Have you checked your upstream gateway/SOHO router? Trying pinging the gateway's IP address to see if your pfSense VM can reach the gateway. Try running the `traceroute` command to google.com to see if and where your ping command is failing.

- Do you have an upstream firewall in place? Is 443/TCP allowed outbound? Is 53/UDP allowed outbound? Is ICMP allowed?
- Did you input the commands correctly? Try entering them again. No, I'm not patronizing you.

Once you have verified that your firewall's connectivity, policies, and services are properly configured, **SNAPSHOT THE VM <u>BEFORE</u>** moving on and trying your hand at setting up the remaining VMs.

Your Turn

The pfSense VM required a lot of custom configuration, both in the ESXi web interface, as well as in the VM itself once the OS was installed. Now that we did that together, I'm going to have you create the rest of the virtual machines yourself -- with the exception of the Metasploitable 2 VM since it's a special case for reasons you will see momentarily. Below are a set of spec sheets to help you create the other virtual machines for the lab on your own. Think of them as checklists you should be able to run through unaccompanied.

All of our Linux-based virtual machines are built on Debian Linux based distros (both, Ubuntu and Kali Linux are derived from Debian), so setup and configuration of all our lab VMs should be very similar.

Kali Linux VM

Kali Linux is a popular penetration testing distro. Popular because hackers and script kiddies are lazy, and practically everything you need for performing a penetration test or joining anonymous is installed by default. Kali is going to be our loud, obnoxious attacker that we're going to use as a noise generator for the express purpose of generating events on our IPS VM. I want you to perform the following tasks:

- Download Kali Linux 64-bit here: https://www.kali.org/downloads/
- Upload your Kali Linux ISO file to your ESXi server's datastore using the Datastore browser
- Create a VM with the following settings:
    - Name the VM "Kali"
    - Set the Guest OS family to "Linux"
    - Set the Guest OS version to "Debian GNU/Linux 8 (64-bit)"
    - Allocate 1 or 2 CPUs
    - Allocate 4GB of RAM
    - Connect the VM's network adapter to the "IPS 1" port group
    - Allocate 80 GB of space for Hard disk 1
        - Configure the Disk Provisioning to Thick provisioned, lazily zeroed
        - Configure the Virtual Device Node to Sata Controller 0 (SATA (0:0))
    - Remove the SCSI controller
    - Add a CD/DVD drive configured to use the Kali Linux ISO file you uploaded to the datastore
        - Ensure that the checkbox next to "Connect at power on" is checked
    - Adjust the Video Card setting "Total Video Memory" to at least 32MB

        - This is required in order for the VM to render the GUI
- Install Kali Linux
    - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.2.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy
- After the install is completed, power off the VM and adjust the following settings:
    - Remove the CD/DVD drive
    - Remove the USB controller
    - Under Network adapter 1, make sure to document the MAC address of this VM
- Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the OPT1 interface DHCP for the MAC address of adapter 1 for the Kali Linux VM; assign it the IP address 172.16.2.2, making sure to enter a description that tells you which VM this static mapping belongs to
- Power on the virtual machine and do the following:
    - Verify that the static mapping for the VM's IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense
        - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
    - Verify the virtual machine can reach the internet
        - In a terminal/shell run the command `curl -I https://www.google.com`
            - This will confirm DNS can resolve domain names, and that the VM can reach the internet
            - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense.
                - **Note:** The `ping` command, in this case, will work from this virtual machine, due to the unique firewall policy on the OPT1 network
    - Update/patch the VM
        - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
            - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
    - Create a snapshot for the VM

SIEM VM

SIEM is shorthand for Security Intrusion Events Manager. This is fancy security nomenclature for “log aggregator”; we will be having our IPS VM log its events here. We’re going to be running Splunk on this VM. Splunk is a commercial program for managing logs on a pretty large scale. By default, and with no licensing, Splunk only allows you to collect or “index” 500MB worth of logs per day, however, there are /ways/ around this that we’ll talk about later. I want you perform the following tasks to set up the SIEM VM:

- Download the latest Ubuntu Server 64-bit LTS release. In our case, this is Ubuntu Server 16.04.1, which can be found here: http://www.ubuntu.com/download/server
- Upload the Ubuntu Server ISO file to your ESXi server’s datastore
- Create a VM with the following settings:
    - Name the VM “SIEM”
    - Set the Guest OS family to “Linux”
    - Set the Guest OS version to “Ubuntu Linux (64-bit)”
    - Allocate 1 or 2 CPUs
    - Allocate 4GB of RAM
    - Connect the VM’s network adapter to the “Management” port group
    - Allocate 80 GB of space for Hard disk 1
        - Configure the Disk Provisioning to Thick provisioned, lazily zeroed
        - Configure the Virtual Device Node to Sata Controller 0 (SATA (0:0))
    - Remove the SCSI controller
    - Add a CD/DVD drive configured to use the Ubuntu Linux ISO file you uploaded to the datastore.
        - Ensure that the checkbox next to “Connect at power on” is checked
- Install Ubuntu Server
    - Be sure to install the Standard System Utilities and OpenSSH Server role when asked. You should not need to install any other roles or features for this server
    - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.1.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy
- After the install is completed, power off the VM and adjust the following settings:
    - Remove the CD/DVD drive
    - Remove the USB controller
    - Under Network adapter 1, make sure to document the MAC address of this VM
- Log in to the pfSense web UI, navigate to Services > DHCP Server, and add a static mapping to the LAN interface DHCP for the MAC address of adapter 1 for the SIEM VM; assign it the IP address 172.16.1.3, making sure to enter a description that tells you which VM this static mapping belongs to
- Power on the virtual machine and do the following:
    - Verify that the static mapping for the VM’s IP address worked; open a

terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter's name is in the VM) was given the IP address you configured its static mapping for in pfSense
        - If the virtual machine does not have an IP address, run the `dhclient` command to have the VM request an IP address from the DHCP server
    - Verify the virtual machine can reach the internet
        - In a terminal/shell run the command `curl -I https://www.google.com`
            - This will confirm DNS can resolve domain names, and that the VM can reach the internet
            - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense.
                - **Note:** The `ping` command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail.
    - Update/patch the VM
        - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
            - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
    - Create a snapshot for the VM

IPS VM

This VM is going to be responsible for running the AFPACKET bridge between the IPS 1 and IPS 2 virtual switches. Perform the following tasks to install the IPS VM:

- If you installed the SIEM VM already, you should already have Ubuntu Server downloaded. If not, download the latest Ubuntu Server 64-bit LTS release. In our case, this is Ubuntu Server 16.04.1, which can be found here: http://www.ubuntu.com/download/server
- Upload the Ubuntu Server ISO file to your ESXi server's datastore (if it isn't already there)
- Create a VM with the following settings:
    - Name the VM "ips"
    - Set the Guest OS family to "Linux"
    - Set the Guest OS version to "Ubuntu Linux (64-bit)"
    - Allocate 1 CPU
    - Allocate 2GB of RAM
    - Create two additional network adapters (in addition to the one ESXi provides you by default)
        - Connect Network adapter 1 to the "Management" port group
        - Connect Network adapter 2 to the "IPS 1" port group

            - Ensure that the *Connect* checkbox is unchecked
        - Connect Network adapter 3 to the “IPS 2” port group
            - Ensure that the *Connect* checkbox is unchecked
        - Verify that that the Promiscuous mode and the Forged transmit options are both set to Accept on the IPS 1 and the IPS 2 virtual switches. This is very important
    - Allocate 80 GB of space for Hard disk 1
        - Configure the Disk Provisioning to Thick provisioned, lazily zeroed
        - Configure the Virtual Device Node to Sata Controller 0 (SATA (0:0))
    - Remove the SCSI controller
    - Add a CD/DVD drive configured to use the Ubuntu Linux iso file you uploaded to the datastore.
        - Ensure that the checkbox next to “Connect at power on” is checked
- Install Ubuntu Server
    - When prompted, select the option to install security updates automatically
    - Be sure to install the standard system utilities and OpenSSH Server software packages when asked. You should not need to install any other roles or features for this server.
    - **Note:** If you configured the firewall on the pfSense VM, and decided to install and enable Squid proxy services, during the apt package retrieval portion of the installer, you will be ask if your network uses a proxy server, and asked to enter the address of the proxy server as well as a username and password (if necessary). You will need to enter http://172.16.1.1:3128 as the proxy server address. There is no username or password required to use the Squid proxy
- After the install is completed, power off the VM and adjust the following settings:
    - Remove the CD/DVD drive
    - Remove the USB controller
    - Under Network adapter 1, 2, and 3, make sure to document the MAC addresses and the corresponding network(s) they are connected to.
        - Ensure that the *Connect* checkbox for Network adapter 2 and 3 (the adapters connected to the *IPS 1* and *IPS 2* networks, respectively) are checked again
- Log in to the pfSense web UI, navigate to *Services > DHCP Server*, and add a static mapping to the LAN interface DHCP for the MAC address of the network adapter connected to the “Management” network for the ips VM; assign it the IP address 172.16.1.4, making sure to enter a description that tells you which VM this static mapping belongs to
- Power on the virtual machine and do the following:
    - Verify that the static mapping for the VM’s IP address worked; open a terminal/shell and run `ifconfig -a | less` and verify that `eth0` (or whatever the network adapter’s name is in the VM) was given the IP address you configured its static mapping for in pfSense
        - If the virtual machine does not have an IP address, run the `dhclient`

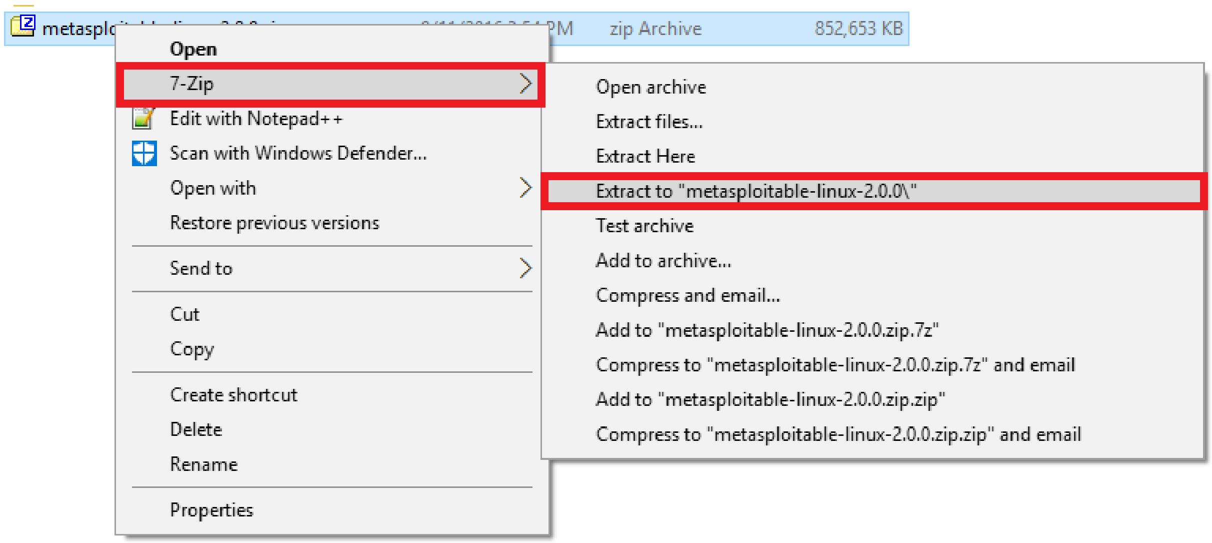
        command to have the VM request an IP address from the DHCP server

- Verify the virtual machine can reach the internet
    - In a terminal/shell run the command `curl -I https://www.google.com`
        - This will confirm DNS can resolve domain names, and that the VM can reach the internet
        - If the command appears to hang or does not return output, use the "Final Connectivity Checks and Troubleshooting" guide we used for testing connectivity with pfSense.
            - **Note:** The `ping` command will NOT work from any of your virtual machines, because we do not allow ICMP on the firewall. This means that attempts to ping the firewall or external websites will fail.
- Update/patch the VM
    - In a terminal/shell, run the following, and depending on your network connection, etc., be prepared to wait:
        - `export DEBIAN_FRONTEND=noninteractive; apt-get -q update; apt-get -y -q dist-upgrade`
- Create a snapshot for the VM

Metasploitable 2

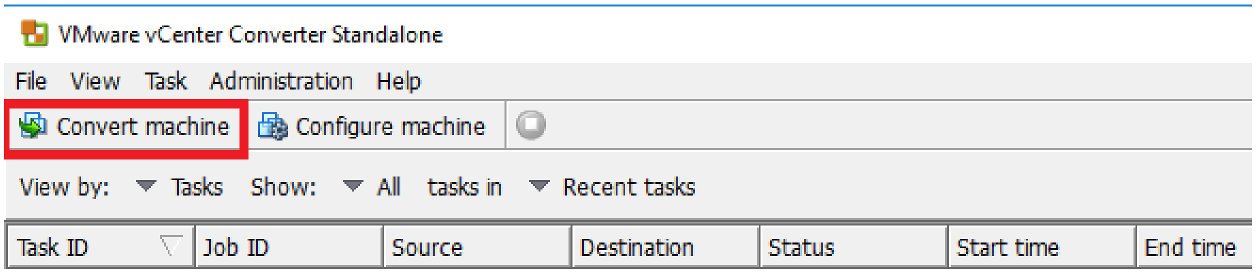
Setting up Metasploitable 2 on ESXi is slightly more complex than just creating a virtual machine. While there are multiple ways to get the VM onto the ESXi server, the easiest and fastest way to achieve this is through VMware's vCenter Converter Standalone. To download and use this tool, you need a Windows system, and an account on VMware's website, which you should already have from downloading ESXi earlier. Go download the installer here: https://my.VMware.com/group/VMware/evalcenter?p=converter.

After setting up the vCenter Converter, go and download Metasploitable 2 from https://sourceforge.net/projects/metasploitable/files/Metasploitable2/metasploitable-linux-2.0.0.zip/download. Metasploitable 2 is distributed as a zip archive, so you need a compression utility to extract it in order for the VMware tool to use its content. As we are using a Windows system, I recommend 7-Zip (http://www.7-zip.org/download.html). Once downloaded, locate the file using Windows Explorer (usually in the “Downloads” directory for the current user). Once located, right click on the zip file and select *7-Zip > Extract to metasploitable-linux-2.0.0\.*

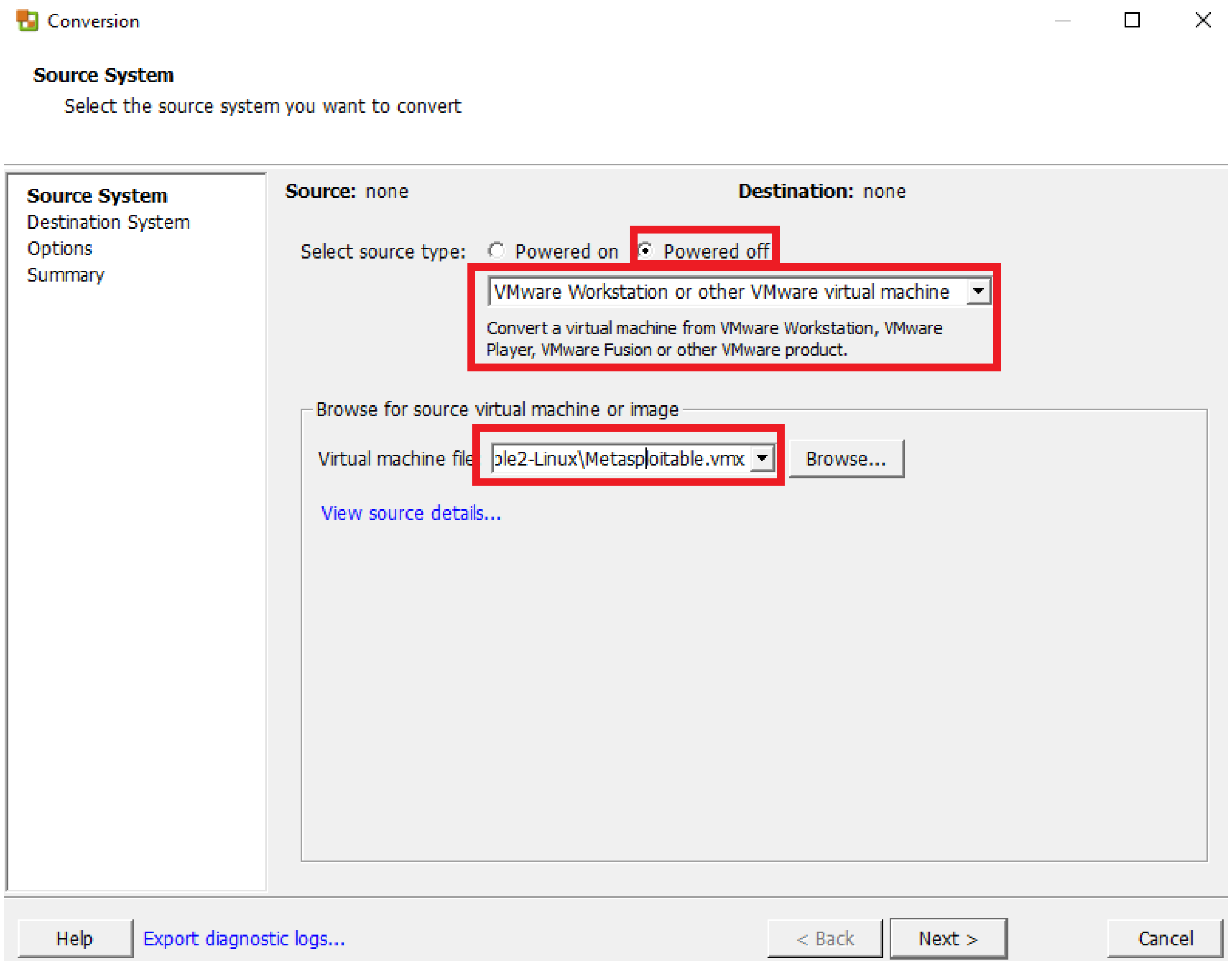


metasplo
PM
zip Archive
852,653 KB
Open
7-Zip
Edit with Notepad++
Scan with Windows Defender...
Open with
Restore previous versions
Send to
Cut
Copy
Create shortcut
Delete
Rename
Properties
Open archive
Extract files...
Extract Here
Extract to "metasploitable-linux-2.0.0\"
Test archive
Add to archive...
Compress and email...
Add to "metasploitable-linux-2.0.0.zip.7z"
Compress to "metasploitable-linux-2.0.0.zip.7z" and email
Add to "metasploitable-linux-2.0.0.zip.zip"
Compress to "metasploitable-linux-2.0.0.zip.zip" and email

After extracting the Metasploitable 2 archive, open VMware vCenter Converter Standalone and click “Convert machine”.



The Conversion wizard you just started guides you through a variety of steps. The first step is choosing the source system to convert. Click the *Powered off* radio button on the top of the central pane. From the drop-down list right below, select *VMware Workstation or other VMware virtual machine*. In the section underneath, entitled *Browse for source virtual machine or image*, click the *Browse* button. Navigate to the directory you extracted Metasploitable 2 in. Find the `Metasploitable.vmx` file, and select it. In my case, this was located in `D:\VMs\metasploitable-linux-2.0.0\Metasploitable2-Linux\Metasploitable.vmx`. After selecting the vmx file, click *Next*.



Source: none   Destination: none

Select source type: Powered on [✓ Powered off]

VMware Workstation or other VMware virtual machine

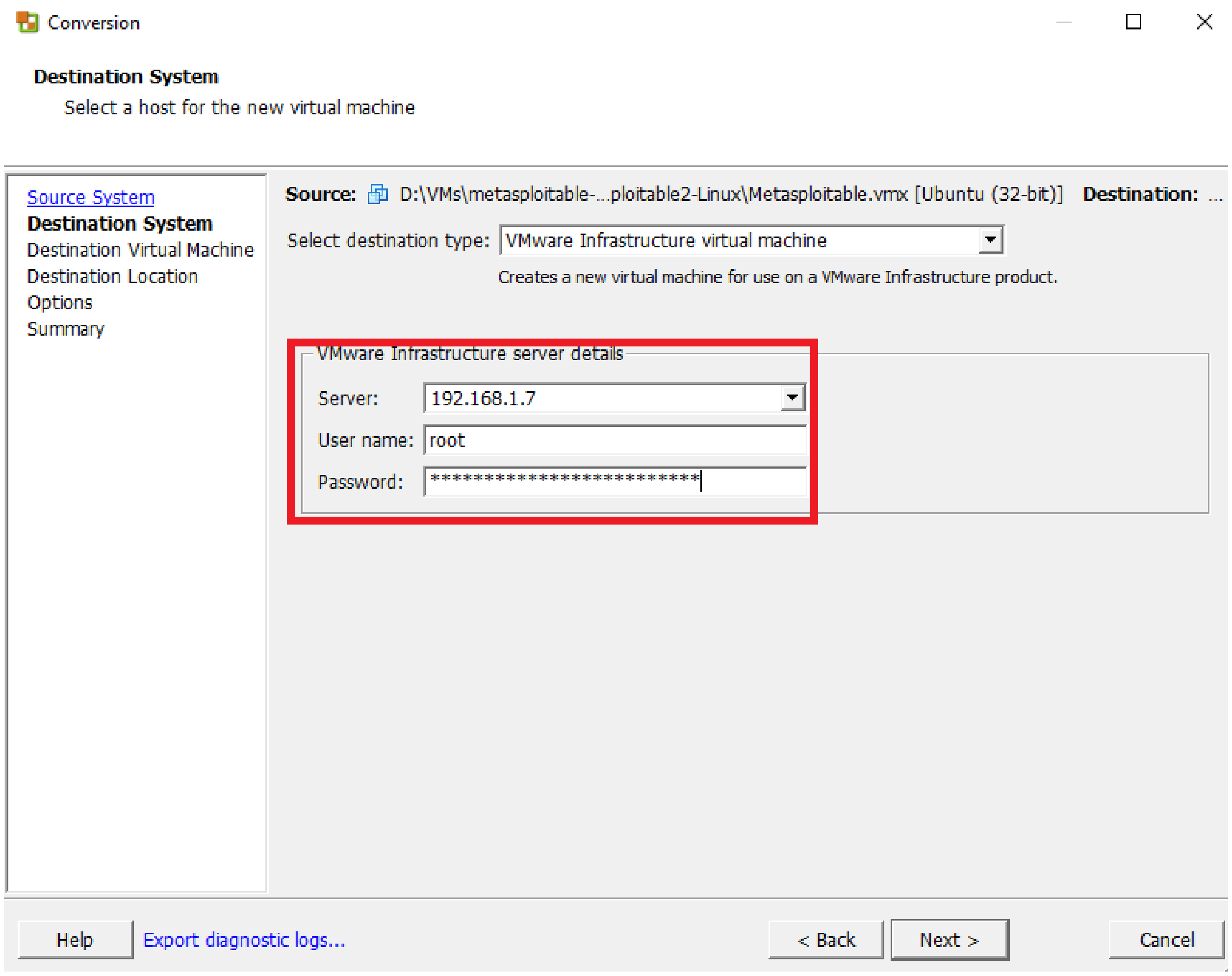
Convert a virtual machine from VMware Workstation, VMware Player, VMware Fusion or other VMware product.

Browse for source virtual machine or image

Virtual machine file: ple2-Linux\Metasploitable.vmx   Browse...

View source details...

On the next screen, make sure that the *Select destination type:* drop-down is set to. *VMware Infrastructure machine*. We will need to enter the IP address of our ESXi server, and the username and password used to log in, because vCenter Converter uploads the VM to ESXi as a part of the conversion process. Enter the IP address or hostname of your ESXi server, and username/password in the *VMware Infrastructure server details* section, then click *Next*.


Conversion

Destination System

Select a host for the new virtual machine

Source: D:\VMs\metasploitable-...ploitable2-Linux\Metasploitable.vmx [Ubuntu (32-bit)] **Destination:** ...

Select destination type: VMware Infrastructure virtual machine

Creates a new virtual machine for use on a VMware Infrastructure product.

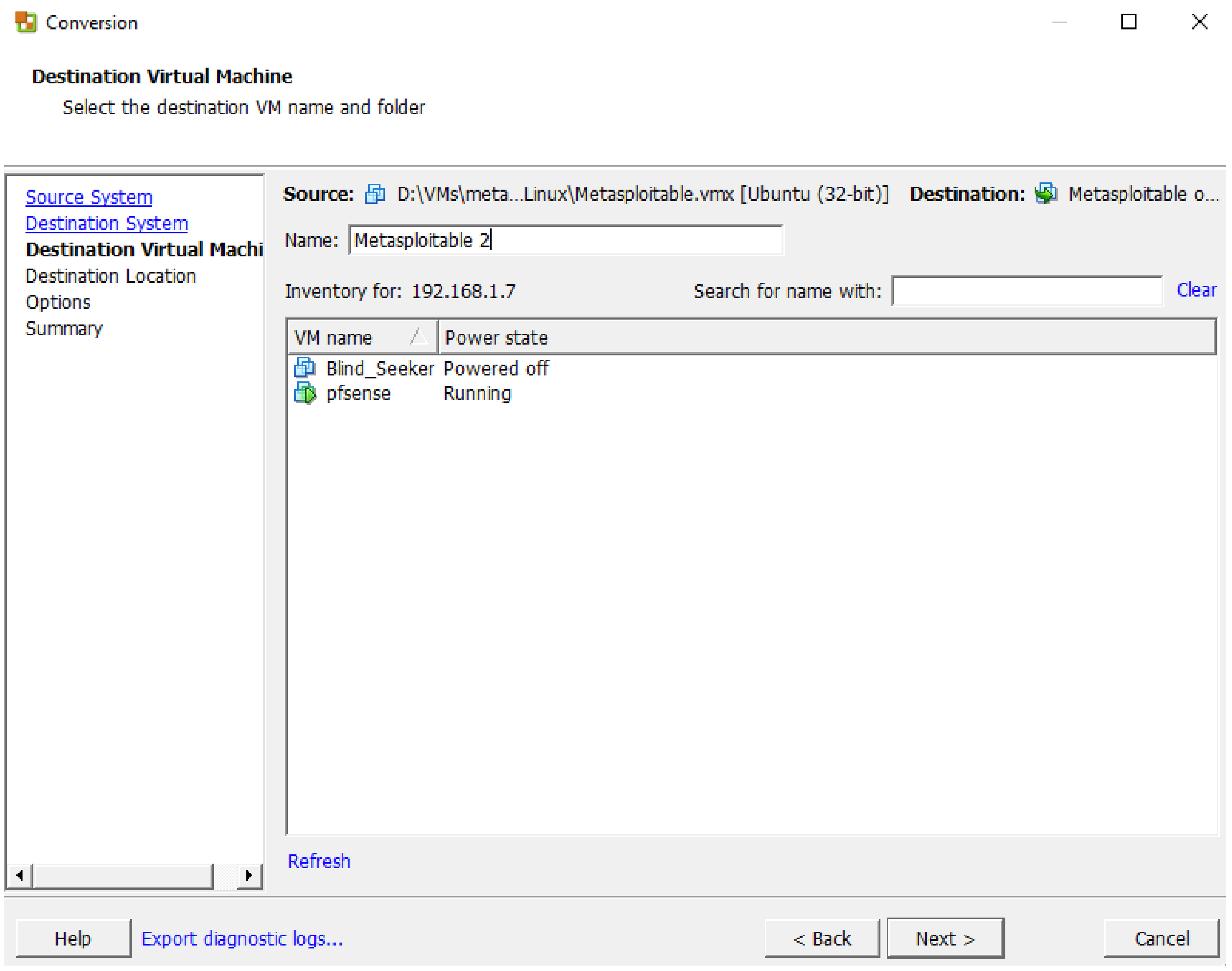
VMware Infrastructure server details

Server: 192.168.1.7

User name: root

Password: ***********************

If the wizard was able to successfully connect and authenticate, on the next screen, it displays a list of the VMs that are currently available on the ESXi server. In the *Name* field, change the text to *Metasploitable 2*, then click *Next*.



Conversion

Destination Virtual Machine
Select the destination VM name and folder

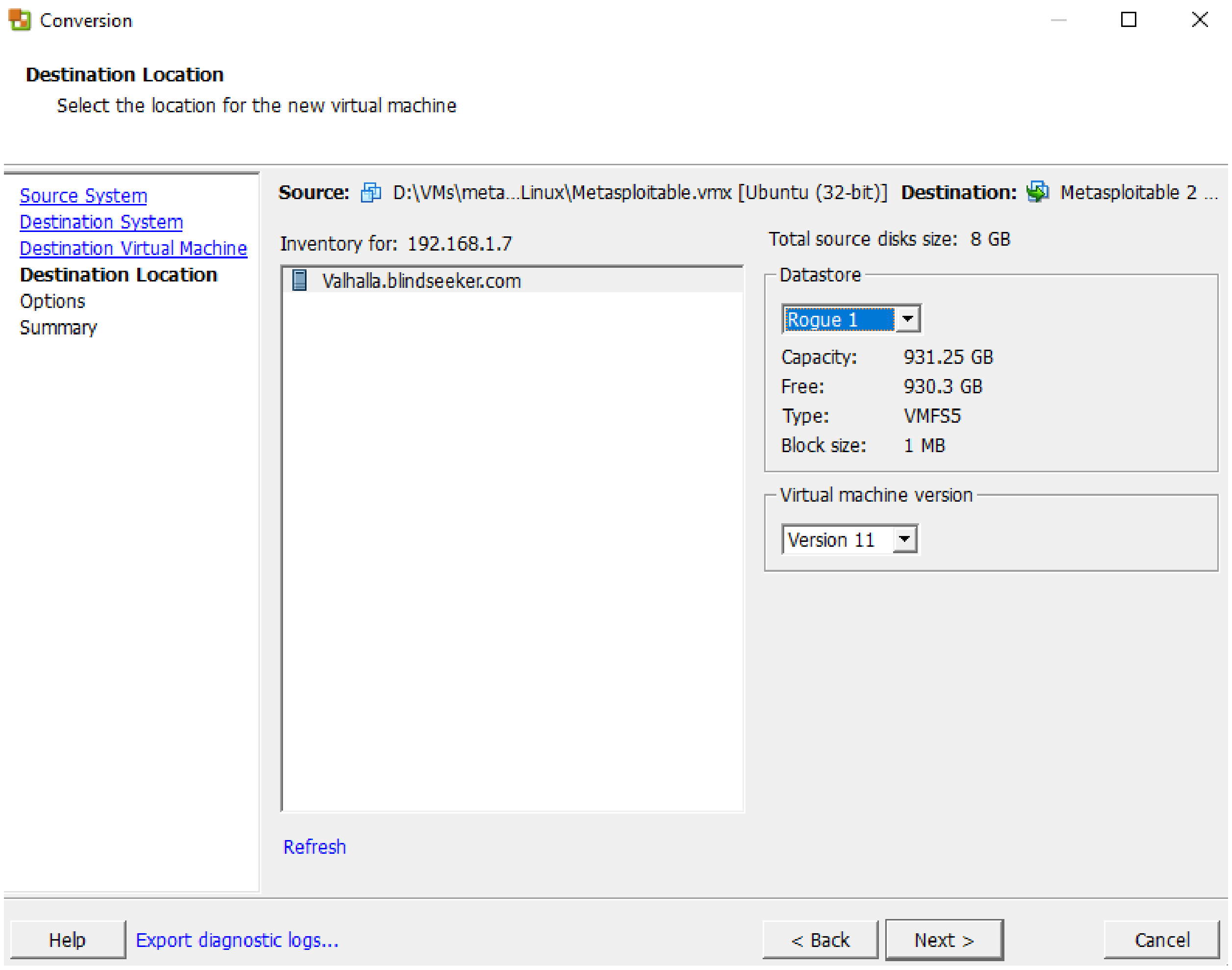
Source: D:\VMs\meta...Linux\Metasploitable.vmx [Ubuntu (32-bit)] **Destination:** Metasploitable o...

Name: Metasploitable 2

Inventory for: 192.168.1.7 Search for name with:

VM name	Power state
Blind_Seeker	Powered off
pfsense	Running

On the following screen, choose the datastore the vCenter Converter should to store the VM in. The wizards offers the option to adjust the Virtual machine hardware version, but you should leave this setting unchanged. For my lab, I chose to store the Metasploitable 2 virtual machine on the *Rogue 1* datastore. Click *Next* when you have selected your datastore.


Conversion

Destination Location

Select the location for the new virtual machine

Source: D:\VMs\meta...Linux\Metasploitable.vmx [Ubuntu (32-bit)] **Destination:** Metasploitable 2 ...

Inventory for: 192.168.1.7

Valhalla.blindseeker.com

Total source disks size: 8 GB

Datastore

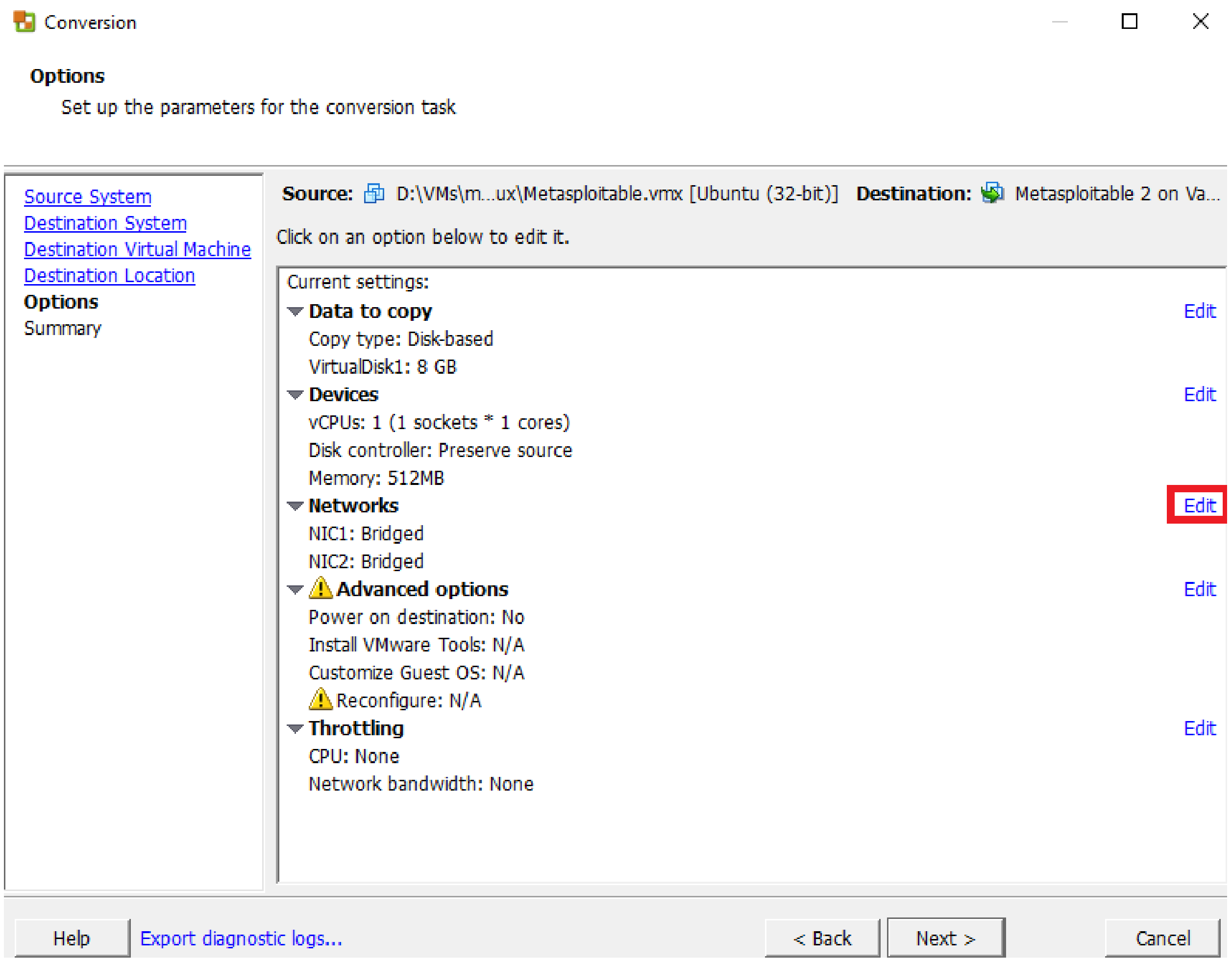
Rogue 1

Capacity:	931.25 GB
Free:	930.3 GB
Type:	VMFS5
Block size:	1 MB

Virtual machine version

Version 11

The next screen of the wizard lists settings for the VM, along with the option to adjust them. We need to remove one of the network cards and change the port group of the remaining one, in order to connect the VM to the *IPS 2* port group. Click the blue *Edit* text to the right of *Networks*.



Source: D:\VMs\m...ux\Metasploitable.vmx [Ubuntu (32-bit)] **Destination:** Metasploitable 2 on Va...

Click on an option below to edit it.

Current settings:

Data to copy
Copy type: Disk-based
VirtualDisk1: 8 GB

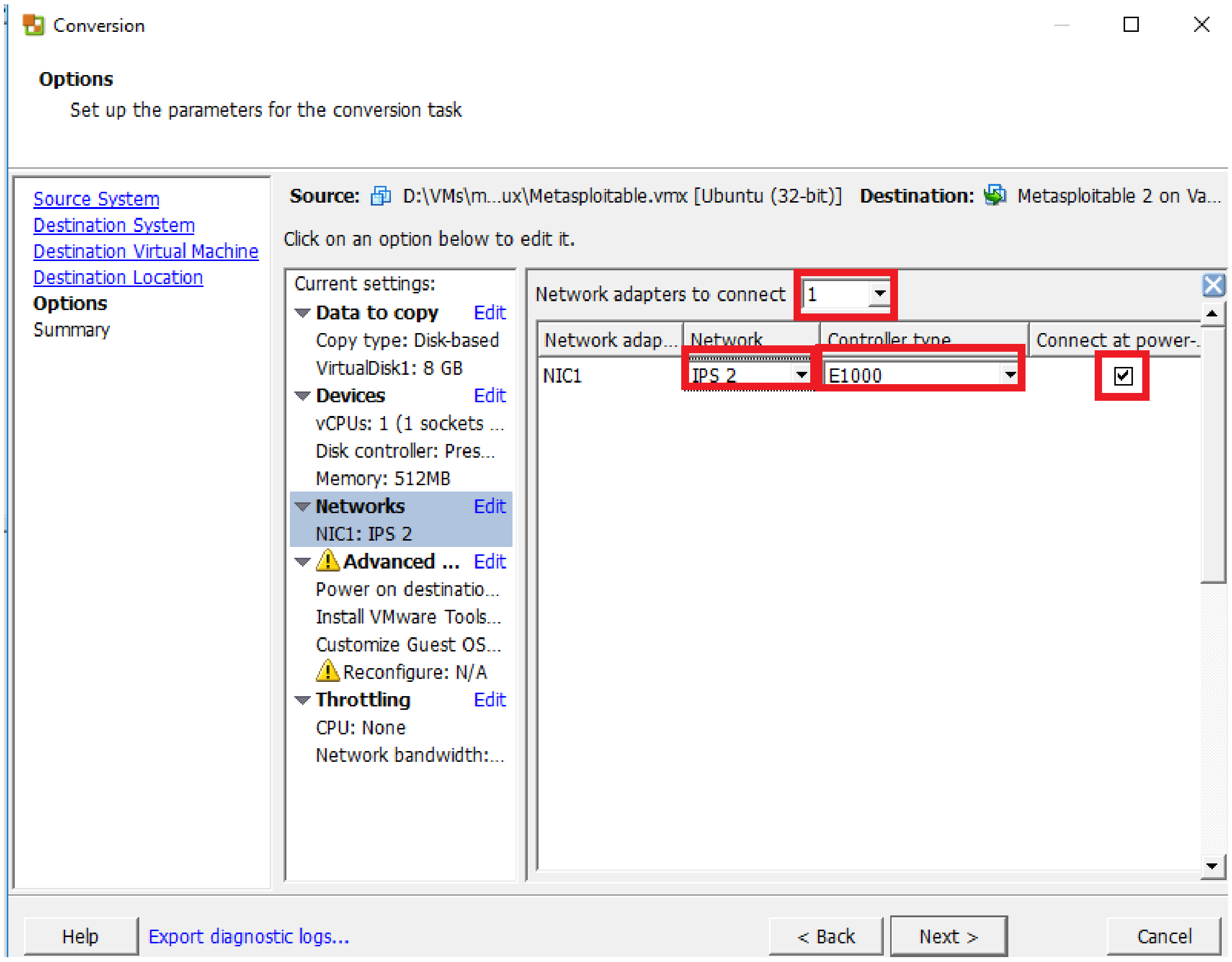
Devices
vCPUs: 1 (1 sockets * 1 cores)
Disk controller: Preserve source
Memory: 512MB

Networks
NIC1: Bridged
NIC2: Bridged

Advanced options
Power on destination: No
Install VMware Tools: N/A
Customize Guest OS: N/A
Reconfigure: N/A

Throttling
CPU: None
Network bandwidth: None

From the *Network adapters to connect* drop-down, select 1. For NIC1, set the *Network* drop-down to *IPS 2*, and the *Controller Type* to *E1000*. Ensure that the *Connect at power on* checkbox is checked, then, click *Next*.



Conversion

Options

Set up the parameters for the conversion task

Source System
Destination System
Destination Virtual Machine
Destination Location
Options
Summary

Source: D:\VMs\m...ux\Metasploitable.vmx [Ubuntu (32-bit)] **Destination:** Metasploitable 2 on Va...

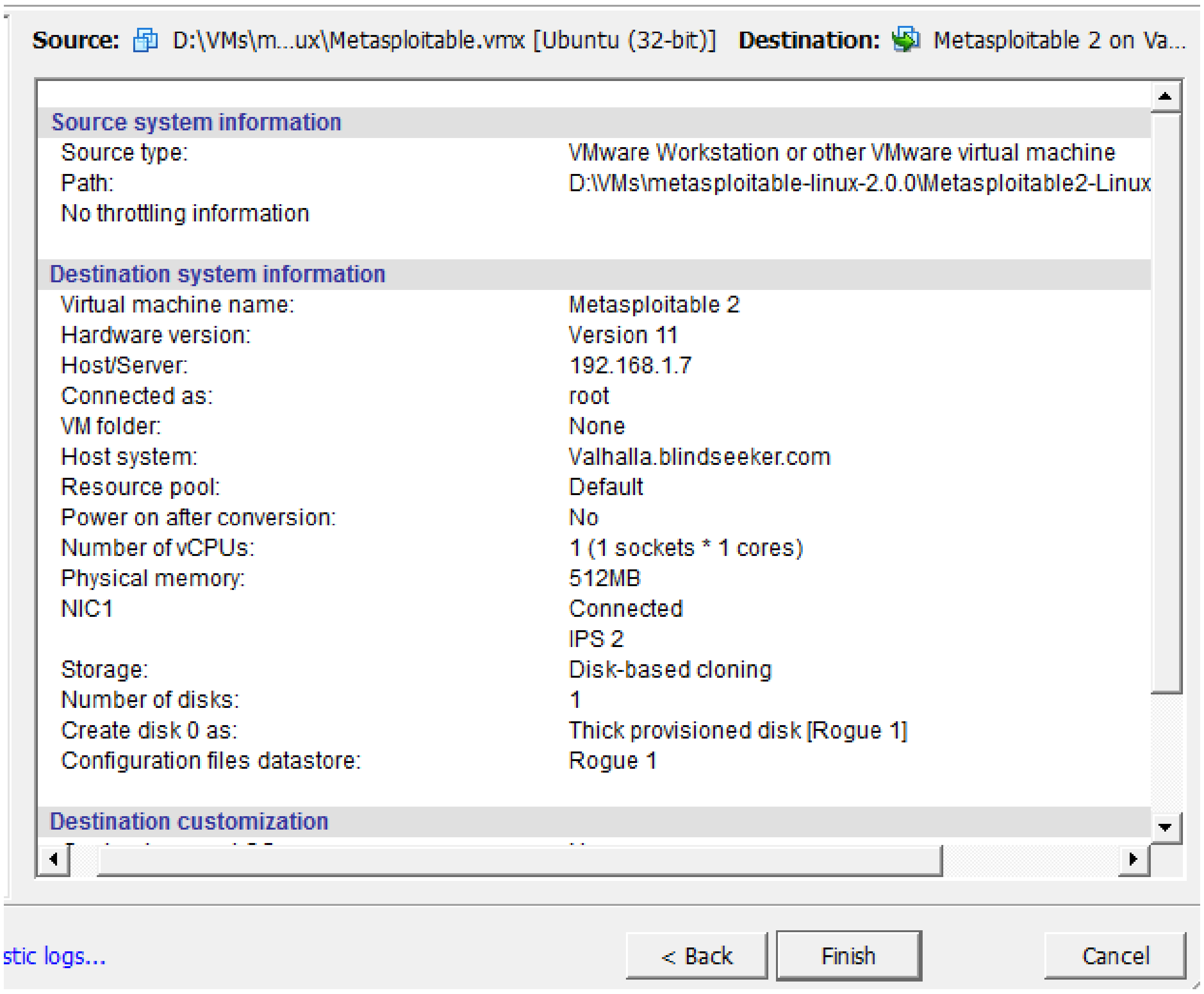
Click on an option below to edit it.

Current settings:
Data to copy Edit
Copy type: Disk-based
VirtualDisk1: 8 GB
Devices Edit
vCPUs: 1 (1 sockets ...
Disk controller: Pres...
Memory: 512MB
Networks Edit
NIC1: IPS 2
Advanced ... Edit
Power on destinatio...
Install VMware Tools...
Customize Guest OS...
Reconfigure: N/A
Throttling Edit
CPU: None
Network bandwidth:...

Network adapters to connect: 1

Network adap...	Network	Controller type	Connect at power-...
NIC1	IPS 2	E1000	☑

The final screen displays a summary of all the configuration settings that will be used to convert the virtual machine: the source, the destination, hardware settings, OS settings, etc. Click *Finish* to begin the conversion process.



Source: D:\VMs\m...ux\Metasploitable.vmx [Ubuntu (32-bit)] **Destination:** Metasploitable 2 on Va...

Source system information

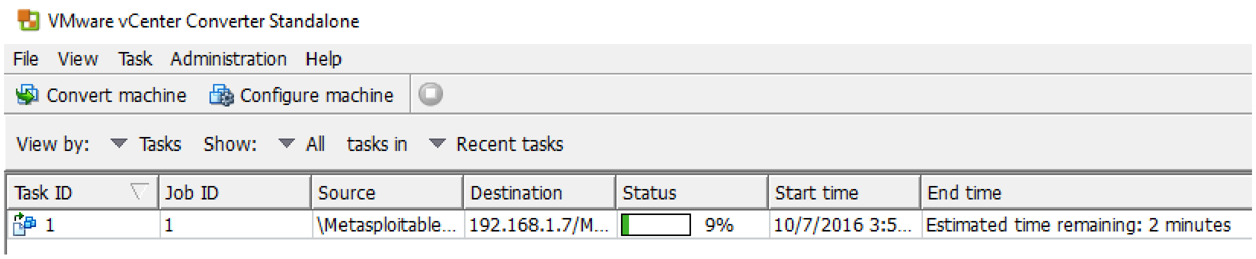
Source type:	VMware Workstation or other VMware virtual machine
Path:	D:\VMs\metasploitable-linux-2.0.0\Metasploitable2-Linux
No throttling information	

Destination system information

Virtual machine name:	Metasploitable 2
Hardware version:	Version 11
Host/Server:	192.168.1.7
Connected as:	root
VM folder:	None
Host system:	Valhalla.blindseeker.com
Resource pool:	Default
Power on after conversion:	No
Number of vCPUs:	1 (1 sockets * 1 cores)
Physical memory:	512MB
NIC1	Connected
	IPS 2
Storage:	Disk-based cloning
Number of disks:	1
Create disk 0 as:	Thick provisioned disk [Rogue 1]
Configuration files datastore:	Rogue 1

Destination customization

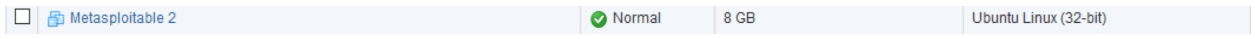
The Conversion wizard closes, and the main application window of the vCenter Converter will update, showing the progress of the current VM conversion. You will need to wait for it to finish. Depending on the hardware resources available, this might take a while.



VMware vCenter Converter Standalone

Task ID	Job ID	Source	Destination	Status	Start time	End time
1	1	\Metasploitable...	192.168.1.7/M...	9%	10/7/2016 3:5...	Estimated time remaining: 2 minutes

After the conversion is complete, the new VM should appear in the Virtual Machines list on your ESXi's web interface.



Metasploitable 2	Normal	8 GB	Ubuntu Linux (32-bit)

Power on the Metasploitable 2 VM to ensure that the system boots up. If it does and reaches the login prompt properly, your work here is done for now. You may shut down this virtual machine for now. Navigate to the *Network Adapter* settings for *Metasploitable 2*, and record the MAC address for this system. Next, log in to the pfSense web UI, navigate to *Services > DHCP Server*, and add a static mapping to the OPT1 interface DHCP for the MAC address of the Metasploitable 2 VM. Assign it the IP address 172.16.2.3, making sure to enter a description that tells you which VM this static mapping belongs to!

Next Steps

The VMware vSphere Hypervisor initial setup is all but done at this point. However, you're not quite out of the woods just yet. Here is a checklist of tasks to complete:

- If you haven't completed the section Defense in Depth for Windows Hosts, and you are using a Windows host as your management workstation, I would highly advise doing so.
- You will also want to complete the Remote Lab Management section that pertains to the OS of the system you will be using to access your lab VMs (e.g. Windows Remote Access, Linux/Unix Remote Access). You might also be interested in Enabling SSH on Kali Linux, and/or if you're lazy like me, the section on Securing root SSH access.
- Are you in a network environment where your laptop or workstation's IP address cannot be statically assigned or configured? You will want to set up what is called a bastion host, or "jumpbox". The section Network Design Factors When Working with bare-metal Hypervisors will help you better understand bare-metal hypervisor networks, and how to create and configure a jumpbox for consistent access to your lab environment.
- You have to install IDS/IPS software on the IPS VM. You'll need to complete the IPS Installation Guide to learn how to do this with either Snort or Suricata as your IPS software of choice.
- The SIEM VM needs to have Splunk installed and configured. You'll need to complete the Splunk Installation Guide.

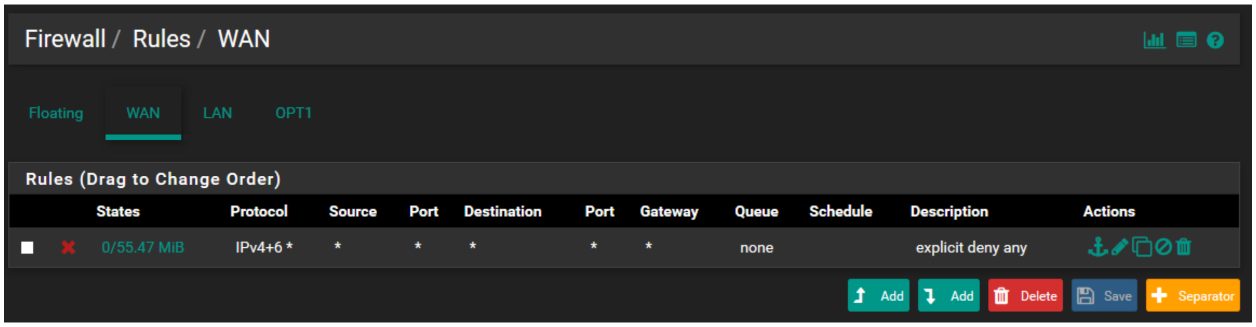
- You may want to consider reading the Automated Patching for Linux Lab VMs chapter, and implementing the `updater.sh` script for your Linux VMs.
- Do you want some ideas on where to take your lab? Check out the chapter In Your Own Image, for some tips on how to mold your VM lab to better suit your needs.

pfSense Firewall Rules and Network Services Guide

This chapter will guide you through finalizing the setup of pfSense firewall for your lab network. We will be going over firewall rule recommendations for the *WAN* (*Bridged* network gateway), *LAN* (*Management* network gateway), and *OPT1* (gateway for the *IPS 1* and *IPS 2* networks, which will collectively be referred to as the IPS network) interfaces, and configuring core network services for our lab virtual machines.

Firewall Rule Configuration - Hosted Hypervisors

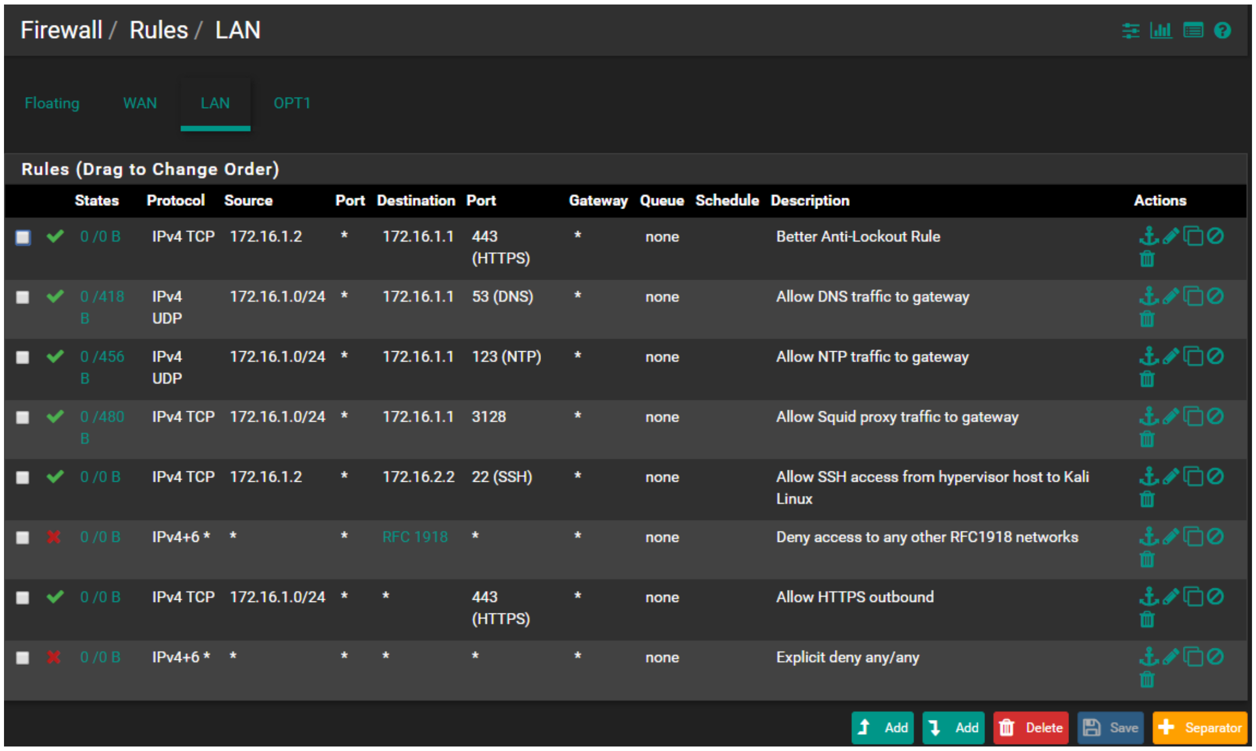
Firewall Rules for the Bridged Network



		States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐	✗	0/55.47 MiB	IPv4+6 *	*	*	*	*	*	none		explicit deny any	

The illustration above is a screen capture from the pfSense web interface, showing the firewall policy for the *WAN* interface, connected to your lab's *Bridged* network. While pfSense has an implicit deny any/any firewall rule, I chose to manually create an explicit version of this rule to serve as a reminder of this behavior. This firewall rule will serve to block all traffic inbound to the lab networks, but does NOT prevent the VMs in our network from interacting with hosts on the network you are bridged to, or the internet. That task falls to firewall rules for the other network interfaces, which we will be going over shortly.

Firewall Rules for the Management Network



Rules (Drag to Change Order)

		States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	✓	0 /0 B	IPv4 TCP	172.16.1.2	*	172.16.1.1	443 (HTTPS)	*	none		Better Anti-Lockout Rule	
	✓	0 /418 B	IPv4 UDP	172.16.1.0/24	*	172.16.1.1	53 (DNS)	*	none		Allow DNS traffic to gateway	
	✓	0 /456 B	IPv4 UDP	172.16.1.0/24	*	172.16.1.1	123 (NTP)	*	none		Allow NTP traffic to gateway	
	✓	0 /480 B	IPv4 TCP	172.16.1.0/24	*	172.16.1.1	3128	*	none		Allow Squid proxy traffic to gateway	
	✓	0 /0 B	IPv4 TCP	172.16.1.2	*	172.16.2.2	22 (SSH)	*	none		Allow SSH access from hypervisor host to Kali Linux	
	✗	0 /0 B	IPv4+6 *	*	*	RFC 1918	*	*	none		Deny access to any other RFC1918 networks	
	✓	0 /0 B	IPv4 TCP	172.16.1.0/24	*	*	443 (HTTPS)	*	none		Allow HTTPS outbound	
	✗	0 /0 B	IPv4+6 *	*	*	*	*	*	none		Explicit deny any/any	

The image above shows the firewall policy for the pfSense's *LAN* interface, which is connected to the *Management* network. First and foremost, it has to be stressed that the **order of the firewall rules is important**. Network traffic is evaluated against the set of rules from the top down. If there is a rule denying a certain kind of network traffic placed ABOVE a rule that is meant to allow the traffic in question, that deny rule will be checked first, and the traffic WILL be dropped. **pfSense, like most other firewalls, evaluates firewall rules in a policy from top to bottom**. Be mindful of this!

The first rule from the top in the illustration above, is a stricter version of the pfSense anti-lockout rule. By default, pfSense includes an "anti-lockout" rule for the LAN interface that allows any system on the LAN (e.g. *Management*) network to access the firewall over HTTPS. I didn't like how open this rule was, so this new rule only allows access to the web interface (port 443/tcp) originating from a specific IP address on this network, 172.16.1.2. If you are using a hosted hypervisor, this should be the IP address you statically assigned to the virtual interface attached to this network, that has been allocated to the host operating system.

The next three rules allow access from the 172.16.1.0/24 network to the host with the IP address 172.16.1.1. This way, VMs on the *Management* network are able to use the pfSense gateway for network services, specifically NTP (port 123/udp), DNS (port 53/udp), and Squid proxy (port 3128/tcp).

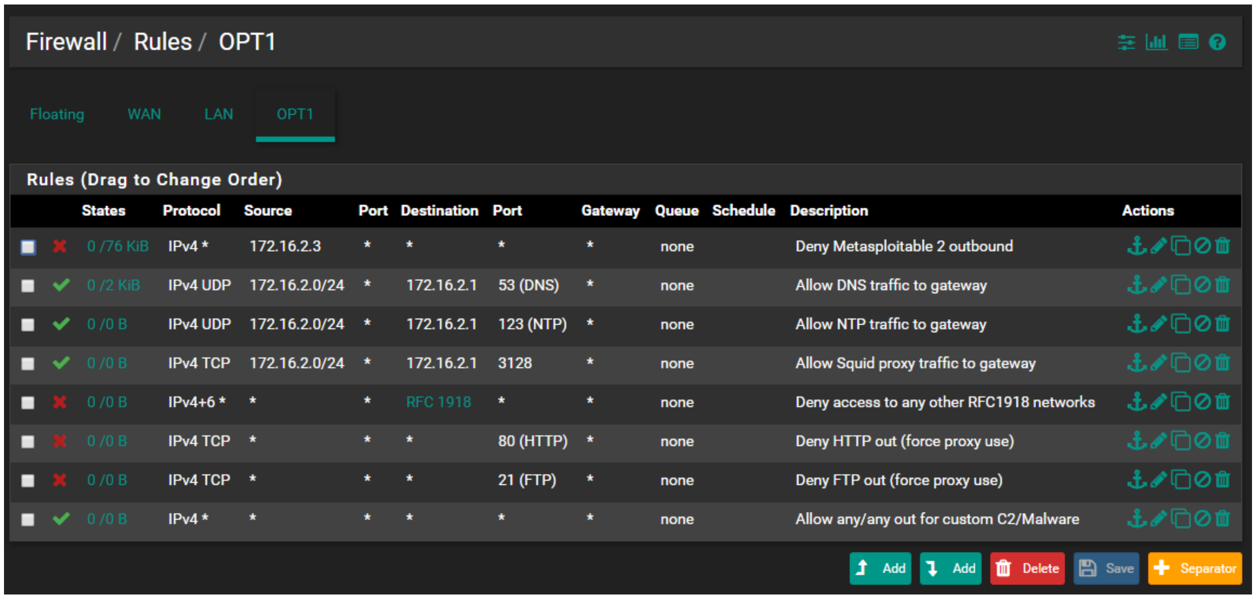
The rule allowing port 22/tcp (the default port for SSH) from 172.16.1.2 to 172.16.2.2 should allow you to access the Kali Linux VM from your hypervisor host via SSH with ease. Please note that the Kali Linux virtual machine will need to have the SSH service running, and that your hypervisor host will need to have a route to the 172.16.2.0/24 network in order to establish SSH connections. The details on how to perform these actions are discussed in the Remote Lab Management guide.

The third rule from the bottom exists to explicitly deny access to any local networks (RFC 1918 addresses - 172.16.0.0/12, 192.168.0.0/16, and 10.0.0.0/8). This ensures that the VMs in the *Management* network can neither talk to the physical network nor to any of the other virtual networks without having rules in place that specifically allow certain kinds of connections. Please note that if you need to add firewall rules to allow specific traffic between the virtual networks, or from the Management network to the physical network, those rules MUST be defined ABOVE this rule in order to function. For example, the SSH rule above would cease working if it was defined below this firewall rule due to the firewall interpreting the rules from top to bottom, blocking the connection to an RFC1918 address (172.16.2.2).

The second to last rule, allows HTTPS traffic (443/tcp) outbound, to the internet. This rule is defined to make it possible for VMs in the 172.16.1.0/24 network (e.g. the *Management* network) to connect to the internet over HTTPS, to allow for downloading patches or new software via HTTPS. The pfSense HTTP proxy is able to handle HTTP and FTP, so we only need to define a rule to explicitly allow HTTPS traffic. This entry is placed BELOW the rule that blocks access to RFC 1918 addresses so that the RFC1918 firewall rule is evaluated first. This ensures that systems on the *Management* network cannot establish HTTPS connections to the *IPS* network(s), or your local physical network, yet still allows HTTPS connections to the internet.

Finally, there is an explicit deny any/any catch-all rule to block any network traffic that hasn't been defined in the firewall policy for this interface. pfSense already implicitly denies any traffic not defined in a firewall rule, but having the rule present and explicitly defined serves as a reminder of this behavior for network troubleshooting.

Firewall Rules for the IPS Network



Rules (Drag to Change Order)

		States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	✗	0 /76 KiB	IPv4 *	172.16.2.3	*	*	*	*	none		Deny Metasploitable 2 outbound	
	✓	0 /2 KiB	IPv4 UDP	172.16.2.0/24	*	172.16.2.1	53 (DNS)	*	none		Allow DNS traffic to gateway	
	✓	0 /0 B	IPv4 UDP	172.16.2.0/24	*	172.16.2.1	123 (NTP)	*	none		Allow NTP traffic to gateway	
	✓	0 /0 B	IPv4 TCP	172.16.2.0/24	*	172.16.2.1	3128	*	none		Allow Squid proxy traffic to gateway	
	✗	0 /0 B	IPv4+6 *	*	*	RFC 1918	*	*	none		Deny access to any other RFC1918 networks	
	✗	0 /0 B	IPv4 TCP	*	*	*	80 (HTTP)	*	none		Deny HTTP out (force proxy use)	
	✗	0 /0 B	IPv4 TCP	*	*	*	21 (FTP)	*	none		Deny FTP out (force proxy use)	
	✓	0 /0 B	IPv4 *	*	*	*	*	*	none		Allow any/any out for custom C2/Malware	

The picture above displays the firewall policy for pfSense's *OPT1* interface, representing connectivity to and from our lab's *IPS* network. You'll notice that there is a firewall rule at the bottom of this firewall policy, that allows all network traffic not explicitly defined in the rules above inbound or outbound. Explaining the purpose of this rule first, will make the rule deviations from the *Management* network's firewall rules make more sense. When I set up the *IPS* network, I intended it to serve as a malware analysis environment. There is a lot of malware that uses custom protocols and/or ports. If that traffic isn't allowed outbound, then you cannot observe the malware's command and control traffic. This rule is to allow for malware that uses custom TCP or UDP ports for command and control to connect to the internet. In addition to this rule's purpose as a catch allow to allow random traffic, it also means I don't need an explicit firewall rule to allow HTTPS (443/tcp) traffic outbound.

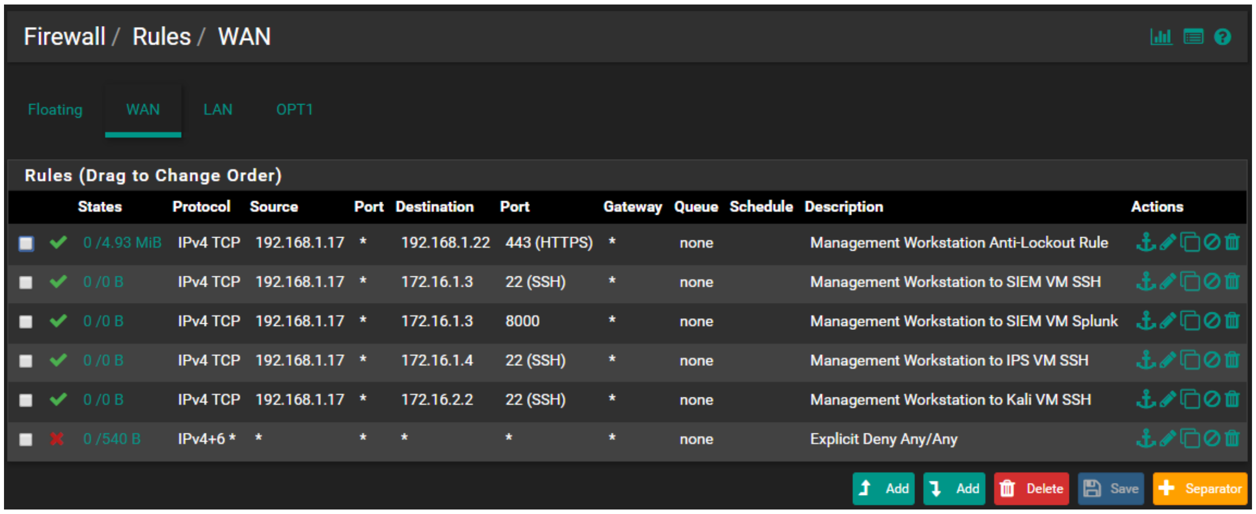
Now that you know the purpose of the last rule, let's discuss the rest of the firewall rules, starting from the top. The first rule exists to specifically deny the Metasploitable 2 VM access to any systems outbound. **Metasploitable 2 is an extremely vulnerable virtual machine that should not be exposed, or have connectivity to external networks ever**. The next three rules are there to allow VMs on this network to use pfSense for DNS (53/udp), NTP (123/udp), and Squid proxy (3128/tcp) services. Just like with the management network, the next rule prohibits systems on the IPS network from establishing connections to other RFC1918 networks, effectively preventing hosts on this network from being able to initiate connections to hosts on the *Management* network, or your physical, *Bridged* network. The next two rules explicitly deny HTTP and FTP traffic outbound. This is because the Squid proxy we will be installing can proxy both HTTP and FTP traffic. Due to our last firewall rule allowing any traffic in or out that isn't explicitly defined above, we have to block HTTP and FTP outbound to force systems on this network to use the Squid proxy, so that these connections can be logged. You may have noticed that I haven't explicitly blocked port 53/udp outbound for the *IPS* network. This is another malware analysis configuration I have accounted for. Some malware families will alter DNS settings, or use DNS tunneling for command and control, so I allow systems on this network to use external DNS servers.

Please note, that if you are not planning on using the *IPS* network for malware analysis, you may want to adapt the firewall rules for the *Management* network, with the exception of the anti-lockout and SSH rule.

Firewall Rule Configuration - Bare-metal Hypervisors

The firewall rulesets above assume that your lab network is running on a hosted hypervisor. Hosted hypervisors have the capability to add a virtual network card to the Management network that allows them more or less direct access to the lab networks. Bare-metal hypervisors do not have this luxury, requiring you to access both the hypervisor software and your lab network through an external management workstation, or some sort of a jump box, hosted on an external, physical network. As such, there are some unique configurations baremetal hypervisor users need to be aware of when crafting their firewall rules.

Firewall rules for the Bridged Network



Firewall / Rules / WAN

Rules (Drag to Change Order)

		States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	✓	0 /4.93 MiB	IPv4 TCP	192.168.1.17	*	192.168.1.22	443 (HTTPS)	*	none		Management Workstation Anti-Lockout Rule	
	✓	0 /0 B	IPv4 TCP	192.168.1.17	*	172.16.1.3	22 (SSH)	*	none		Management Workstation to SIEM VM SSH	
	✓	0 /0 B	IPv4 TCP	192.168.1.17	*	172.16.1.3	8000	*	none		Management Workstation to SIEM VM Splunk	
	✓	0 /0 B	IPv4 TCP	192.168.1.17	*	172.16.1.4	22 (SSH)	*	none		Management Workstation to IPS VM SSH	
	✓	0 /0 B	IPv4 TCP	192.168.1.17	*	172.16.2.2	22 (SSH)	*	none		Management Workstation to Kali VM SSH	
	✗	0 /540 B	IPv4+6 *	*	*	*	*	*	none		Explicit Deny Any/Any	

If you followed the ESXi setup guide, you should have a firewall rule on the *WAN* interface created by the `easyrule` command line utility that you used to access the pfSense webConfigurator from your management workstation. In the illustration above, our first rule allows our management workstation (192.168.1.17 in my case, your management system's IP address may vary) to access the pfSense firewall's WAN interface (192.168.1.22 in my case, again your IP address may vary) over port 443/tcp. As the rule description states, this is our webConfigurator anti-lockout rule.

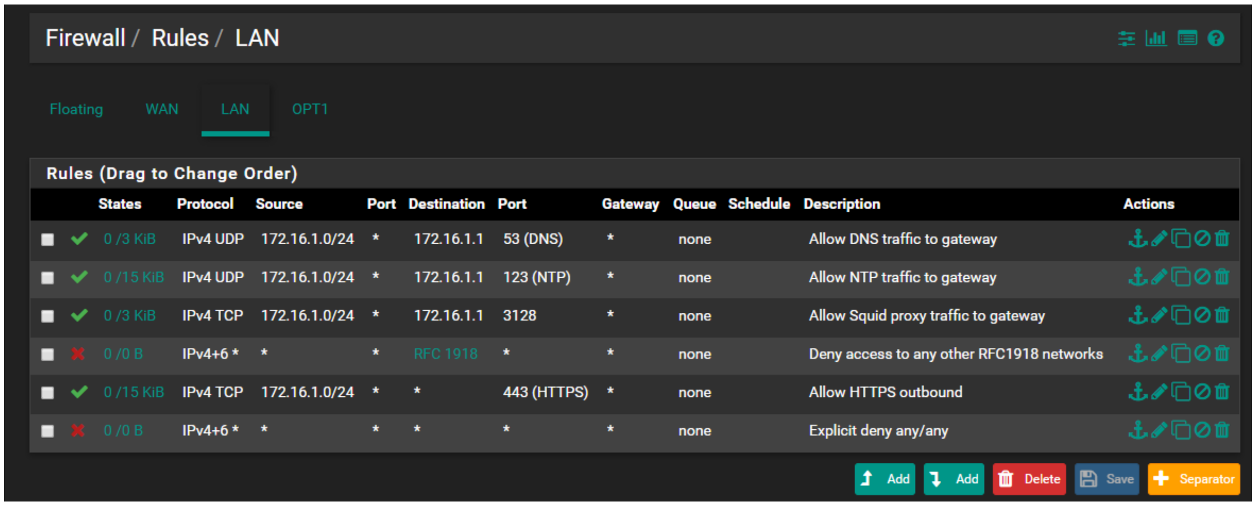
Please note that this anti-lockout rule relies on both the management workstation and the pfSense WAN interface having either a static IP address or a static DHCP allocation to ensure that these IP addresses do not change. Make peace with your network or system administrators, look up the documentation for your network gear, do what you must, but **the pfSense VM's WAN interface must be consistent**. If you work in a network where your management workstation cannot acquire a static DHCP address or static IP address, you may want to considering implementing a bastion host, or "jump box" that you can configure with a static IP address or DHCP mapping and use that to tunnel or proxy your network traffic in order to maintain access to your lab network. Information on how to do this is covered at the end of the ESXi hypervisor guide, after you have finished creating your firewall rules and remaining lab virtual machines.

The next four allow rules are configured to allow our management workstation's IP address to access the SIEM, IPS and Kali Linux virtual machines over SSH (22/tcp), as well as access the

SIEM VM over port 8000/tcp (The default TCP port that the splunk web interface listens on. Please bear in mind that in addition to these firewall rules, your management workstation or jump box will require static routes to the 172.16.1.0/24 and 172.16.2.0/24 networks. Information on how to configure static routes is covered at the end of the ESXi hypervisor guide.

The final rule at the bottom of the WAN interface is an explicit deny any/any rule. By default pfSense drops any traffic there is not a firewall rule to explicitly allow. I create this rule explicitly to serve as a reminder for network troubleshooting.

Firewall Rules for the Management Network

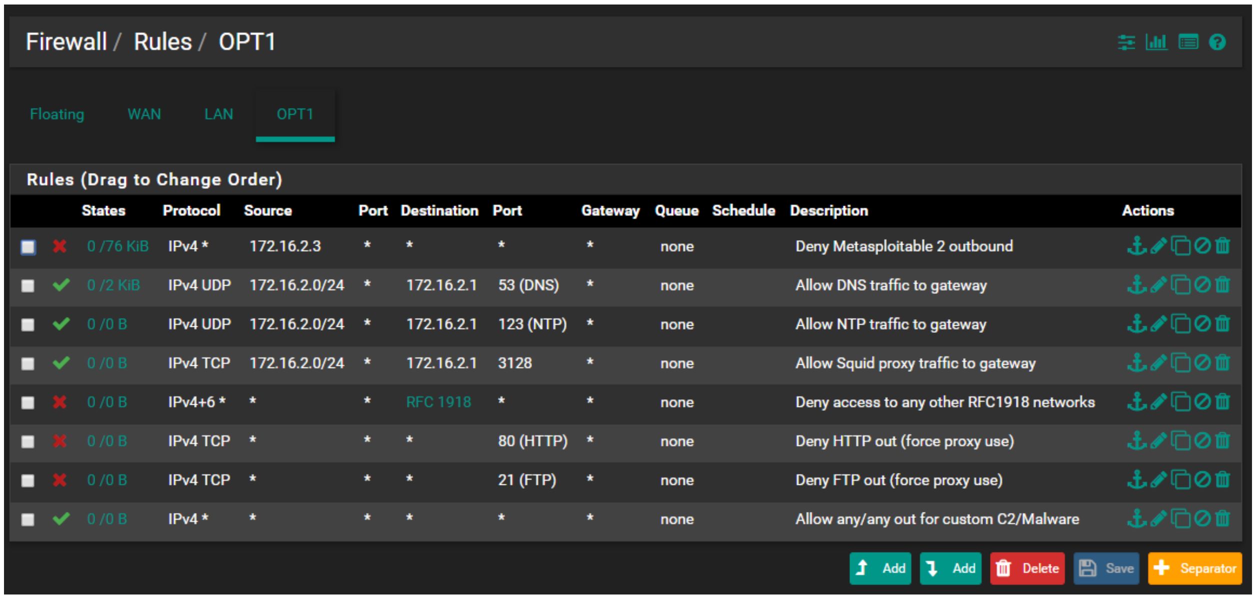


Rules (Drag to Change Order)

		States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	✓	0 /3 KiB	IPv4 UDP	172.16.1.0/24	*	172.16.1.1	53 (DNS)	*	none		Allow DNS traffic to gateway	
	✓	0 /15 KiB	IPv4 UDP	172.16.1.0/24	*	172.16.1.1	123 (NTP)	*	none		Allow NTP traffic to gateway	
	✓	0 /3 KiB	IPv4 TCP	172.16.1.0/24	*	172.16.1.1	3128	*	none		Allow Squid proxy traffic to gateway	
	✗	0 /0 B	IPv4+6 *	*	*	RFC 1918	*	*	none		Deny access to any other RFC1918 networks	
	✓	0 /15 KiB	IPv4 TCP	172.16.1.0/24	*	*	443 (HTTPS)	*	none		Allow HTTPS outbound	
	✗	0 /0 B	IPv4+6 *	*	*	*	*	*	none		Explicit deny any/any	

The rules for the LAN interface are pretty straightforward, so let's go over them from the top. Our first three rules allow VMs on this network to use the pfSense for DNS (53/udp), NTP (123/udp), and Squid proxy (3128/tcp) services. The next rule denies virtual machines on this network the ability to access other local RFC1918 networks, such as the *IPS* network, or the physical, *Bridged* network. The next rule allows virtual machines on this network to make HTTPS requests outbound. This rule comes after the rule denying access to RFC1918 networks, so that means that systems can only make HTTPS requests to the internet, any local HTTPS requests outbound will be dropped. This rule is specifically available because we cannot use Squid to proxy HTTPs connections (without some serious effort), and systems on this network may need HTTPS access to download updates or additional software. Finally, the last rule is our explicit deny any/any rule, identical to the last rule on the WAN interface, used mainly as a reminder of pfSense’s default firewall rule behavior and/or troubleshooting.

Firewall Rules for the IPS Network



Firewall / Rules / OPT1

Floating WAN LAN OPT1

Rules (Drag to Change Order)

		States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	✗	0 /76 KiB	IPv4 *	172.16.2.3	*	*	*	*	none		Deny Metasploitable 2 outbound	
	✓	0 /2 KiB	IPv4 UDP	172.16.2.0/24	*	172.16.2.1	53 (DNS)	*	none		Allow DNS traffic to gateway	
	✓	0 /0 B	IPv4 UDP	172.16.2.0/24	*	172.16.2.1	123 (NTP)	*	none		Allow NTP traffic to gateway	
	✓	0 /0 B	IPv4 TCP	172.16.2.0/24	*	172.16.2.1	3128	*	none		Allow Squid proxy traffic to gateway	
	✗	0 /0 B	IPv4+6 *	*	*	RFC 1918	*	*	none		Deny access to any other RFC1918 networks	
	✗	0 /0 B	IPv4 TCP	*	*	*	80 (HTTP)	*	none		Deny HTTP out (force proxy use)	
	✗	0 /0 B	IPv4 TCP	*	*	*	21 (FTP)	*	none		Deny FTP out (force proxy use)	
	✓	0 /0 B	IPv4 *	*	*	*	*	*	none		Allow any/any out for custom C2/Malware	

You'll notice that there is a firewall rule at the bottom of this firewall policy, that allows all network traffic not explicitly defined in the rules above inbound or outbound. Explaining the purpose of this rule first, will make the rule deviations from the *Management* network's firewall rules make more sense. When I set up the *IPS* network, I intended it to serve as a malware analysis environment. There is a lot of malware that uses custom protocols and/or ports. If that traffic isn't allowed outbound, then you cannot observe the malware's command and control traffic. This rule is to allow for malware that uses custom TCP or UDP ports for command and control to connect to the internet. In addition to this rule's purpose as a catch allow to allow random traffic, it also means I don't need an explicit firewall rule to allow HTTPS (443/tcp) traffic outbound.

Now that you know the purpose of the last rule, let's discuss the rest of the firewall rules, starting from the top. The first rule exists to specifically deny the Metasploitable 2 VM access to any systems outbound. **Metasploitable 2 is an extremely vulnerable virtual machine that should not be exposed, or have connectivity to external networks ever**. The next three rules are there to allow VMs on this network to use pfSense for DNS (53/udp), NTP (123/udp), and Squid proxy (3128/tcp) services. Just like with the management network, the next rule prohibits systems on the IPS network from establishing connections to other RFC1918 networks, effectively preventing hosts on this network from being able to initiate connections to hosts on the *Management* network, or your physical, *Bridged* network. The next two rules explicitly deny HTTP and FTP traffic outbound. This is because the Squid proxy we will be installing can proxy both HTTP and FTP traffic. Due to our last firewall rule allowing any traffic in or out that isn't explicitly defined above, we have to block HTTP and FTP outbound to force systems on this network to use the Squid proxy, so that these connections can be logged. You may have noticed that I haven't explicitly blocked port 53/udp outbound for the *IPS* network. This is another malware analysis configuration I have accounted for. Some malware families will alter DNS settings, or use DNS tunneling for command and control, so I allow systems on this network to use external DNS servers.

Please note, that if you are not planning on using the *IPS* network for malware analysis, you may want to adapt the firewall rules for the *Management* network

Network Configuration - Core Network Services

As hinted above and throughout the sections dedicated to the setup of the different hypervisors, pfSense is configured to offer a few core network services for the systems of the lab environment. In the following sections, we will go over the services used, and how to set them up to meet the needs of our lab network.

NTP

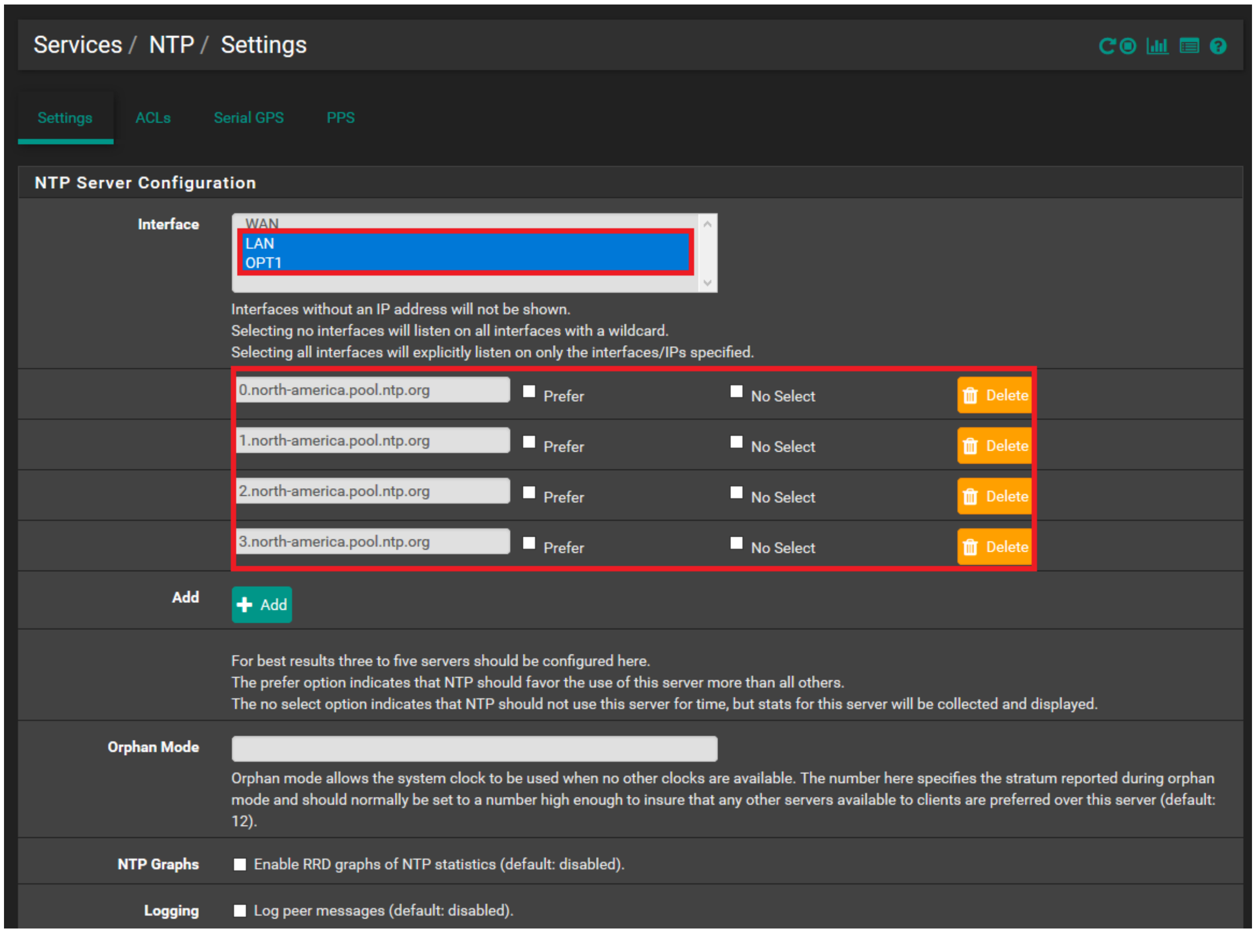
NTP is shorthand for Network Time Protocol. It is a service that allows systems to synchronize their internal clocks with a known (or at least assumed) good time source. Using NTP prevents system clocks from drifting, which has very important connotations for incident response, since you need to ensure that the timestamps on all the data collected are accurate.

On the pfSense web UI, navigate to *Services* > *NTP*. In the listbox on top of the page, highlight the LAN and the OPT1 interfaces to ensure that upon completion, the NTP service and the adjusted settings will be made available for those two networks. (To select the two entries, press and hold the Ctrl key (Windows) or the Cmd key (macOS) respectively, click LAN and then, in turn, click OPT1.) By default, pfSense uses the NTP pool provided by its maintainers, and in most cases, this should work well enough. However, it never hurts to know alternatives. Visit the NTP Pool Project at http://www.pool.ntp.org/en/ and find the region of the world you are living in. In my case, I selected “North America” from the list of active servers, and on the next page, I was presented the following NTP server information:

```
0.north-america.pool.ntp.org
1.north-america.pool.ntp.org
2.north-america.pool.ntp.org
3.north-america.pool.ntp.org
```

Using the information NTP Pool Project page for your region, you can click the green *Add* button and add the additional NTP servers. After adding all four servers I found listed, I removed the default NTP server pool provided by pfSense by clicking on the corresponding *Delete* button on the right. After making the changes for the LAN and the OPT1 interface, click *Save* to enable the service.

Note: If you are setting up your lab network on a corporate/enterprise network, you will need to ensure that port 123/udp is allowed outbound on your enterprise/local network. Otherwise, your network may be big enough to where there is already a designated NTP server in use on your network. You'll want to use the IP address of that NTP server instead, if necessary.



NTP Server Configuration

Interface: WAN / LAN / OPT1 (LAN and OPT1 selected)

Interfaces without an IP address will not be shown.
Selecting no interfaces will listen on all interfaces with a wildcard.
Selecting all interfaces will explicitly listen on only the interfaces/IPs specified.

0.north-america.pool.ntp.org	Prefer	No Select	Delete
1.north-america.pool.ntp.org	Prefer	No Select	Delete
2.north-america.pool.ntp.org	Prefer	No Select	Delete
3.north-america.pool.ntp.org	Prefer	No Select	Delete

Add: Add

For best results three to five servers should be configured here.
The prefer option indicates that NTP should favor the use of this server more than all others.
The no select option indicates that NTP should not use this server for time, but stats for this server will be collected and displayed.

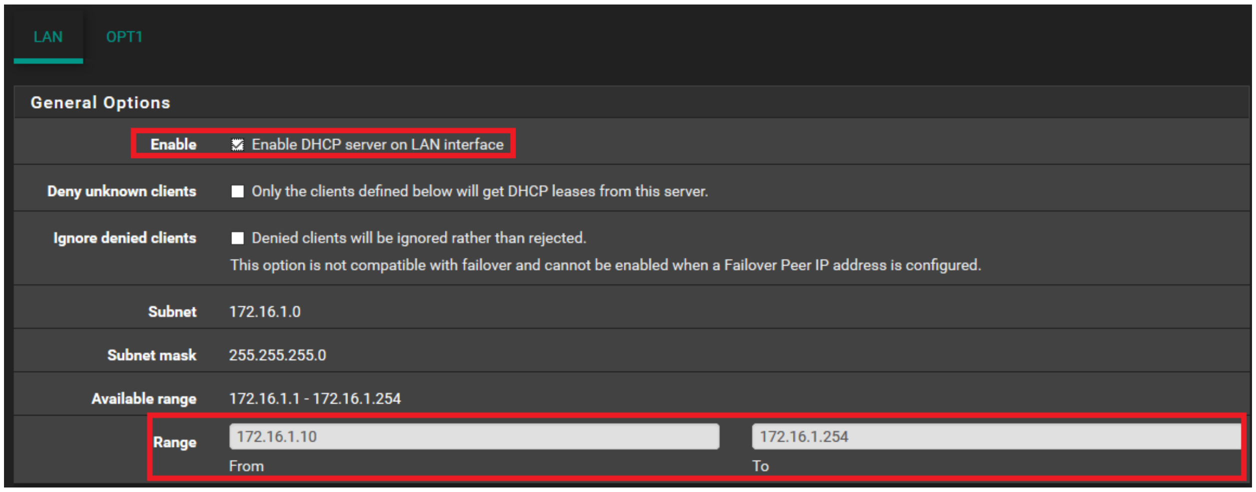
Orphan Mode:
Orphan mode allows the system clock to be used when no other clocks are available. The number here specifies the stratum reported during orphan mode and should normally be set to a number high enough to insure that any other servers available to clients are preferred over this server (default: 12).

NTP Graphs: Enable RRD graphs of NTP statistics (default: disabled).

Logging: Log peer messages (default: disabled).

DHCP

If you followed the instructions on how to configure the network interfaces of the pfSense VM in any of the previous hypervisor setup sections, then DHCP should be set up and enabled already. Systems on the network connected to the LAN interface should be served addresses from 172.16.1.10 to 172.16.1.254. Hosts on the OPT1 interface network should be served addresses from 172.16.2.10 to 172.16.2.254. While the CLI of pfSense lets you set the IP address scopes easily enough, the web interface allows a lot more granular control. DHCP configuration is available via the *Services > DHCP Server* page of the web UI.

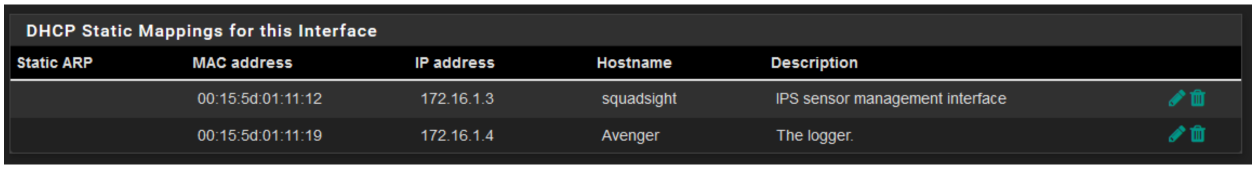


General Options

Enable	✓ Enable DHCP server on LAN interface
Deny unknown clients	Only the clients defined below will get DHCP leases from this server.
Ignore denied clients	Denied clients will be ignored rather than rejected. This option is not compatible with failover and cannot be enabled when a Failover Peer IP address is configured.
Subnet	172.16.1.0
Subnet mask	255.255.255.0
Available range	172.16.1.1 - 172.16.1.254
Range	From 172.16.1.10 To 172.16.1.254

As you can see in the screen capture above, there are separate tabs for the *LAN* and *OPT1* interfaces. Ensure the checkbox next to *Enable DHCP Server on the [interface name] interface* is checked (for both LAN and OPT1 interfaces), and confirm the corresponding IP address range for each interface is correct. By default, pfSense's DHCP server propagates the IP address set on the LAN and on the OPT1 interface as the default gateway as well as the DNS server for the corresponding network. For the purposes of our lab this is perfect, since pfSense handles the routing of the network traffic, has its DNS Resolver service enabled by default, and will use the DNS servers we configured during the initial setup for forwarding DNS requests.

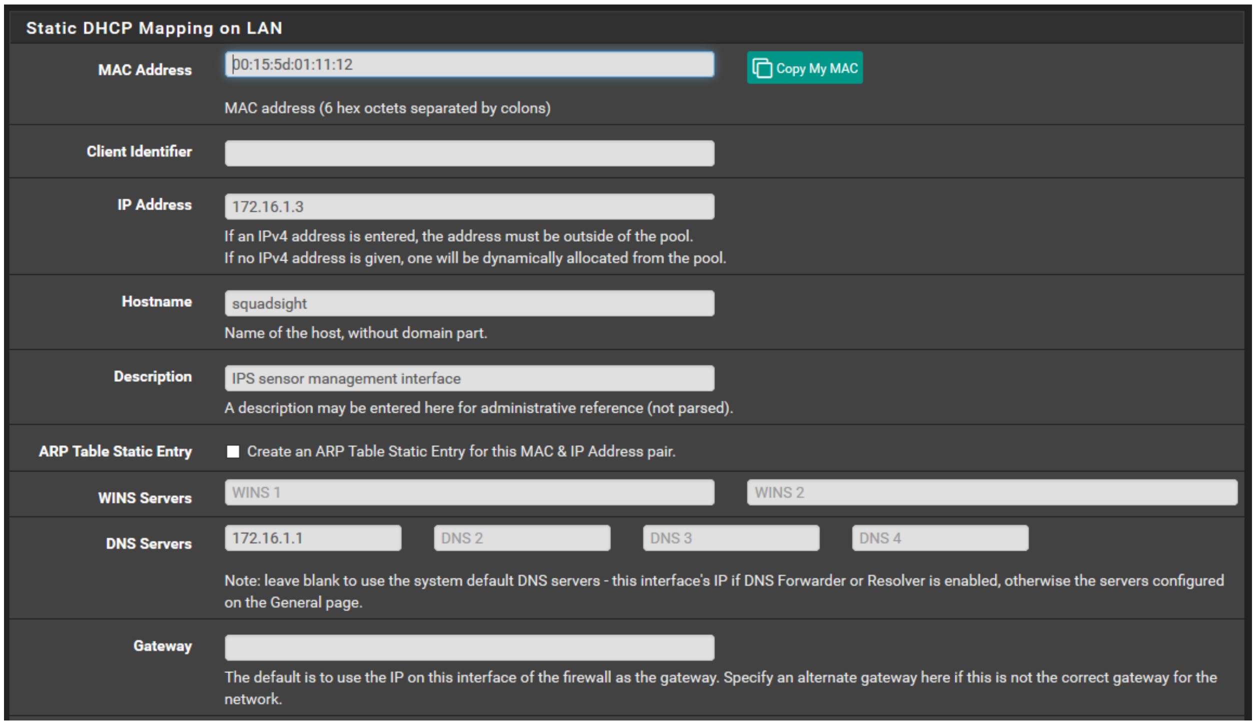
pfSense also allows us to configure static DHCP allocations for any network it is running DHCP for. Static mappings ensure that a virtual machine with a specific MAC address on its network card, connected to a particular network will ALWAYS get the same IP address from the DHCP server. The static mappings have to be for IP addresses from the network that the corresponding pfSense interface is in, but NOT for any address in the actual DHCP range. In the illustration above, the DHCP range is from 172.16.1.10 to 172.16.1.254. The addresses 172.16.1.1 and 172.16.1.2 are already in use by pfSense and our host system's network interfaces (if you are using a hosted hypervisor). That means we can only use the IP addresses from 172.16.1.3to 172.16.1.9 for static DHCP allocations, unless the DHCP range is adjusted.



DHCP Static Mappings for this Interface

Static ARP	MAC address	IP address	Hostname	Description
	00:15:5d:01:11:12	172.16.1.3	squadsight	IPS sensor management interface
	00:15:5d:01:11:19	172.16.1.4	Avenger	The logger.

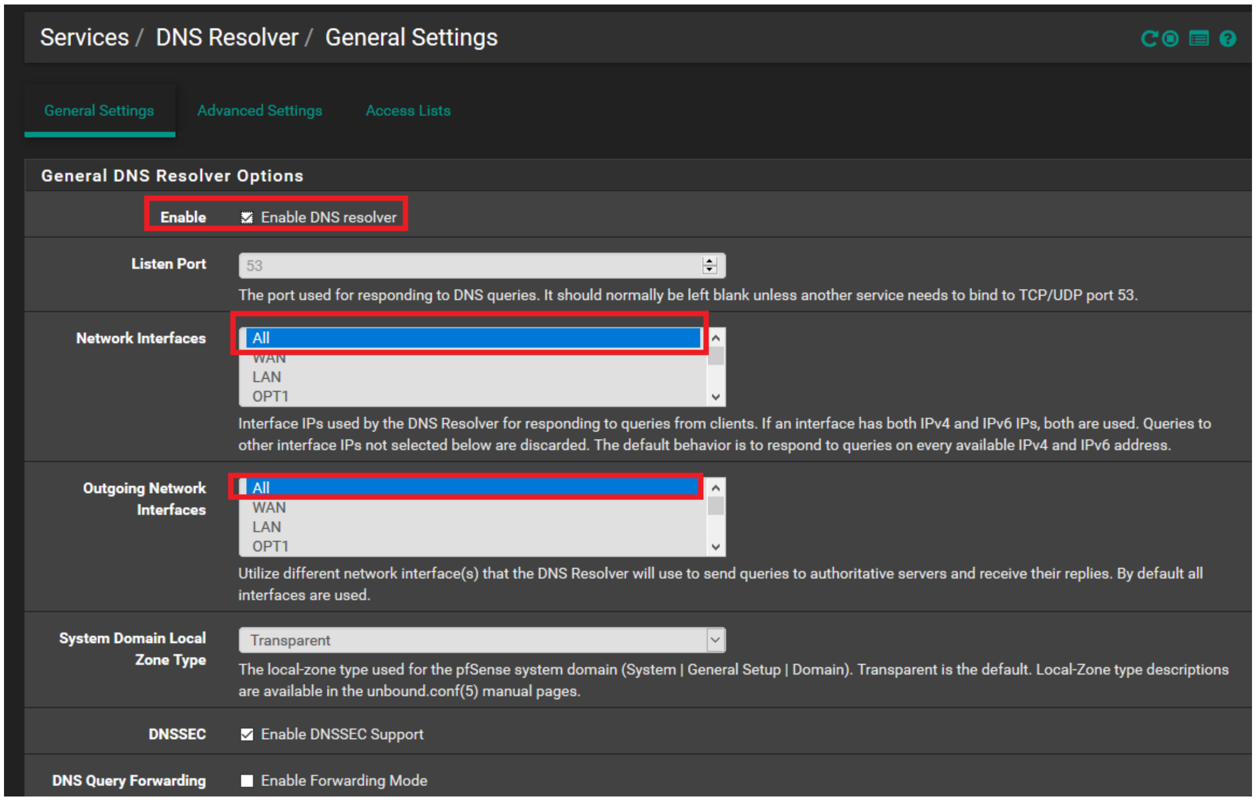
As an example, the following image shows the static DHCP mapping I am using for "Squadsight", the IPS VM's management interface, connected to the *Management* network. To create a static DHCP mapping, identify the MAC address of the network interface you want to create the mapping entry for, enter it into the *MAC Address* field (separated by colons [:]) choose the IP address you want the DHCP server to deliver that VM's network interface, enter it into the *IP Address* field and finally, save the settings. Note that the *DNS Servers* configuration is redundant, and you could get away with leaving the *DNS Servers* fields fields blank.


Static DHCP Mapping on LAN

MAC Address	00:15:5d:01:11:12
	MAC address (6 hex octets separated by colons)
Client Identifier	
IP Address	172.16.1.3
	If an IPv4 address is entered, the address must be outside of the pool. If no IPv4 address is given, one will be dynamically allocated from the pool.
Hostname	squadsight
	Name of the host, without domain part.
Description	IPS sensor management interface
	A description may be entered here for administrative reference (not parsed).
ARP Table Static Entry	☐ Create an ARP Table Static Entry for this MAC & IP Address pair.
WINS Servers	WINS 1, WINS 2
DNS Servers	172.16.1.1, DNS 2, DNS 3, DNS 4
	Note: leave blank to use the system default DNS servers - this interface's IP if DNS Forwarder or Resolver is enabled, otherwise the servers configured on the General page.
Gateway	
	The default is to use the IP on this interface of the firewall as the gateway. Specify an alternate gateway here if this is not the correct gateway for the network.

DNS Resolver

By default, pfSense has the DNS Resolver service enabled. To verify this, navigate to *Services > DNS Resolver*. Verify that the *Enable DNS resolver* checkbox is checked, and that the *All* option is selected for *Network Interfaces* field as well as for *Outgoing Network Interfaces* field.

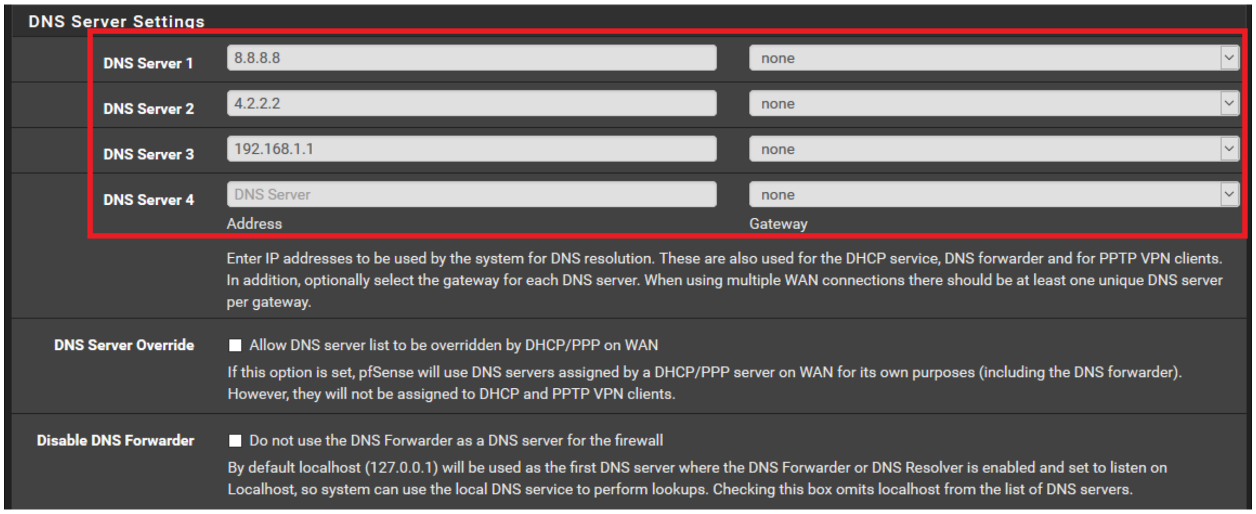


Services / DNS Resolver / General Settings

General DNS Resolver Options

Enable	✓ Enable DNS resolver
Listen Port	53 The port used for responding to DNS queries. It should normally be left blank unless another service needs to bind to TCP/UDP port 53.
Network Interfaces	All WAN LAN OPT1 Interface IPs used by the DNS Resolver for responding to queries from clients. If an interface has both IPv4 and IPv6 IPs, both are used. Queries to other interface IPs not selected below are discarded. The default behavior is to respond to queries on every available IPv4 and IPv6 address.
Outgoing Network Interfaces	All WAN LAN OPT1 Utilize different network interface(s) that the DNS Resolver will use to send queries to authoritative servers and receive their replies. By default all interfaces are used.
System Domain Local Zone Type	Transparent The local-zone type used for the pfSense system domain (System | General Setup | Domain). Transparent is the default. Local-Zone type descriptions are available in the unbound.conf(5) manual pages.
DNSSEC	✓ Enable DNSSEC Support
DNS Query Forwarding	Enable Forwarding Mode

To confirm which DNS servers pfSense will forward queries to if it can't resolve the domain name itself, navigate to System > General Setup, and view the DNS Server Settings. In my case, I'm using 8.8.8.8 (Google DNS), 4.2.2.2 (Level 3 DNS) and 192.168.1.1 (the default gateway/DNS for my physical network).


DNS Server Settings

	Address	Gateway
DNS Server 1	8.8.8.8	none
DNS Server 2	4.2.2.2	none
DNS Server 3	192.168.1.1	none
DNS Server 4	DNS Server	none

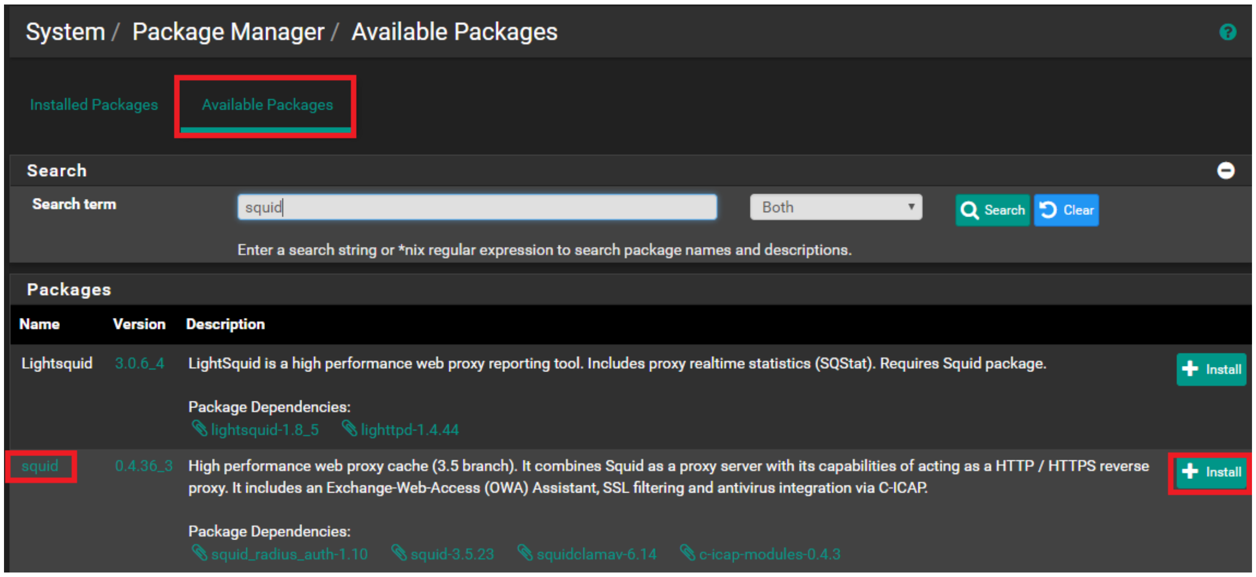
Enter IP addresses to be used by the system for DNS resolution. These are also used for the DHCP service, DNS forwarder and for PPTP VPN clients. In addition, optionally select the gateway for each DNS server. When using multiple WAN connections there should be at least one unique DNS server per gateway.

DNS Server Override: ☐ Allow DNS server list to be overridden by DHCP/PPP on WAN
If this option is set, pfSense will use DNS servers assigned by a DHCP/PPP server on WAN for its own purposes (including the DNS forwarder). However, they will not be assigned to DHCP and PPTP VPN clients.

Disable DNS Forwarder: ☐ Do not use the DNS Forwarder as a DNS server for the firewall
By default localhost (127.0.0.1) will be used as the first DNS server where the DNS Forwarder or DNS Resolver is enabled and set to listen on Localhost, so system can use the local DNS service to perform lookups. Checking this box omits localhost from the list of DNS servers.

Squid Proxy

pfSense supports utilizing the Squid HTTP proxy; all you need to do is install the package for it. Having a proxy server installed provides more granular control over the HTTP and FTP traffic allowed to flow into and out of the lab network. In addition to that, at a later date, we can choose to log the requests made, and have them sent to our SIEM VM to review event data. To install Squid, navigate to *System* > *Package Manager*, and click on *Available Packages*. In the search bar, type “squid”, and click the green *Install* button to the right of the *squid* package, then on the next page, click *Confirm*.

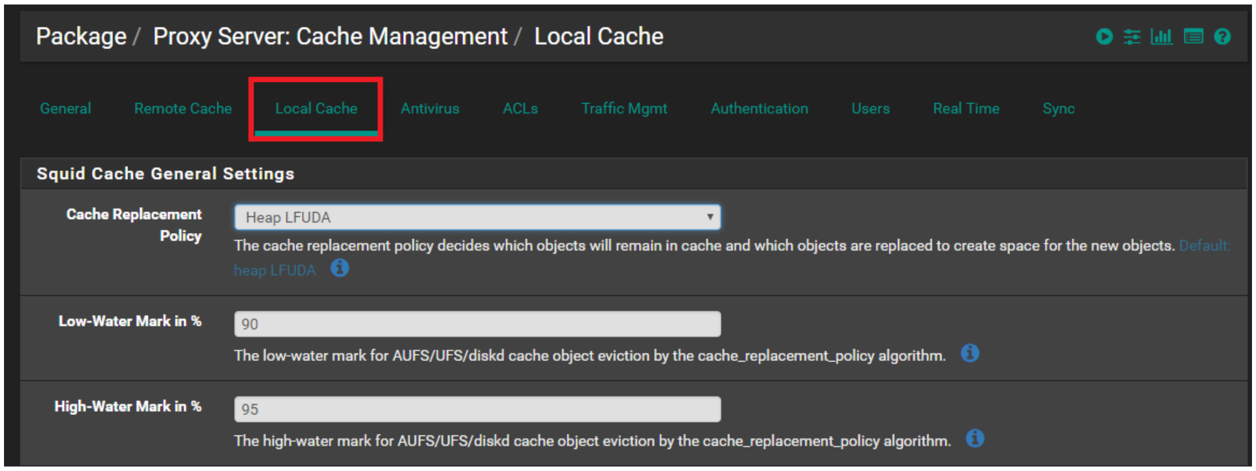


Enter a search string or *nix regular expression to search package names and descriptions.

Packages

Name	Version	Description
Lightsquid	3.0.6_4	LightSquid is a high performance web proxy reporting tool. Includes proxy realtime statistics (SQStat). Requires Squid package. Package Dependencies: lightsquid-1.8_5 lighttpd-1.4.44
squid	0.4.36_3	High performance web proxy cache (3.5 branch). It combines Squid as a proxy server with its capabilities of acting as a HTTP / HTTPS reverse proxy. It includes an Exchange-Web-Access (OWA) Assistant, SSL filtering and antivirus integration via C-ICAP. Package Dependencies: squid_radius_auth-1.10 squid-3.5.23 squidclamav-6.14 c-icap-modules-0.4.3

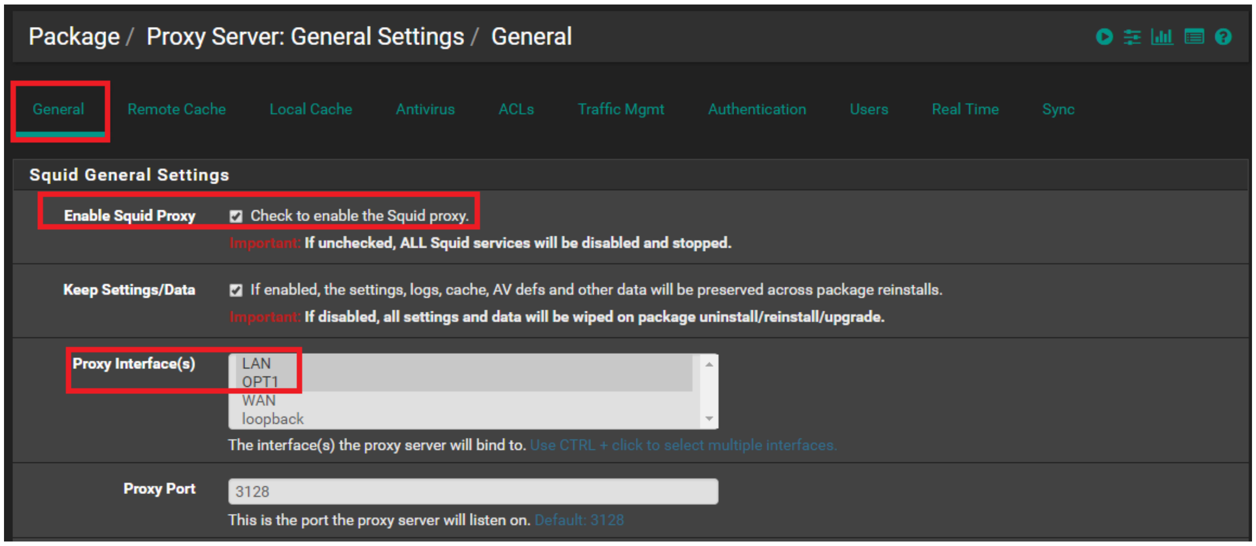
This should kick off the installer. Upon completion, the installer log in the middle of the screen will read *Success*. To confirm that Squid was installed properly, navigate to *Services*. If *Squid Proxy Server* is listed as an option there, the installation was successful. Proceed by clicking on the menu entry. On the page that appears, click on the tab labeled *Local Cache*. We won't be modifying any of the default settings, however, pfSense requires you to confirm the default local cache settings before you can actually start the Squid service. Simply scroll to the bottom of the page, and click the *Save* button.



Squid Cache General Settings

Cache Replacement Policy	Heap LFUDA The cache replacement policy decides which objects will remain in cache and which objects are replaced to create space for the new objects. Default: heap LFUDA
Low-Water Mark in %	90 The low-water mark for AUFS/UFS/diskd cache object eviction by the cache_replacement_policy algorithm.
High-Water Mark in %	95 The high-water mark for AUFS/UFS/diskd cache object eviction by the cache_replacement_policy algorithm.

Next, click on the *General* tab. Click to check the checkbox next to *Enable Squid Proxy*. In the *Proxy Interface(s)* pane, make sure that the entries for *LAN* and for *OPT1* are highlighted; this ensures that the proxy server is available on both of these networks. Afterwards, scroll to the bottom of the page, and click *Save*.


Package / Proxy Server: General Settings / General

Squid General Settings

Enable Squid Proxy ☑ Check to enable the Squid proxy.
Important: If unchecked, ALL Squid services will be disabled and stopped.

Keep Settings/Data ☑ If enabled, the settings, logs, cache, AV defs and other data will be preserved across package reinstalls.
Important: If disabled, all settings and data will be wiped on package uninstall/reinstall/upgrade.

Proxy Interface(s) LAN, OPT1, WAN, loopback
The interface(s) the proxy server will bind to. Use CTRL + click to select multiple interfaces.

Proxy Port 3128
This is the port the proxy server will listen on. Default: 3128

Note that with the firewall rules we have set up for the Management and the IPS network, we MUST use the proxy for all regular HTTP and FTP traffic. This makes using tools like `wget` or `apt-get` for setting up the other virtual machines a little bit more complicated. In order to work with command line applications that use HTTP or FTP for application updates in Linux/BSD environments, make sure that you familiarize yourself with the `http_proxy` variable and its use for Linux/BSD CLI tools. I found this great tutorial on how to set the `http_proxy` variable to ensure that this isn't a problem:
http://www.putorius.net/2012/09/how-to-set-httpproxy-variable-in-linux.html.

Note: The guide mentioned above assumes that the proxy is listening on port 8080/tcp, whereas our Squid proxy is listening on port 3128/tcp. Keep this in mind!

Defense in Depth for Windows Hosted Hypervisors

Using either VMware Workstation on Windows, VirtualBox on Windows or Microsoft Client Hyper-V to set up your lab environment, you created at least one virtual network that added a virtual network adapter to the Windows host operating system. Each hypervisor has a different name for this network card:

Client Hyper-V labels it "*vEthernet [vswitch name]*" (in our case, *vEthernet (Management)*)

VirtualBox calls this interface *VirtualBox Host Only-Network*

VMware Workstation identifies it as *VMware Network Adapter vmnetx* (where x is the vmnet or VM network the host is attached to; usually, the default host-only network is vmnet1)

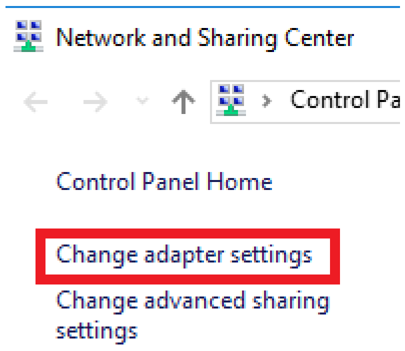
If you are running your VM lab on a laptop or on a device that constantly moves from one network to another, you may not always be in control of your physical system's IP address for accessing your virtual assets. Adding a virtual adapter for the *Management* network (the *LAN* interface in pfSense) to your Windows hypervisor host allows it to directly interact with the IPS and SIEM VMs, without having to worry about firewall rules in pfSense, or adding multiple network routes on the host system. As previously stated, to reach the Kali VM on the IPS network, you need a firewall rule on the LAN interface to allow SSH access to the IP address of this virtual machine. In addition to that, you need to add at least one static network route on the Windows host in order to tell it where to send the packets to interact with the Kali host.

While having this virtual network card directly attached to the *Management* virtual network is convenient, it also means that you are potentially exposing the host system to the devices on the Management network. Even with the pfSense firewall in place to enforce boundaries and segmentation between the various networks, taking additional security measures ensures we are not putting all of our eggs into one basket. The following sections will instruction you on how to unbind various Microsoft network protocols from the virtual network adapter, and how to implement a rule on the host's Windows Firewall that denies inbound connections on the virtual network adapter, but allows the virtual adapter to make connections to the virtual network as necessary.

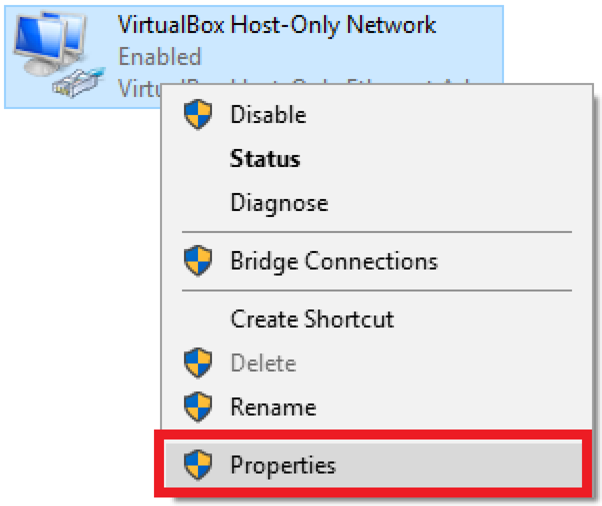
Unbinding Network Protocols on Windows Virtual Adapters

For the sake of simplicity, this tutorial is based on the *VirtualBox Host-Only Network* virtual network adapter. For the other hosted hypervisors discussed, you just need to substitute this description with *vEthernet(Management)* or *VMware Network Adapter vmnet1* as necessary. If there are any special steps or considerations that should be noted for VMware or Client Hyper-V users, I will advise on differences as we come across them.

To begin, navigate to the *Network and Sharing Center*, and click the *Change adapter Settings* option on the left of the screen. Alternatively you can head directly to "View Network Connections".



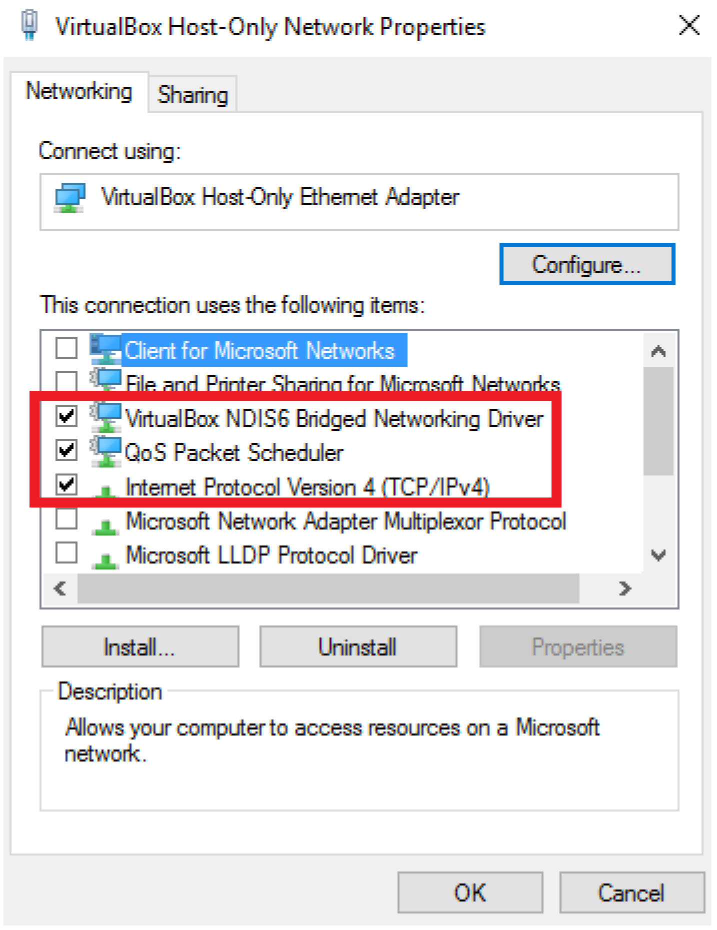
Find the adapter labeled *VirtualBox Host-Only Network*, right click it, and select *Properties*.



On the *Networking* tab of the Properties window, under the pane entitled “This connection uses the following items:”, uncheck everything except for:

- VirtualBox NDIS6 Bridged Networking Driver
- QoS Packet Scheduler
- Internet Protocol Version 4 (TCP/IPv4)

You may need to scroll down to disable all of the remaining items.


VirtualBox Host-Only Network Properties

Networking | Sharing

Connect using:

VirtualBox Host-Only Ethernet Adapter

Configure...

This connection uses the following items:

- [ ] Client for Microsoft Networks
- [ ] File and Printer Sharing for Microsoft Networks
- [✓] VirtualBox NDIS6 Bridged Networking Driver
- [✓] QoS Packet Scheduler
- [✓] Internet Protocol Version 4 (TCP/IPv4)
- [ ] Microsoft Network Adapter Multiplexor Protocol
- [ ] Microsoft LLDP Protocol Driver

Install... | Uninstall | Properties

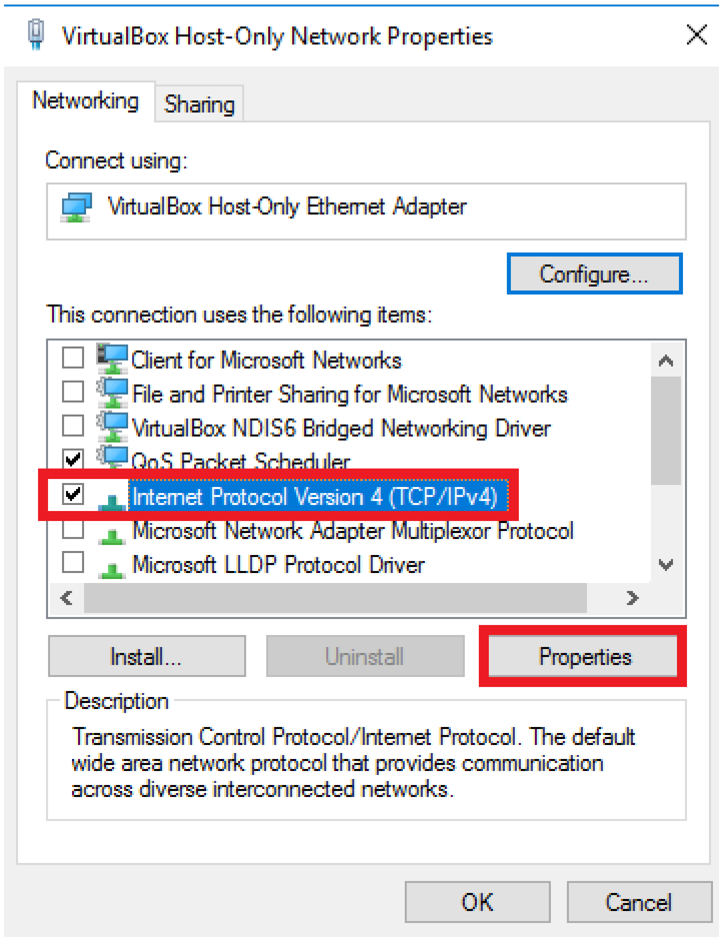
Description

Allows your computer to access resources on a Microsoft network.

OK | Cancel

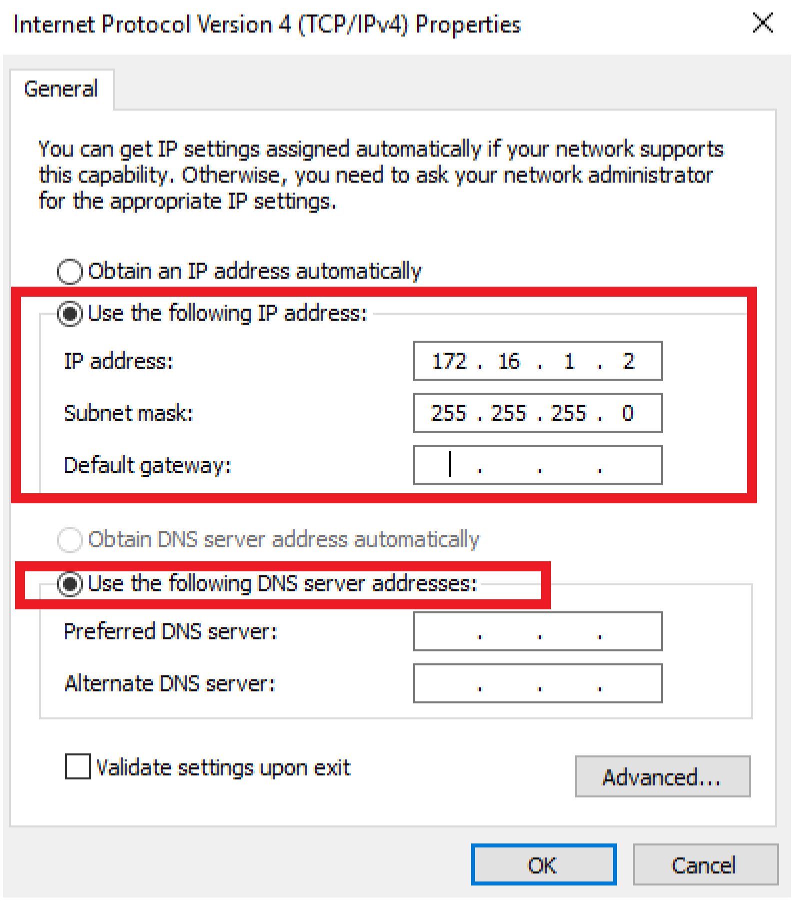
Note for Client Hyper-V and VMware Workstation Hypervisors: Leave only *QoS Packet Scheduler* and *Internet Protocol Version 4 (TCP/IPv4)* enabled. Every other item listed here should be set to disabled.

Next, click to select *Internet Protocol Version 4 (TCP/IPv4)*, then click on the *Properties* button below the pane.


VirtualBox Host-Only Network Properties
Networking
Sharing
Connect using:
VirtualBox Host-Only Ethernet Adapter
Configure...
This connection uses the following items:
Client for Microsoft Networks
File and Printer Sharing for Microsoft Networks
VirtualBox NDIS6 Bridged Networking Driver
QoS Packet Scheduler
Internet Protocol Version 4 (TCP/IPv4)
Microsoft Network Adapter Multiplexor Protocol
Microsoft LLDP Protocol Driver
Install...
Uninstall
Properties
Description
OK
Cancel

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

First thing we need to do is to assign this network adapter a static IP address, so it can access resources in the host-only network, which represents our *Management* network. For my lab, I used the network 172.16.1.0/24, and configured the host-only adapter to have the IP address 172.16.1.2, the subnet mask of 255.255.255.0, and **NO default gateway**. You can input these settings after clicking the radio button next to *Use the following IP address:*. When finished, click the “Use the following DNS servers:” radio button, and leave the input boxes **blank**.



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

Obtain an IP address automatically [✓ Use the following IP address:]

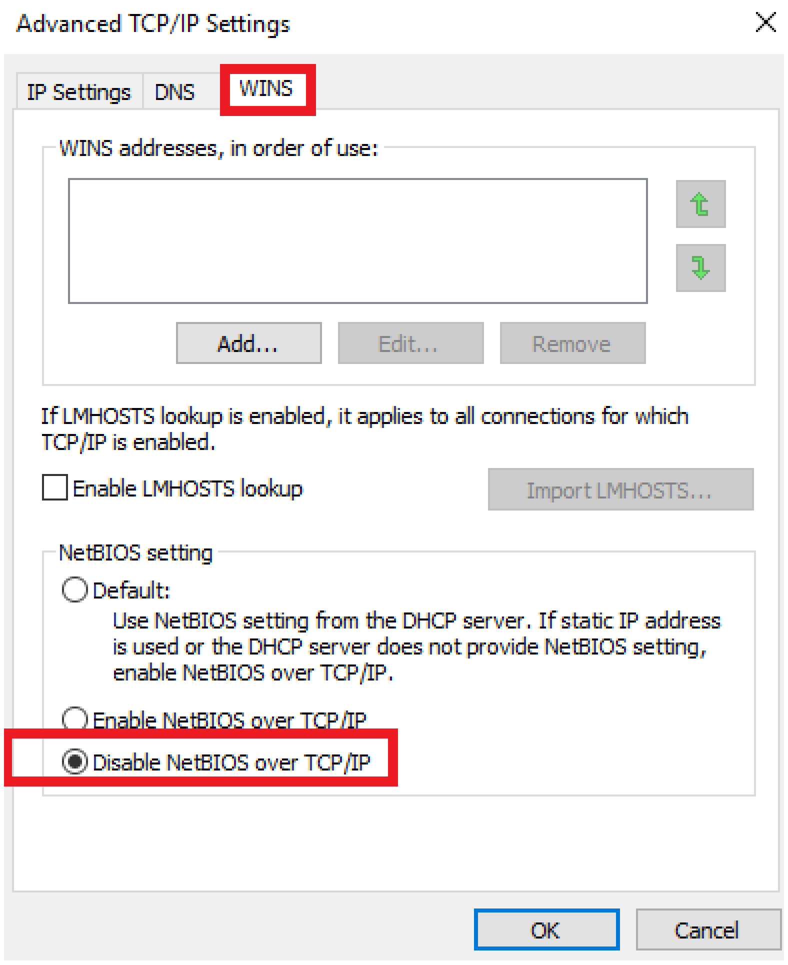
IP address:	172 . 16 . 1 . 2
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	. . .

Obtain DNS server address automatically [✓ Use the following DNS server addresses:]

Preferred DNS server:	. . .
Alternate DNS server:	. . .

☐ Validate settings upon exit

Next, Click the *Advanced…* button. In the Advanced TCP/IP Settings window, switch to the *WINS* tab. At the bottom, in the section captioned *NetBIOS setting*, click the *Disable NetBIOS over TCP/IP* radio button.


Advanced TCP/IP Settings

If LMHOSTS lookup is enabled, it applies to all connections for which TCP/IP is enabled.

[ ] Enable LMHOSTS lookup

NetBIOS setting

○ Default:
Use NetBIOS setting from the DHCP server. If static IP address is used or the DHCP server does not provide NetBIOS setting, enable NetBIOS over TCP/IP.

○ Enable NetBIOS over TCP/IP

● Disable NetBIOS over TCP/IP

To finish, first, click OK to close the Advanced TCP/IP Settings window, next, click OK to close the Internet Protocol Version 4 (TCP/IPv4) Properties window, and finally, click OK to close the VirtualBox Host-Only Network Properties window.

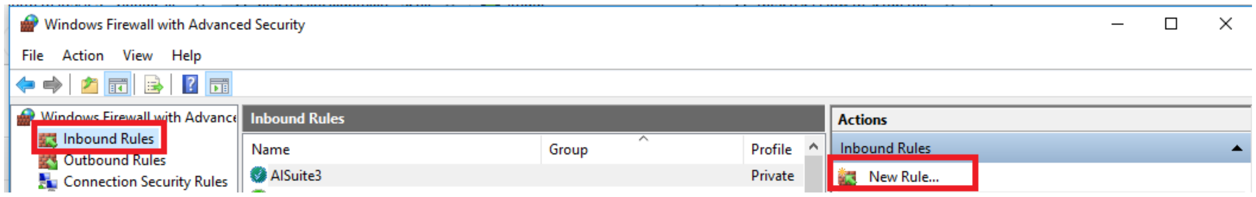
After you are done unbinding the nonessential network protocols from this virtual adapter, I would very highly suggest enabling and configuring Windows Firewall to deny inbound connections to this network card's IP address.

Using Windows Firewall to Limit Exposure of Windows Hypervisor Hosts

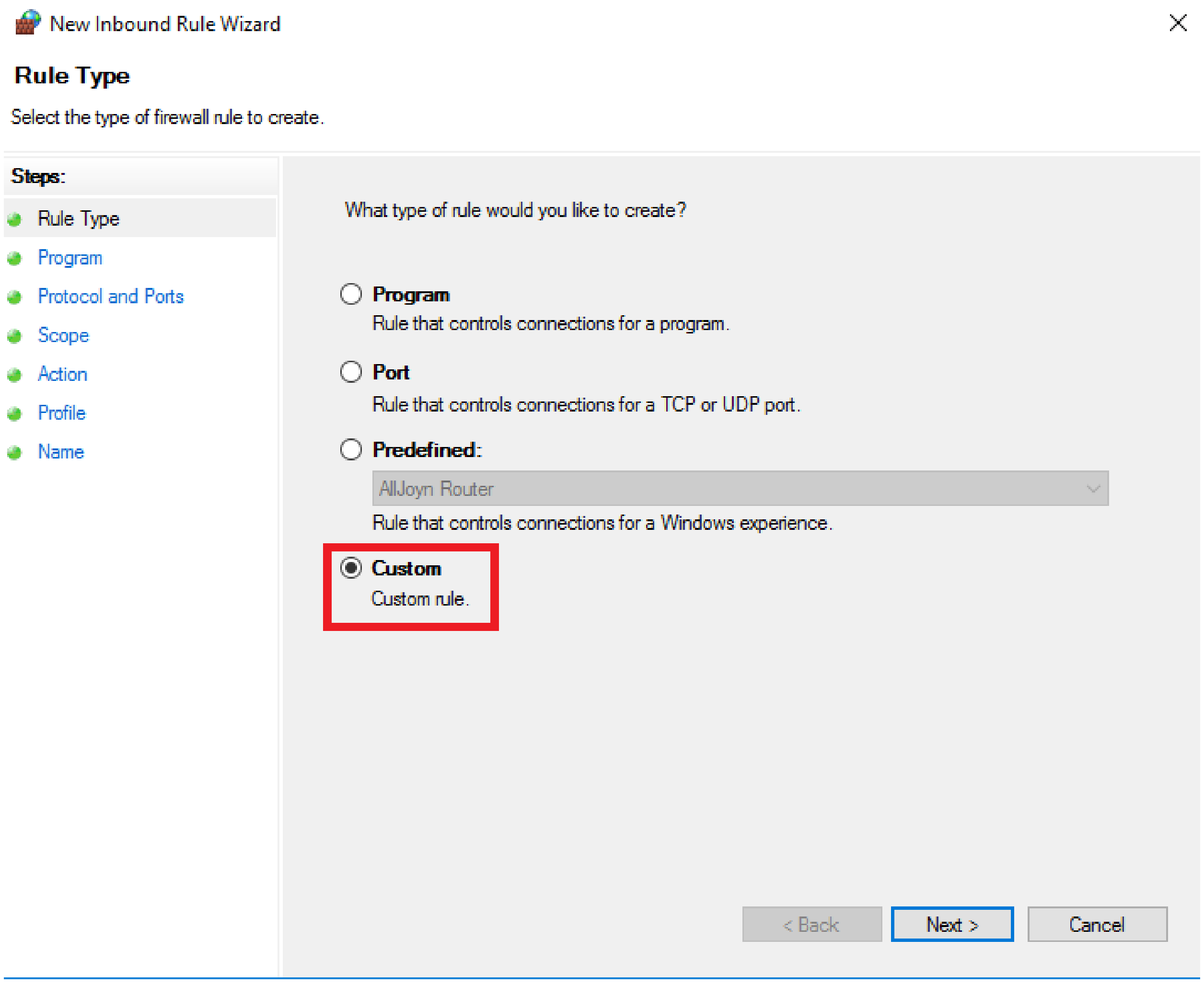
As previously established, we are using the OS of the system the hosted hypervisor is installed on to manage our virtual machines. In my case, I'm using Windows as the operating system to host my hypervisor on. This section outlines how to create rules for the built-in firewall. In spite of whatever misgivings you might have, Windows Firewall is actually nice and easy to use; it serves our purposes perfectly. **The aim of this guide is to specifically protect Windows systems hosting VMs via a hosted hypervisor that will be used to interact with systems in the virtual network.**

This guide will teach you how to create a firewall rule that will deny any and all traffic inbound towards the IP address we allocated to the virtual NIC (in our case, 172.16.1.2). While the pfSense firewall SHOULD be doing its job enforcing security boundaries, defense in depth means being paranoid. When set up, this Windows Firewall rule ensures that hosts on the virtual networks cannot initiate ANY connections to 172.16.1.2, but 172.16.1.2 is allowed to initiate and keep established connections to hosts as necessary.

Let's begin by opening *Windows Firewall with Advanced Security*, selecting *Inbound Rules*, and then clicking on *New Rule...* to launch the *New Inbound Rule Wizard*.



On the first page of the wizard, captioned *Rule Type*, select the “*Custom*” radio button, then click *Next*.



New Inbound Rule Wizard

Rule Type

Select the type of firewall rule to create.

Steps:

- Rule Type
- Program
- Protocol and Ports
- Scope
- Action
- Profile
- Name

What type of rule would you like to create?

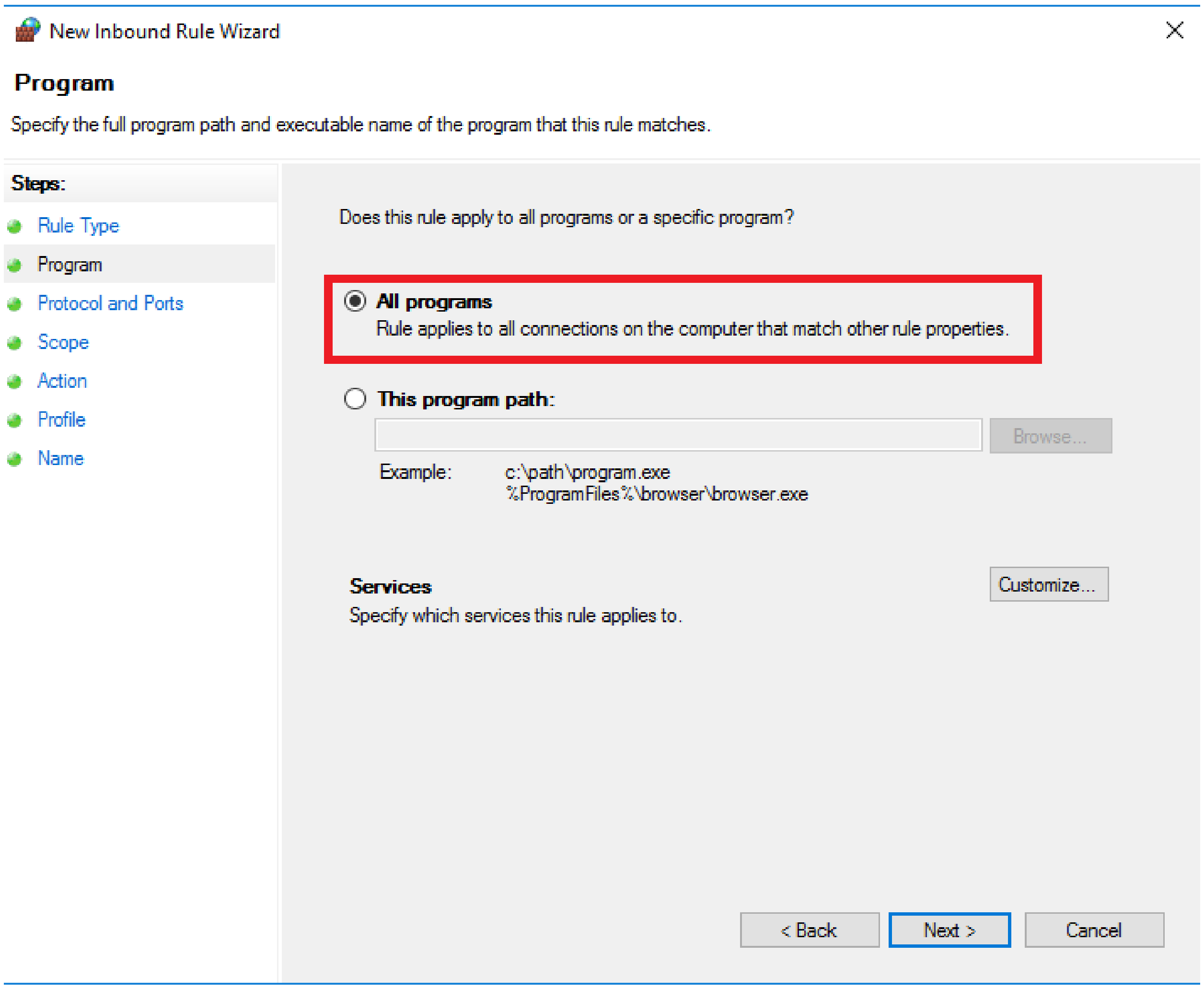
( ) **Program**
Rule that controls connections for a program.

( ) **Port**
Rule that controls connections for a TCP or UDP port.

( ) **Predefined:**
AllJoyn Router
Rule that controls connections for a Windows experience.

(●) **Custom**
Custom rule.

On the next page, click the *All Programs* radio button, then click *Next*.



New Inbound Rule Wizard

Program

Specify the full program path and executable name of the program that this rule matches.

Does this rule apply to all programs or a specific program?

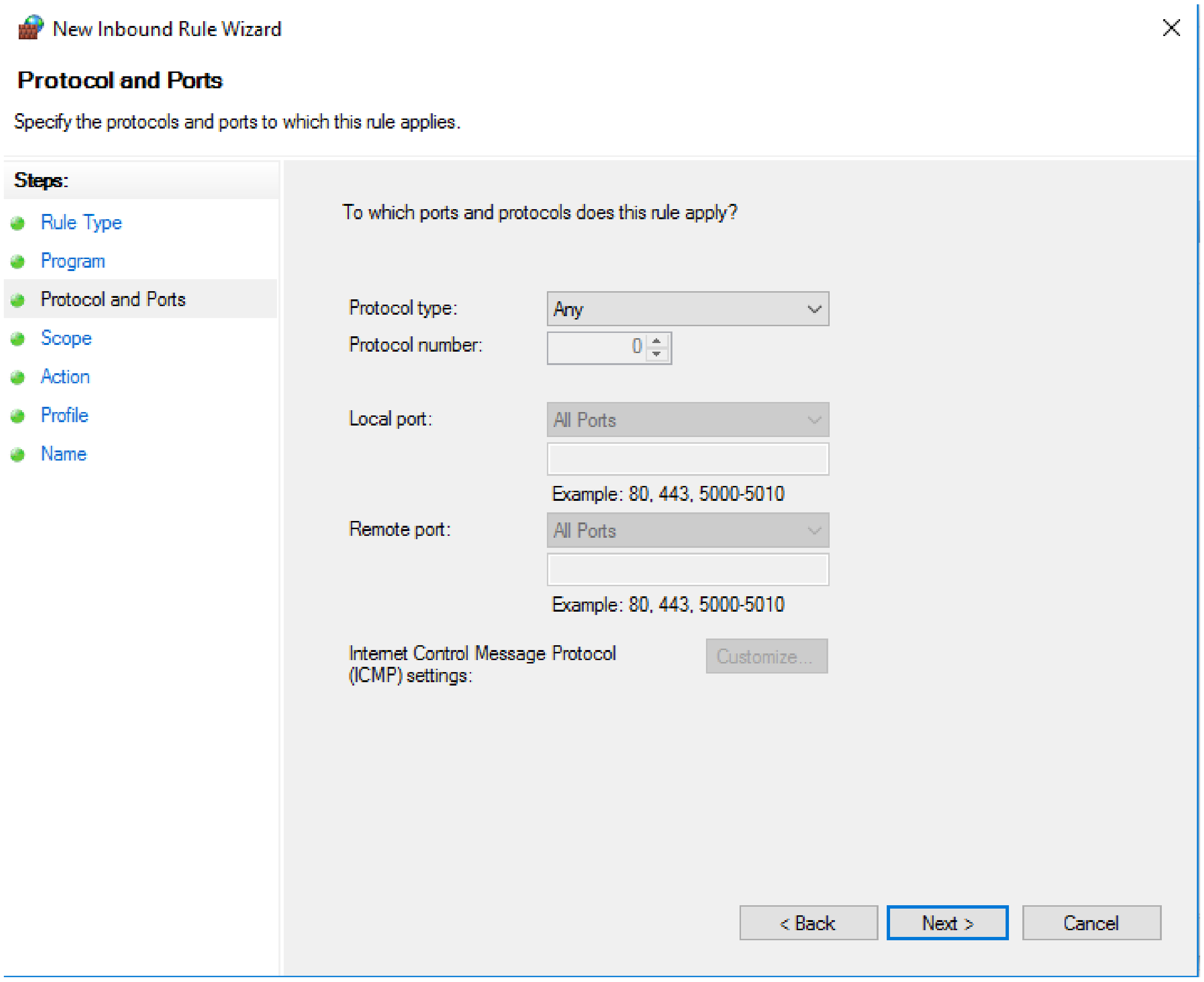
(●) **All programs**
Rule applies to all connections on the computer that match other rule properties.

( ) **This program path:**

Example: c:\path\program.exe
%ProgramFiles%\browser\browser.exe

Services
Specify which services this rule applies to.

On the following page, as we want this rule to apply to ANY protocol, accept the default settings by clicking *Next*.



New Inbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

To which ports and protocols does this rule apply?

Protocol type: Any

Protocol number: 0

Local port: All Ports

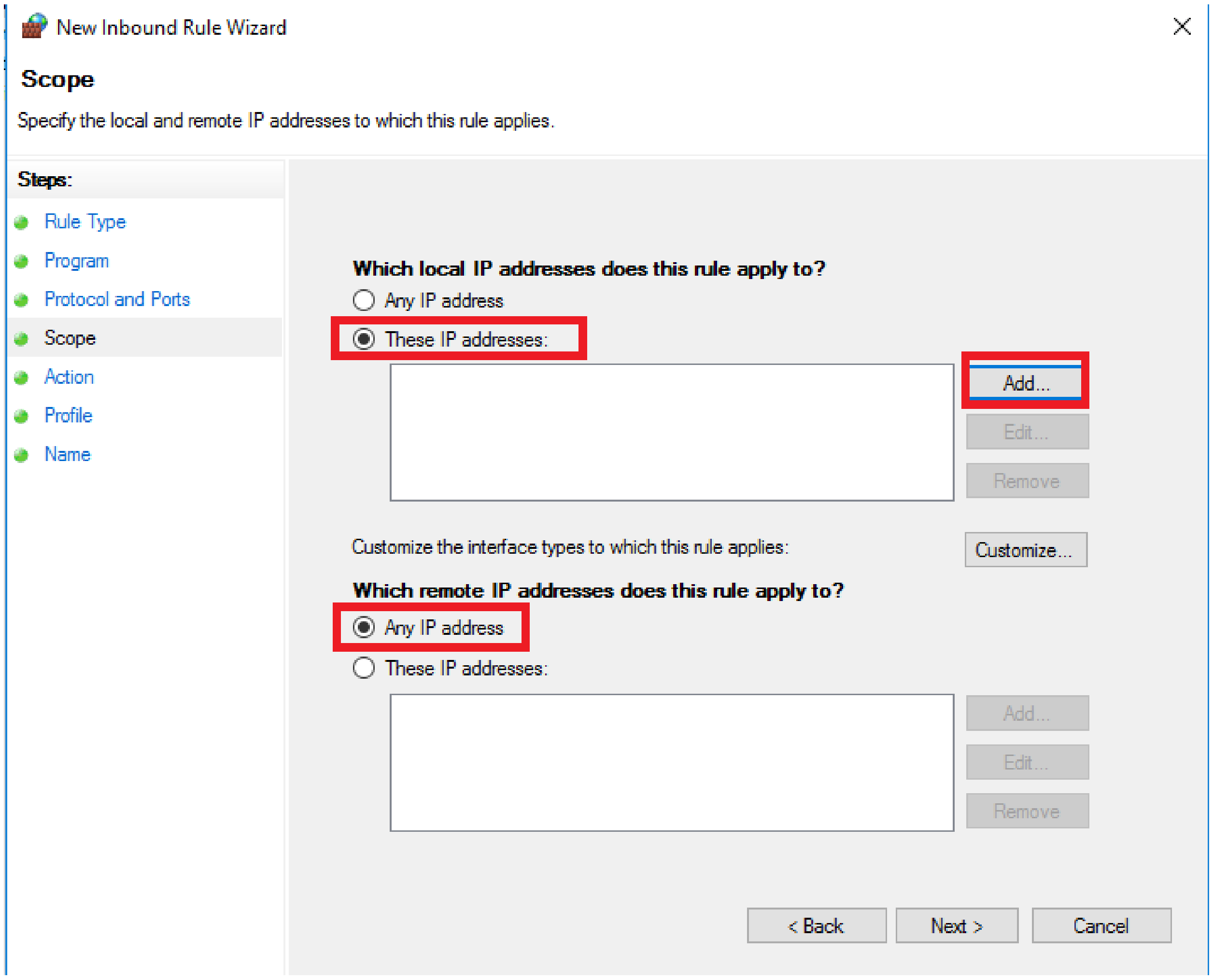
Example: 80, 443, 5000-5010

Remote port: All Ports

Example: 80, 443, 5000-5010

Internet Control Message Protocol (ICMP) settings:

On the page captioned *Scope*, in the *Which remote IP addresses does this apply to?* section in lower part, ensure the *Any IP address* radio button is clicked. Above this, in the part headlined *Which local IP addresses does this apply to?*, select the radio button next to *These IP addresses:*, then click the *Add...* button on the right.


New Inbound Rule Wizard

Scope

Specify the local and remote IP addresses to which this rule applies.

Steps:

- Rule Type
- Program
- Protocol and Ports
- Scope
- Action
- Profile
- Name

Which local IP addresses does this rule apply to?

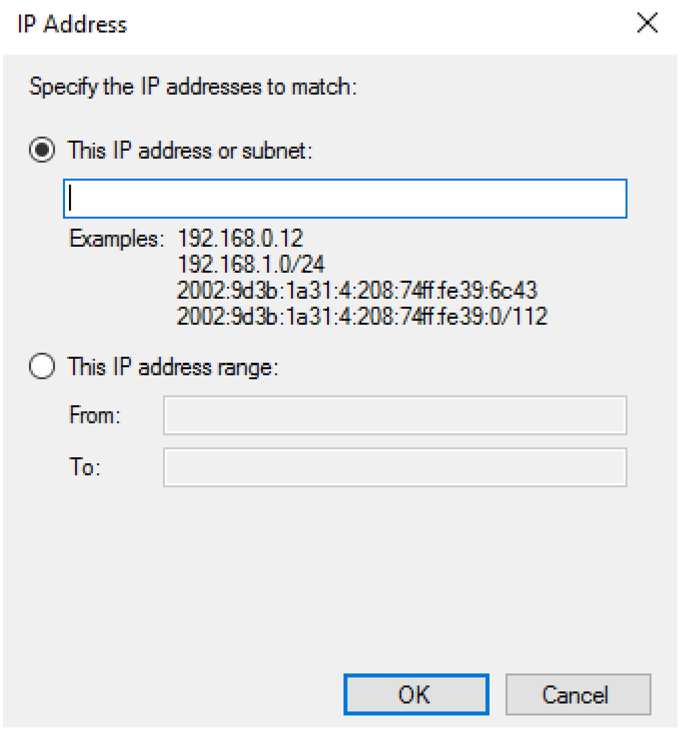
( ) Any IP address
(●) These IP addresses:

Customize the interface types to which this rule applies:

Which remote IP addresses does this rule apply to?

(●) Any IP address
( ) These IP addresses:

We want this rule to apply to the address of the virtual network interface that is connected to the *Management* network. In our case, we have chosen is 172.16.1.2. Enter this value in the box below *This IP address or subnet*. Click OK to close the IP Address window, then click *Next* at the bottom of the *New Inbound Rule Wizard* window.


IP Address

Specify the IP addresses to match:

(●) This IP address or subnet:

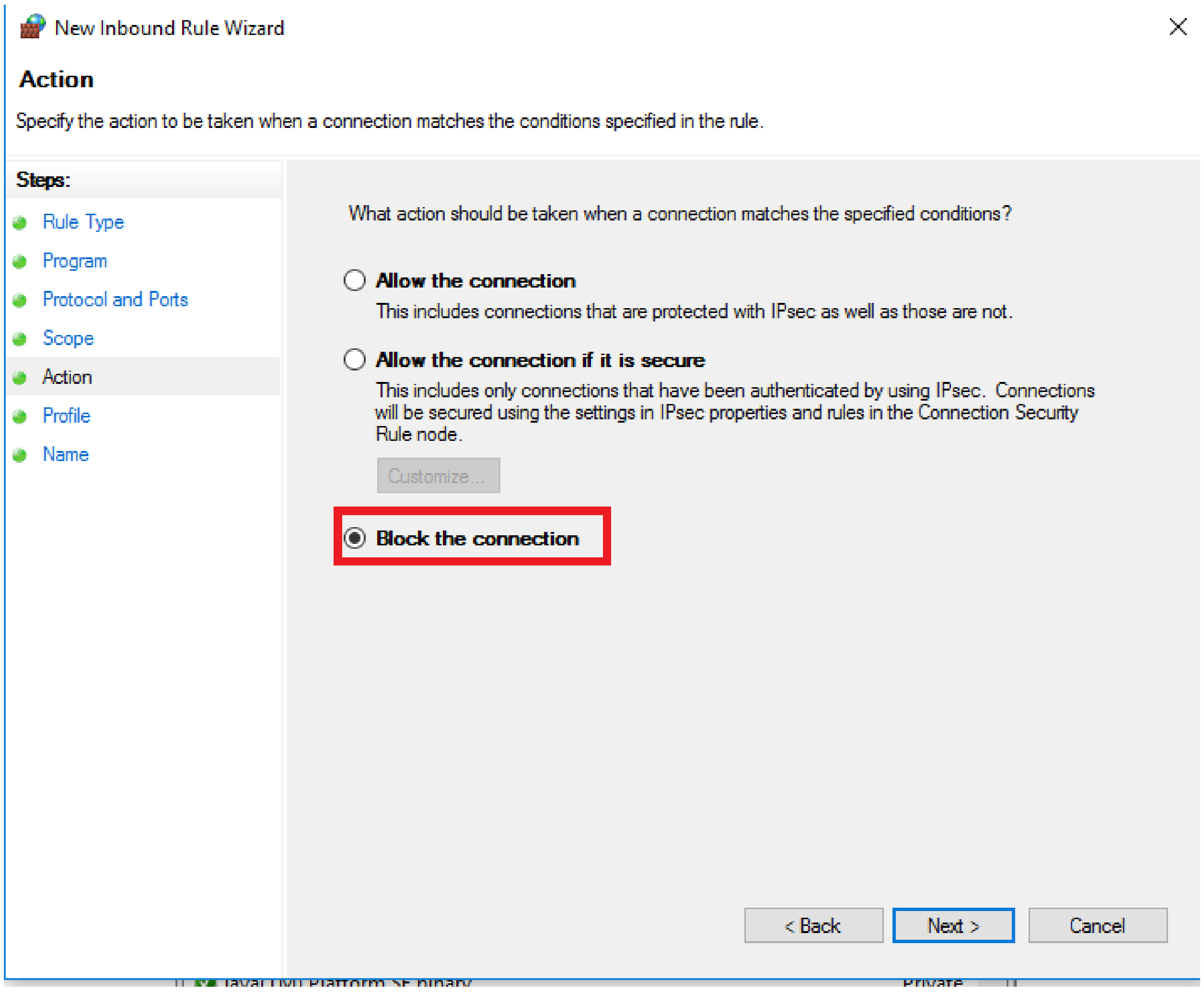
Examples: 192.168.0.12
192.168.1.0/24
2002:9d3b:1a31:4:208:74ff:fe39:6c43
2002:9d3b:1a31:4:208:74ff:fe39:0/112

( ) This IP address range:

From:

To:

On the Action page, click the “Block the connection” radio button, and again, click *Next*.



New Inbound Rule Wizard

Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

Steps:
- Rule Type
- Program
- Protocol and Ports
- Scope
- Action
- Profile
- Name

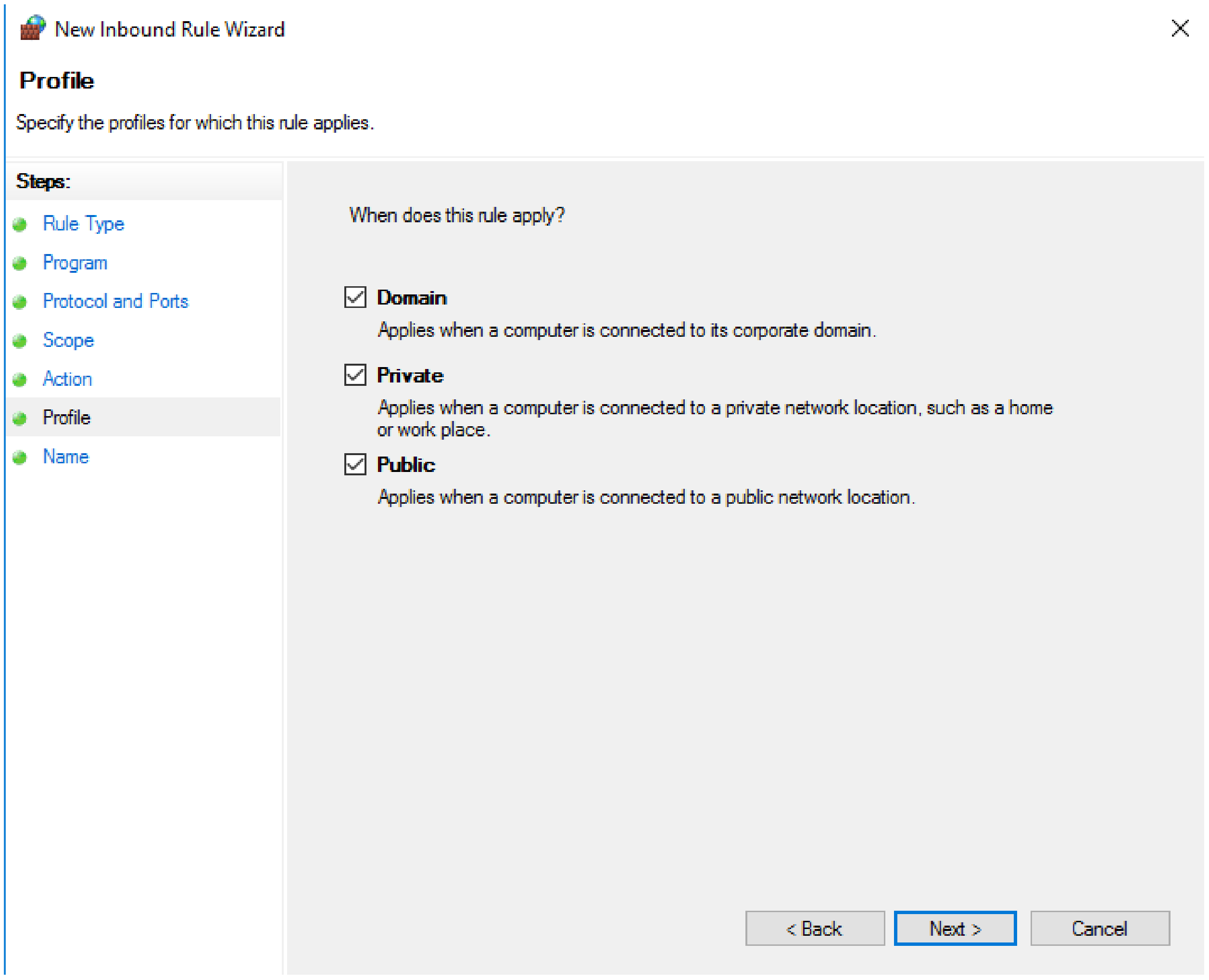
What action should be taken when a connection matches the specified conditions?

( ) **Allow the connection**
This includes connections that are protected with IPsec as well as those are not.

( ) **Allow the connection if it is secure**
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

[✓ **Block the connection**]

Next up is the page with the available network profiles. Make sure that all three checkboxes (*Domain, Private, Public*) are all checked, then click *Next*.


New Inbound Rule Wizard

Profile

Specify the profiles for which this rule applies.

Steps:
- Rule Type
- Program
- Protocol and Ports
- Scope
- Action
- Profile
- Name

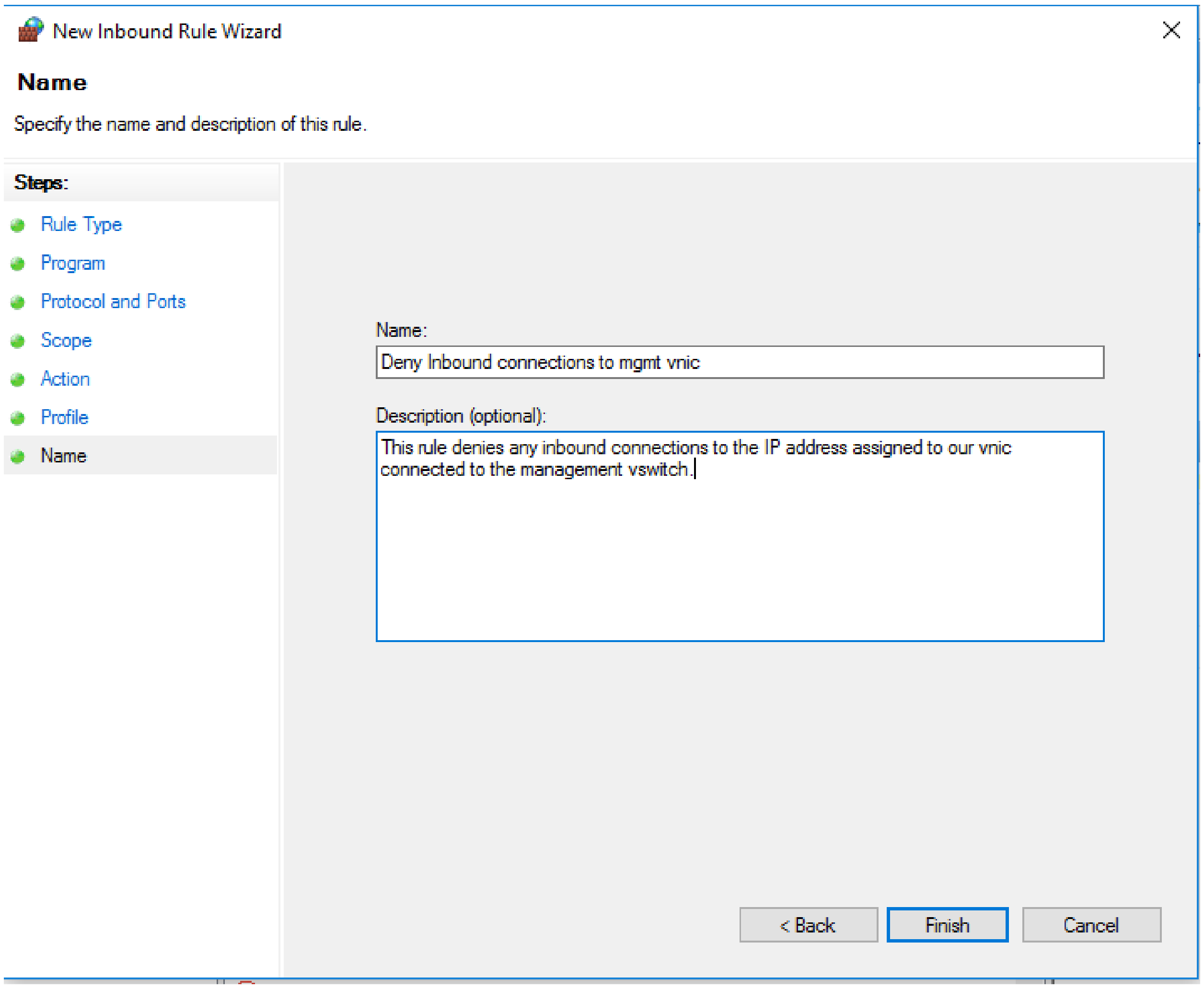
When does this rule apply?

☑ **Domain**
Applies when a computer is connected to its corporate domain.

☑ **Private**
Applies when a computer is connected to a private network location, such as a home or work place.

☑ **Public**
Applies when a computer is connected to a public network location.

On the final page of the wizard, give your firewall rule a name, and add a description that reminds you why the rule exists, and click *Finish*. Make sure to document that this firewall rule exists.



Name:

Deny Inbound connections to mgmt vnic

Description (optional):

This rule denies any inbound connections to the IP address assigned to our vnic connected to the management vswitch.

Automated Patching for Linux Lab VMs

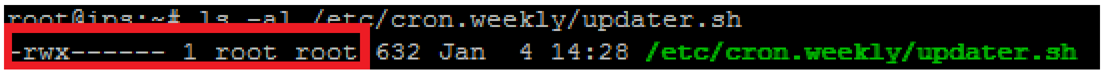
When you created the lab VMs that are based on Ubuntu Linux, you were given the option to have the OS manage security updates for you. What if you want your Linux systems to automatically update ALL of their packages regularly? Proper maintenance and patching of your lab environment on a regular basis is VERY important. Because of the seriousness of this task, I created a simple script and process to follow in order to automate this task.

`updater.sh`

Both Kali and Ubuntu Linux use the `apt` package manager to manage the software packages; this includes software updates. Below is a simple script that utilizes the `apt-get` command to check for, automatically download, and install the latest updates available, then reboot the system:

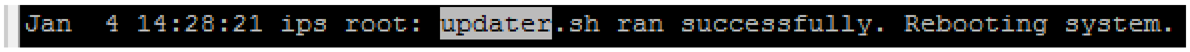
```
#!/bin/bash
#updater.sh - Weekly update script
#This script checks for the latest updates, downloads them, then reboots the
system.
#Place this script in /etc/cron.weekly, ensure it is owned by root (chown
root:root /etc/cron.weekly/updater.sh)
#Ensure the script has at least user execute permissions (chmod 700
/etc/cron.weekly/updater.sh)
#If you want updates to be ran once daily or weekly, simply place this script
into /etc/cron.daily or /etc/cron.monthly as you see fit.
export DEBIAN_FRONTEND=noninteractive
apt-get -q update
apt-get -y -q dist-upgrade
logger updater.sh ran successfully. Rebooting system.
init 6
exit 0
```

As the commented lines of this script read, place this script into `/etc/cron.daily`, `weekly`, or `monthly`, depending on whether you want your system to auto-update on a daily, weekly, or monthly basis. User root needs to be creator/owner of the script (`chown root:root`), and it must have at least the file permissions use and execute (`chmod u+x`, or `chmod 700`) in order to run successfully.



```
root@ips:~# ls -al /etc/cron.weekly/updater.sh
-rwx------ 1 root root 632 Jan  4 14:28 /etc/cron.weekly/updater.sh
```

When the script finishes, the system WILL be rebooted. For this reason, I would NOT, unless it has been agreed upon, deploy this script to a production system that runs anything of any importance. It is designed to keep your lab environment secured and up to date, and only your lab environment. The script will also write a line to `/var/log/syslog` via the `logger` command, to let you know that and when it was run.



```
Jan  4 14:28:21 ips root: updater.sh ran successfully. Rebooting system.
```

You can use this script to keep the Kali, SIEM, and IPS VMs automatically updated, as well as any new VM you decide to add, as long as it uses .deb packages and the `apt` package manager. Hypothetically, the script can also be modified to use the equivalent update commands for `yum` (the package manager for redhat/RPM-based distros) or other package managers as well.

Remote Lab Management

If you're interested in configuring remote management for your lab, read on. The following guides will assist you in setting up SSH access for your lab assets. There are sections dedicated to configuring remote access on Windows hosted hypervisors, on OS X/Unix/Linux hosted hypervisors, as well as on how to manage remote access to bare-metal hypervisors for either Windows or Unix/Linux users.

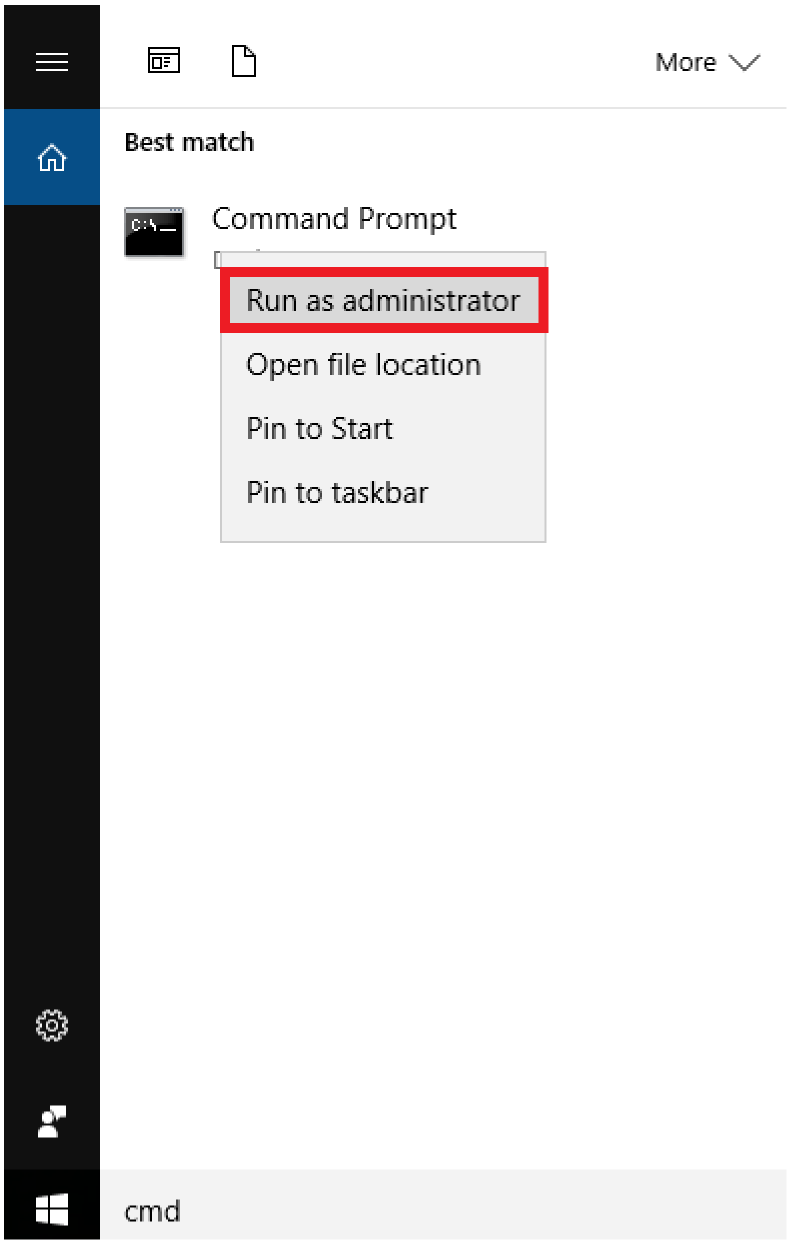
Windows Remote Access

The following are a list of remote access solutions and related configurations aiming to allow users of a Windows hosted hypervisor to remotely administer and interact with their lab systems.

Persistent Static Routes

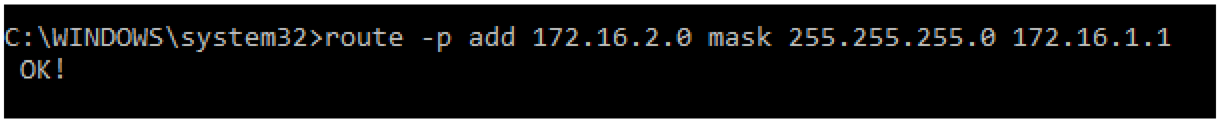
Recall that all of our hosted hypervisors should have a virtual network card attached to the *Management* virtual network. This means that establishing network connections to the *IPS* and *SIEM* virtual machines should be easy, since the virtual network adapter and the two VMs are on the same local network. However, if we want our hypervisor host to communicate with hosts on the IPS network, such as the Kali VM, we need to tell our Windows host where to forward its traffic in order to reach the network (172.16.2.0/24) the Kali Linux VM resides in. Enter the `route` command. `route` allows you to edit the network routing table of the Windows host. We're going to create a persistent route to the 172.16.2.0 network.

Navigate to the *Command Prompt* application in the Start Menu, or find it with the Windows Search. Right click on the icon/link for the program, and choose the option “Run as administrator” from the menu.



In the Command Prompt session, execute the following command:

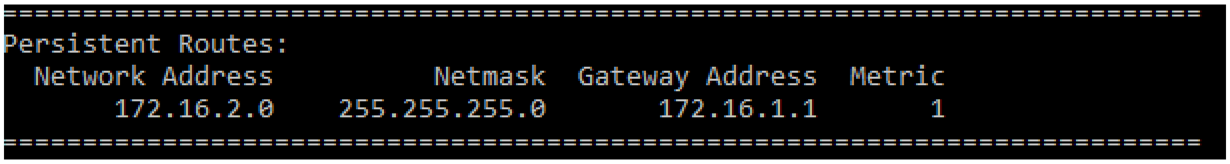
```
route -p add 172.16.2.0 mask 255.255.255.0 172.16.1.1
```



```
C:\WINDOWS\system32>route -p add 172.16.2.0 mask 255.255.255.0 172.16.1.1
 OK!
```

This command instructs Windows to route network traffic destined for 172.16.2.0/24 (the *IPS* network) through 172.16.1.1 (which is pfSense's gateway interface for the *Management* network). The `-p` option makes this route persistent. By default, when the Windows system reboots, any routes added manually without that option are lost. If you want to confirm you entered the command correctly, run the following in the Command Prompt session:

```
route print
```



```
===========================================================================
Persistent Routes:
  Network Address          Netmask  Gateway Address  Metric
       172.16.2.0    255.255.255.0       172.16.1.1       1
===========================================================================
```

You'll notice that the output returned by the command has a section called *Persistent Routes*, and that the route we just added is listed there. You should be able now to access hosts on the 172.16.2.0 network. Please bear in mind, that while you may have a static route to the `172.16.2.0/24` network in the routing table, that in order to establish a connection to systems on that network, you must also have to have firewalls rules configured on pfSense VM to allow traffic from the *Management* network to the *LAN* network, and that the system you want to connect to must also be running the network service you want to interact with.

Windows SSH and SCP Software

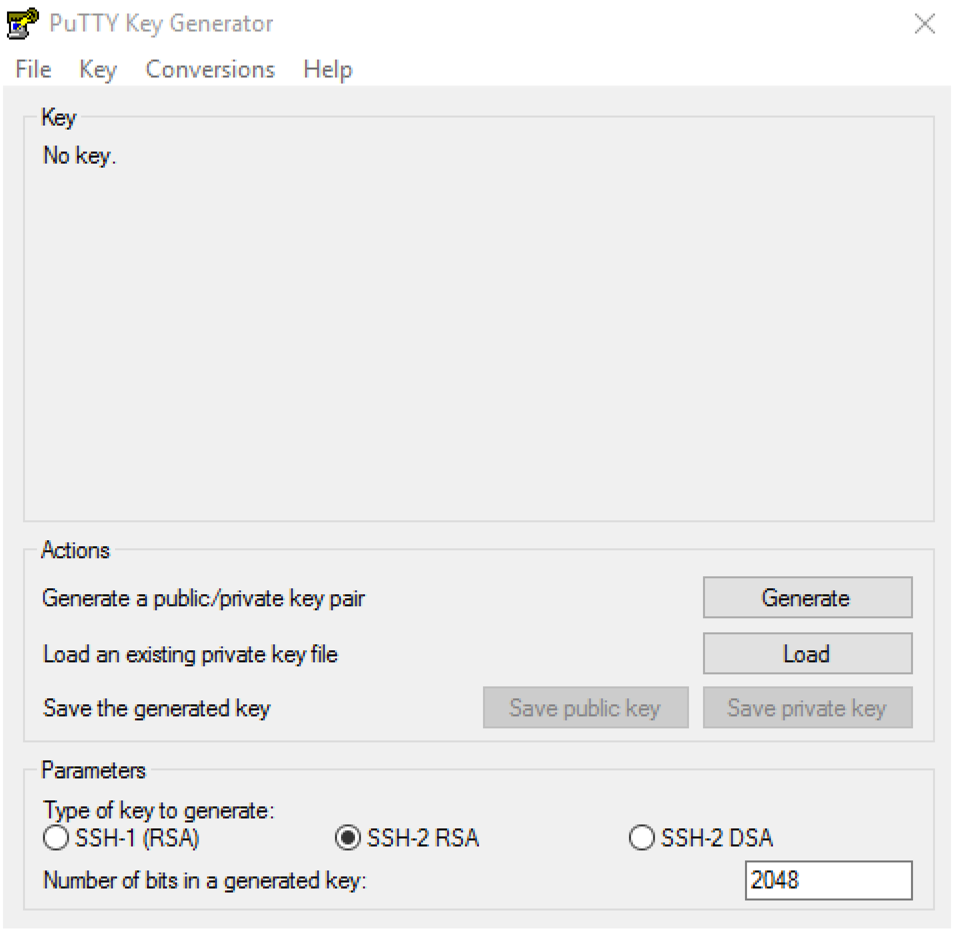
To access our lab VMs remotely, we will be using the SSH protocol. If you want to use Windows to manage your Linux systems over SSH, I recommend downloading and installing the following applications:

- PuTTYgen (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)
    - PuTTYgen is a tool for generating SSH public and private keys for doing key-based SSH authentication from Windows SSH clients.
- mRemoteNG (http://www.mRemoteNG.org/)
    - mRemoteNG is a tremendously useful application for managing a wide variety of remote access protocols - RDP, SSH (v1 and v2), telnet, rlogin, Citrix ICA, etc.
- WinSCP (https://winscp.net/eng/download.php)
    - A graphical SCP client for performing SCP transactions from Windows clients. Also supports key-based authentication.

Another tool that comes in handy now and then is a versatile text editor. While there is a wide range of programs in this category out there, I regularly use Notepad++ on Windows systems. So, in case you don't use it already or don't have another go-to editor by now, I suggest you get yourself a copy of the most current version at https://notepad-plus-plus.org/download/.

Generating an SSH key in Windows using PuTTYgen

Double-click puttygen.exe to open the application. (Please note that if you're running Windows 8 or higher, you may get a Windows SmartScreen "Windows Protected your PC" prompt. Click *More Info*, then click the *Run anyway* button.) You are greeted with this screen:



No key.

Generate a public/private key pair

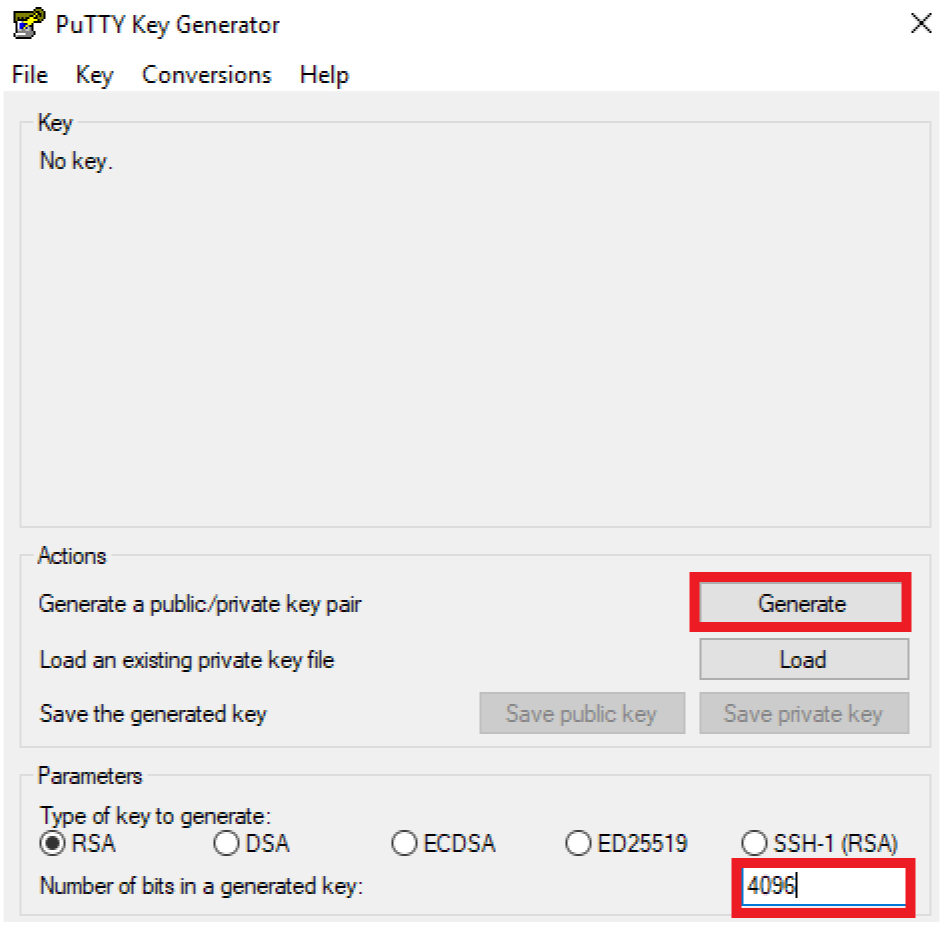
Load an existing private key file

Save the generated key

Type of key to generate:
( ) SSH-1 (RSA) [✓ SSH-2 RSA] ( ) SSH-2 DSA

Number of bits in a generated key: 2048

In the section headlined *Parameters*, make sure that the RSA radio button is selected. Next, in the input box to the right of *Number of bits in a generated key:*, change the default value to 4096. Finish by clicking the *Generate* button in the section labeled *Actions*.



No key.

Generate a public/private key pair

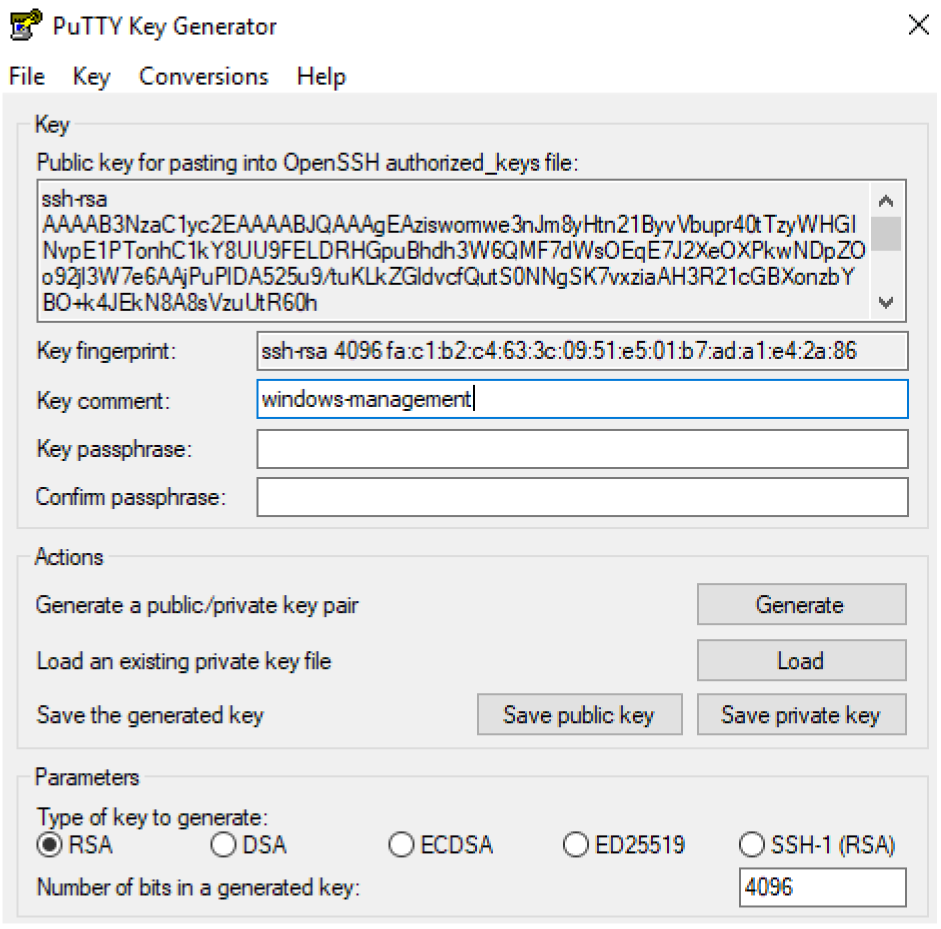
Load an existing private key file

Save the generated key

Type of key to generate:
[✓ RSA] DSA ECDSA ED25519 SSH-1 (RSA)

Number of bits in a generated key: 4096

Once the system successfully generated both the SSH private and the public key, the section titled “Key” is updated, displaying details similar to those shown in the picture below:



Key

Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAgEAziswomwe3nJm8yHtn21ByvVbupr40tTzyWHGI
NvpE1PTonhC1kY8UU9FELDRHGpuBhdh3W6QMF7dWsOEqE7J2XeOXPkwNDpZO
o92jl3W7e6AAjPuPIDA525u9/tuKLkZGldvcfQutS0NNgSK7vxziaAH3R21cGBXonzbY
BO+k4JEkN8A8sVzuUtR60h
```

Key fingerprint: ssh-rsa 4096 fa:c1:b2:c4:63:3c:09:51:e5:01:b7:ad:a1:e4:2a:86

Key comment: windows-management

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair

Load an existing private key file

Save the generated key

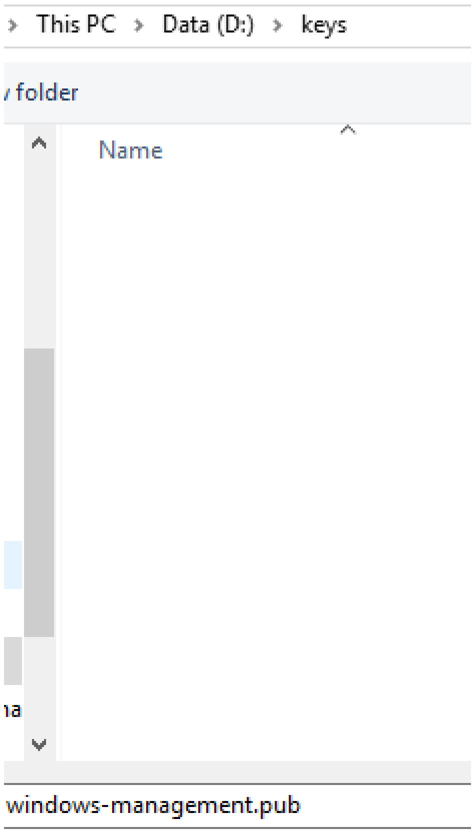
Parameters

Type of key to generate:
[✓ RSA] DSA ECDSA ED25519 SSH-1 (RSA)

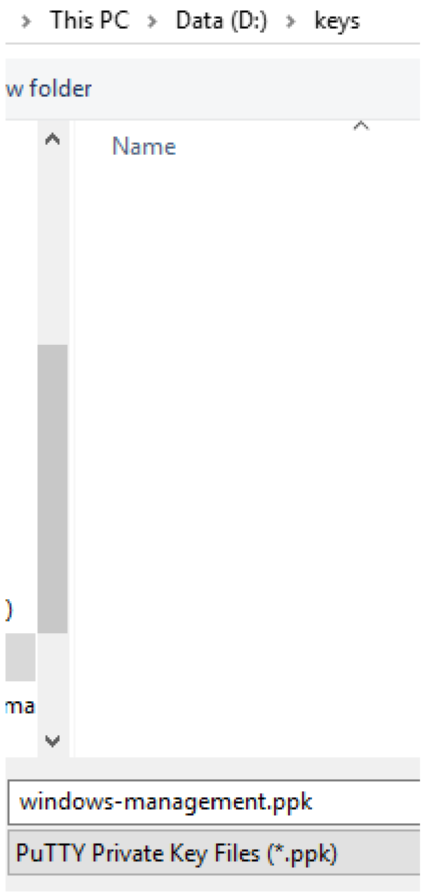
Number of bits in a generated key: 4096

Feel free to modify the default key comment provided by PuTTYgen to something that makes more sense to you. In my case, I changed it to windows-management to signify that this SSH key is for my Windows workstation. In addition to that, PuTTYgen offers the option to set a key passphrase for your new SSH key. If you decide to use one, this means that not only MUST you have the SSH key itself, and that you must ALSO know its passphrase in order to be able to use this SSH key to enable remote access to your lab VMs. You can also choose to leave the key passphrase field blank. Since this key is intended to be used only in a lab network environment that AT NO TIME should host anything of any value or significance, entering a passphrase isn’t exactly important in this case; leave it blank.

After adjusting the key comment, click the *Save public key* button. In the Explorer window that appears, enter a descriptive name like `windows-management.pub` for the public key file, and save it.



Next, click on the *Save private key* button, and save the private key as well. I would highly suggest saving this file in the same place you stored the the public key.



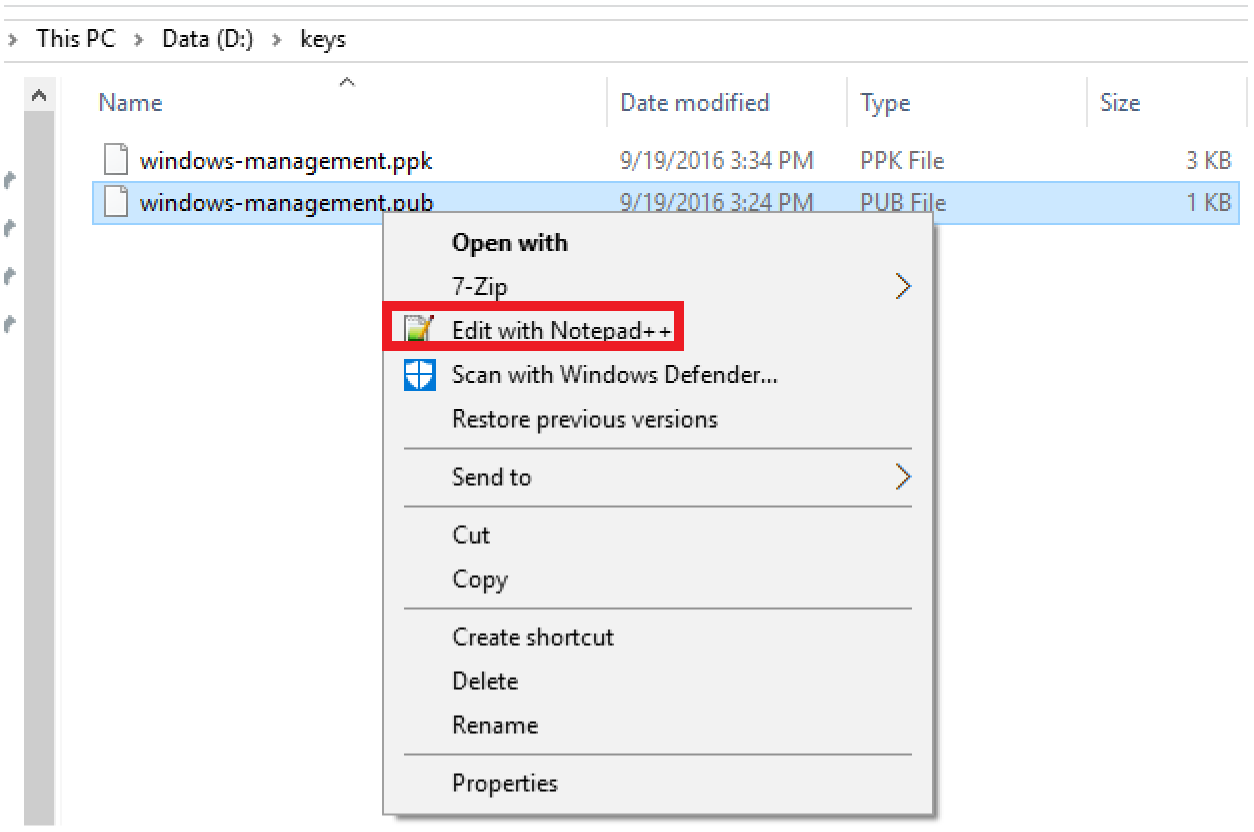
windows-management.ppk

PuTTY Private Key Files (*.ppk)

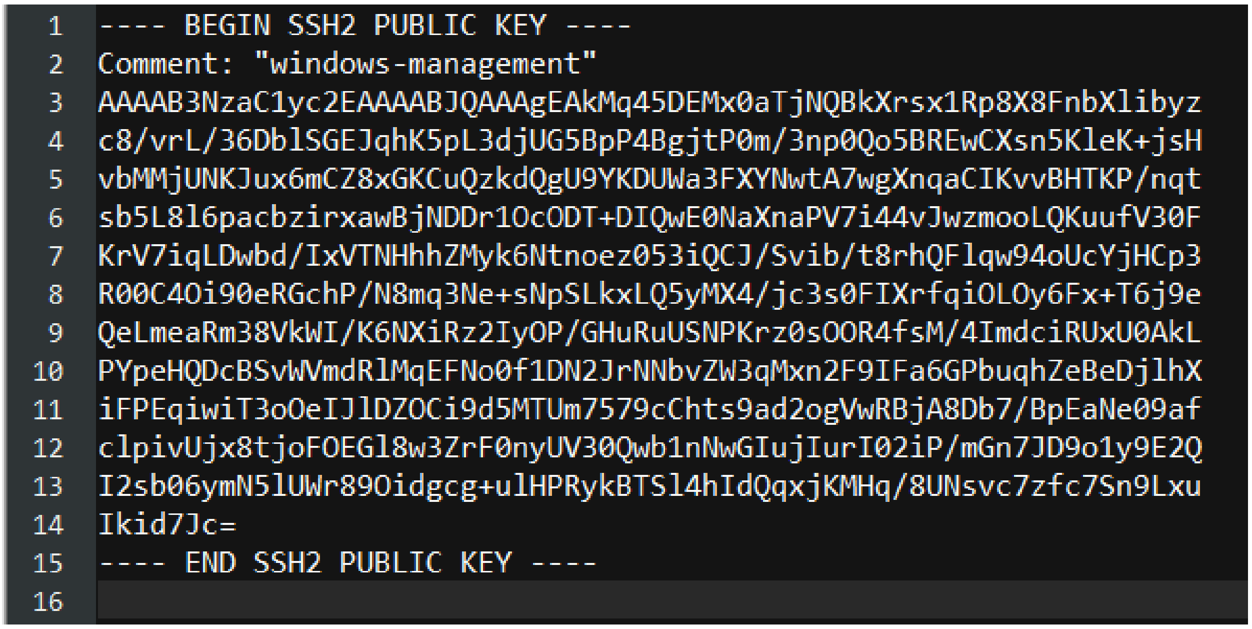
It is EXTREMELY important to keep these keys (especially the private one) in a safe location that only you have access to, in order to ensure others do not discover your SSH keys and attempt to abuse them. While this isn't much of a concern with a private lab environment, key security is something to ALWAYS be mindful of on real-world systems if you choose to use key-based SSH authentication.

The next step involves making some minor edits to the public key we just saved, since Linux and BSD systems expect SSH public keys to be formatted in a very specific way. To do this, we will be using Notepad++.

If you haven't done so already, download and install Notepad++ (https://notepad-plus-plus.org/). After doing so, open up Windows explorer and navigate to where you saved your public key file. Right click on the file, and select *Edit with Notepad++*.



Notepad++ displays the contents of the file, which should look similar to what is shown in the picture below:

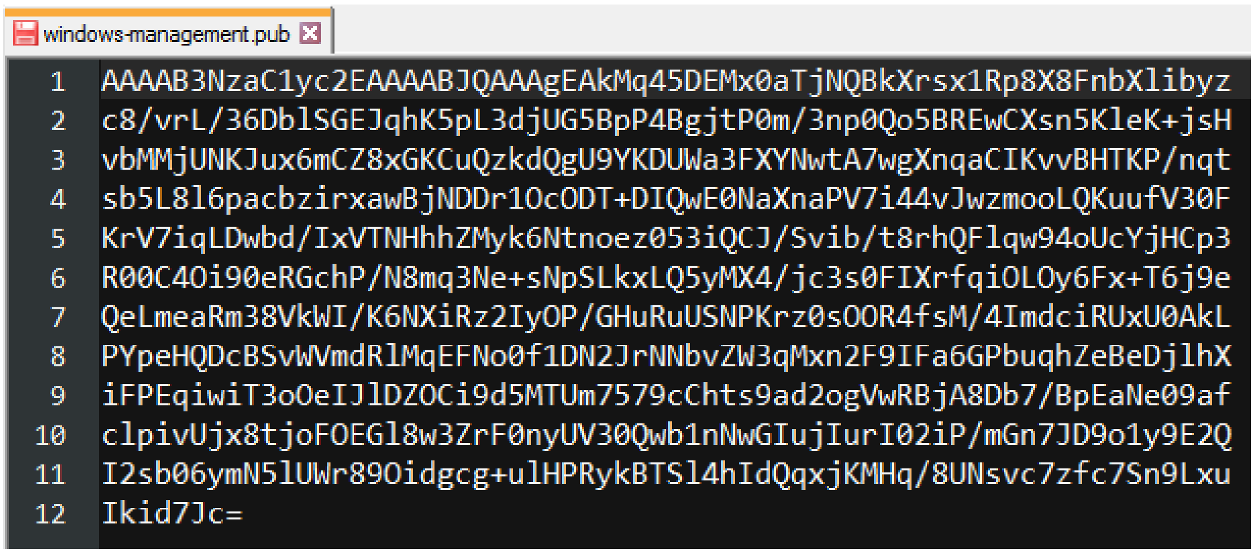


```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "windows-management"
AAAAB3NzaC1yc2EAAAABJQAAAgEAkMq45DEMx0aTjNQBkXrsx1Rp8X8FnbXlibyz
c8/vrL/36DblSGEJqhK5pL3djUG5BpP4BgjtP0m/3np0Qo5BREwCXsn5KleK+jsH
vbMMjUNKJux6mCZ8xGKCuQzkdQgU9YKDUWa3FXYNwtA7wgXnqaCIKvvBHTKP/nqt
sb5L8l6pacbzirxawBjNDDr1OcODT+DIQwE0NaXnaPV7i44vJwzmooLQKuufV30F
KrV7iqLDwbd/IxVTNHhhZMyk6Ntnoez053iQCJ/Svib/t8rhQFlqw94oUcYjHCp3
R00C4Oi90eRGchP/N8mq3Ne+sNpSLkxLQ5yMX4/jc3s0FIXrfqiOLOy6Fx+T6j9e
QeLmeaRm38VkWI/K6NXiRz2IyOP/GHuRuUSNPKrz0sOOR4fsM/4ImdciRUxU0AkL
PYpeHQDcBSvWVmdRlMqEFNo0f1DN2JrNNbvZW3qMxn2F9IFa6GPbuqhZeBeDjlhX
iFPEqiwiT3oOeIJlDZOCi9d5MTUm7579cChts9ad2ogVwRBjA8Db7/BpEaNe09af
clpivUjx8tjoFOEGl8w3ZrF0nyUV30Qwb1nNwGIujIurI02iP/mGn7JD9o1y9E2Q
I2sb06ymN5lUWr89Oidgcg+ulHPRykBTSl4hIdQqxjKMHq/8UNsvc7zfc7Sn9Lxu
Ikid7Jc=
---- END SSH2 PUBLIC KEY ----
```

As mentioned above, Linux and BSD systems expect SSH public keys to be formatted in a very specific way. The key must be a SINGLE continuous line, and no other information apart from the key itself is allowed. So first of all, remove the following lines:

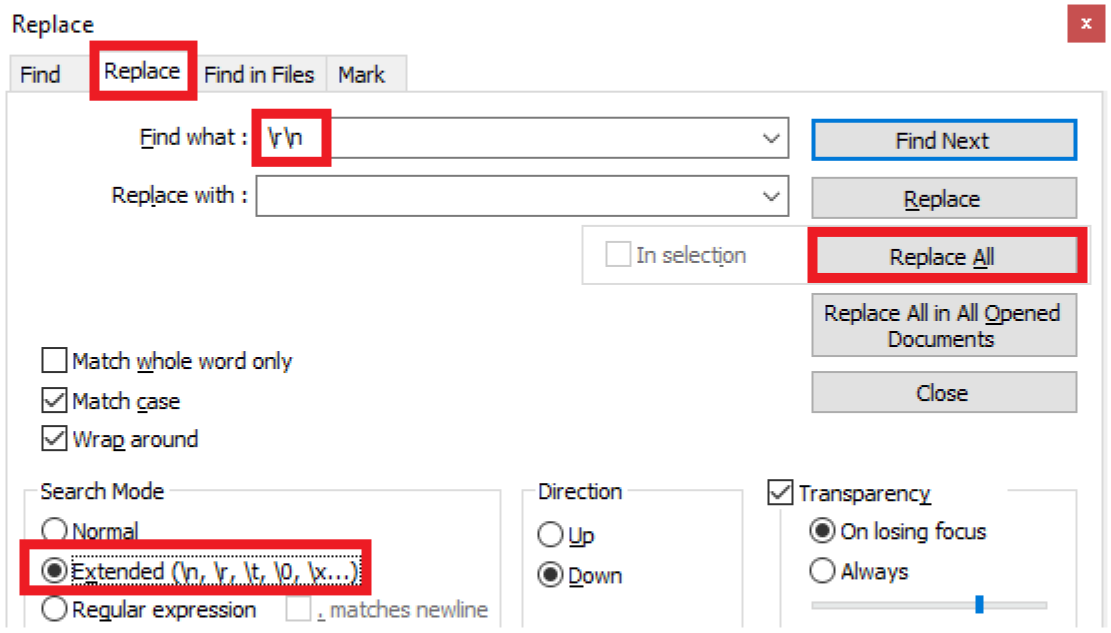
- ---- BEGIN SSH2 PUBLIC KEY ----
- Comment: "windows-management"
- ---- END SSH2 PUBLIC KEY ----

What is left is the data the actual public key is made up of:



```
AAAAB3NzaC1yc2EAAAABJQAAAgEAkMq45DEMx0aTjNQBkXrsx1Rp8X8FnbXlibyz
c8/vrL/36DblSGEJqhK5pL3djUG5BpP4BgjtP0m/3np0Qo5BREwCXsn5KleK+jsH
vbMMjUNKJux6mCZ8xGKCuQzkdQgU9YKDUWa3FXYNwtA7wgXnqaCIKvvBHTKP/nqt
sb5L8l6pacbzirxawBjNDDr1OcODT+DIQwE0NaXnaPV7i44vJwzmooLQKuufV30F
KrV7iqLDwbd/IxVTNHhhZMyk6Ntnoez053iQCJ/Svib/t8rhQFlqw94oUcYjHCp3
R00C4Oi90eRGchP/N8mq3Ne+sNpSLkxLQ5yMX4/jc3s0FIXrfqiOLOy6Fx+T6j9e
QeLmeaRm38VkWI/K6NXiRz2IyOP/GHuRuUSNPKrz0sOOR4fsM/4ImdciRUxU0AkL
PYpeHQDcBSvWVmdRlMqEFNo0f1DN2JrNNbvZW3qMxn2F9IFa6GPbuqhZeBeDjlhX
iFPEqiwiT3oOeIJlDZOCi9d5MTUm7579cChts9ad2ogVwRBjA8Db7/BpEaNe09af
clpivUjx8tjoFOEGl8w3ZrF0nyUV30Qwb1nNwGIujIurI02iP/mGn7JD9o1y9E2Q
I2sb06ymN5lUWr89Oidgcg+ulHPRykBTSl4hIdQqxjKMHq/8UNsvc7zfc7Sn9Lxu
Ikid7Jc=
```

Highlight all of the remaining text, then, on the main menu, go to *Search* > *Replace...* to open up the Find/Replace window with the Replace tab selected. In the Search Mode section at the bottom of the window, select the radio button next to Extended. Next, make sure that the *Replace with:* input box is empty, then enter the following for *Find what: field:*



Find what : \r\n

Replace with :

[✓] Match case

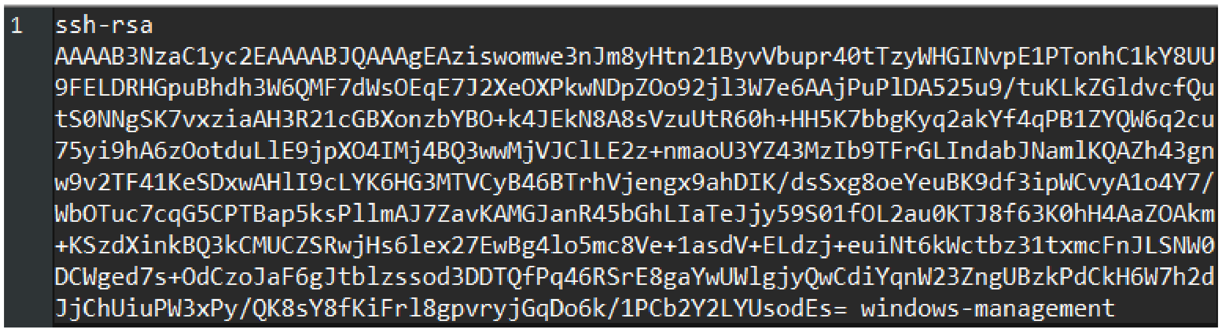
[✓] Wrap around

[ ] Match whole word only

Search Mode: [✓ Extended (\n, \r, \t, \0, \x...)]

Direction: [✓ Down]

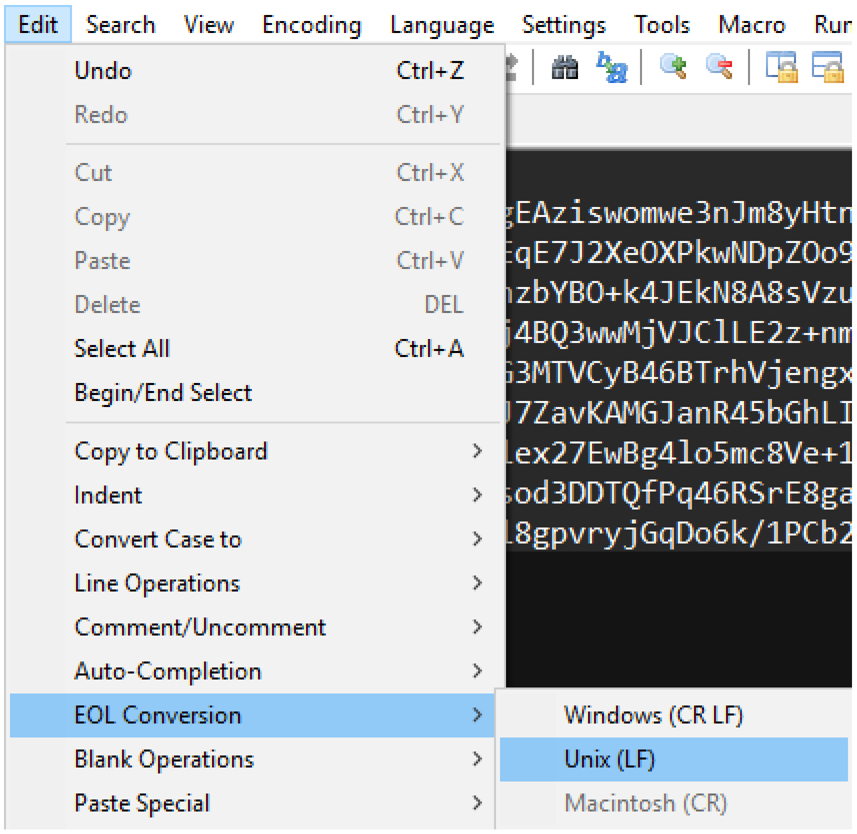
Click *Replace All*, then click *Close* to exit the Find/Replace window. We just replaced all of the newline and carriage return characters in our text file, turning the key into a single, continuous line. Now, at the beginning of the line, insert the text “`ssh-rsa `“. It is important to include the blank space at the end, exactly like it is given in the quotation marks. Next, at the end of the line, add the text “` windows-management`” (or whatever comment you want to associate with this key). This time around, it is mandatory to have the blank space between the END of the key and the beginning of the comment. If done correctly, the key should look like this:



```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAgEAziswomwe3nJm8yHtn21ByvVbupr40tTzyWHGINvpE1PTonhC1kY8UU
9FELDRHGpuBhdh3W6QMF7dWsOEqE7J2XeOXPkwNDpZOo92jl3W7e6AAjPuPlDA525u9/tuKLkZGldvcfQu
tS0NNgSK7vxziaAH3R21cGBXonzbYBO+k4JEkN8A8sVzuUtR60h+HH5K7bbgKyq2akYf4qPB1ZYQW6q2cu
75yi9hA6zOotduLlE9jpXO4IMj4BQ3wwMjVJClLE2z+nmaoU3YZ43MzIb9TFrGLIndabJNamlKQAZh43gn
w9v2TF41KeSDxwAHlI9cLYK6HG3MTVCyB46BTrhVjengx9ahDIK/dsSxg8oeYeuBK9df3ipWCvyA1o4Y7/
WbOTuc7cqG5CPTBap5ksPllmAJ7ZavKAMGJanR45bGhLIaTeJjy59S01fOL2au0KTJ8f63K0hH4AaZOAkm
+KSzdXinkBQ3kCMUCZSRwjHs6lex27EwBg4lo5mc8Ve+1asdV+ELdzj+euiNt6kWctbz31txmcFnJLSNW0
DCWged7s+OdCzoJaF6gJtblzssod3DDTQfPq46RSrE8gaYwUWlgjyQwCdiYqnW23ZngUBzkPdCkH6W7h2d
JjChUiuPW3xPy/QK8sY8fKiFrl8gpvryjGqDo6k/1PCb2Y2LYUsodEs= windows-management
```

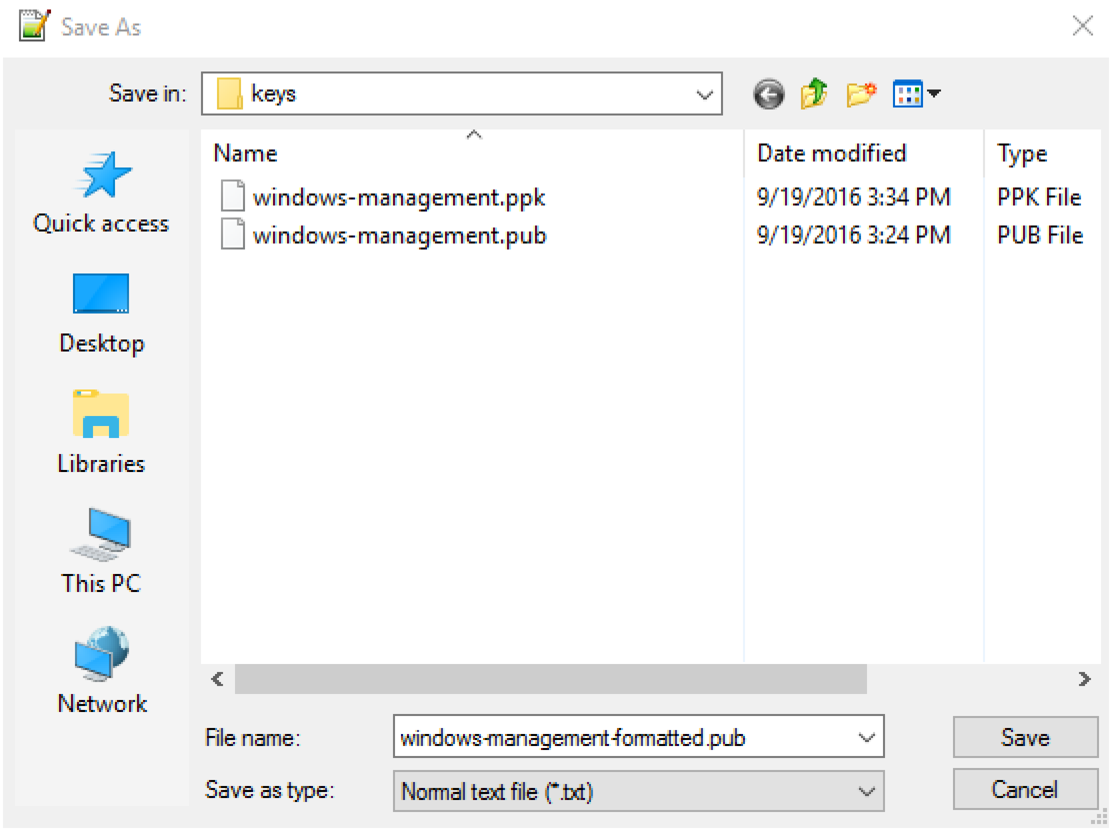
Note: that the line wrap after `ssh-rsa` is caused by the trailing blank space; there is no actual carriage return.

After modifying the text as described above, select *Edit > EOL Conversion > Unix (LF)* from the notepad++ main menu.



This option converts newline characters from the windows format or \r\n (the symbols that represent carriage return and newlines in a Windows text file) to just \n (newline characters). While the file should only be a single line, I have found that Unix/Linux systems behave strangely if this EOL Conversion has not been performed in advance.

Afterwards, select *File* > *Save As…* from the main menu. In the Save As dialog window, change the value in the “File name:” input field to `windows-management-formatted.pub`. Make sure that the file is going to be stored in the same folder the public and private SSH keys generated with PuTTYgen have been placed in. Click the Save button, and finish by quitting Notepad++.



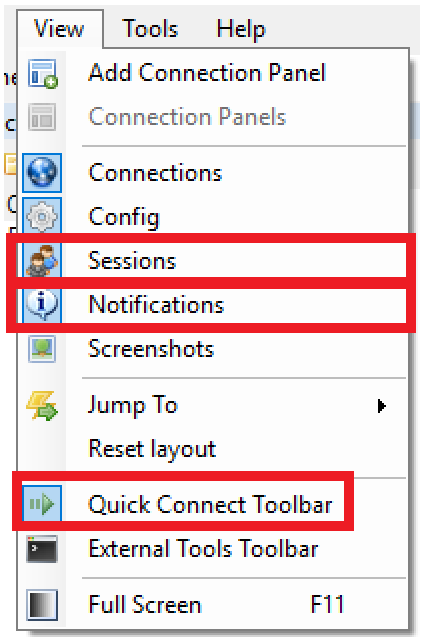
Name	Date modified	Type
windows-management.ppk	9/19/2016 3:34 PM	PPK File
windows-management.pub	9/19/2016 3:24 PM	PUB File

File name: windows-management-formatted.pub

Save as type: Normal text file (*.txt)

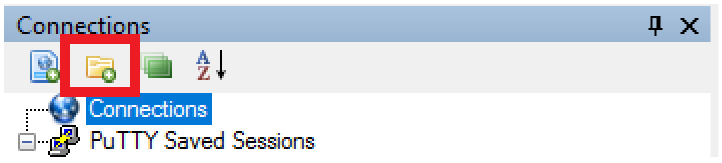
Using mRemoteNG - Connection Files

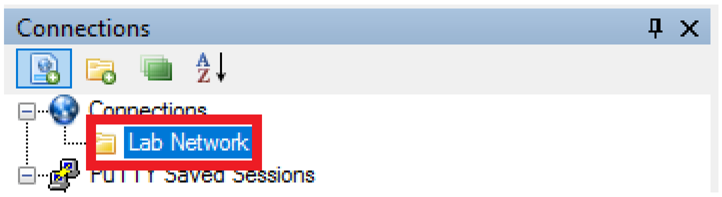
Start up mRemoteNG and open the View menu. From the highlighted options, click on *Sessions*, *Notifications* and *Quick Connect Toolbar*. Removing these items from view declutters mRemoteNG's user interface significantly.



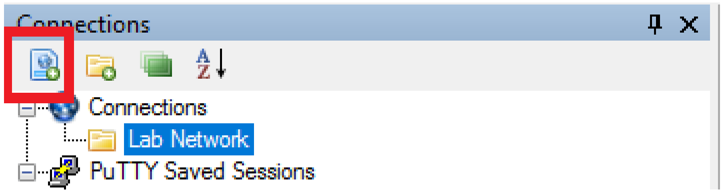
Disabling the above-mentioned views leaves us with three panes in the main screen: *Connections*, *Config*, and a large, grey window.

In the menu bar on top of the *Connection*s pane, first, click to highlight the Connections root entry. Next, click on the folder icon with the small green dot to add a folder underneath the Connections root. Name the new folder *Lab Network*.



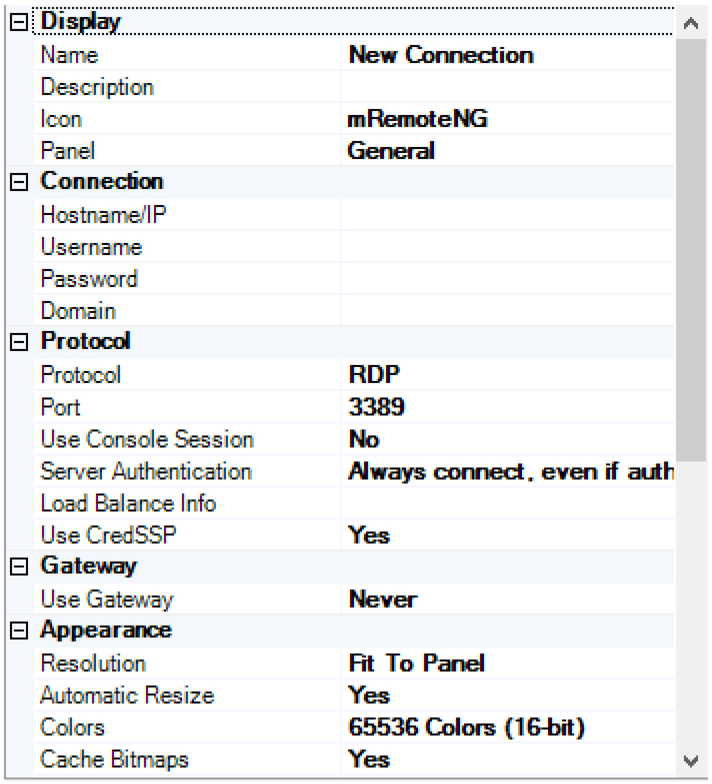


Click the *Lab Network* folder to highlight it, then click the leftmost icon in the Connections pane menu bar .



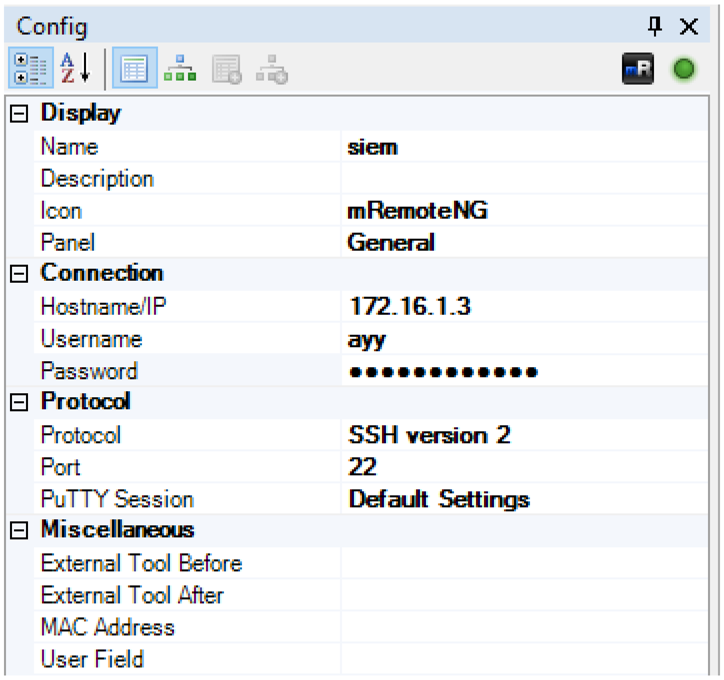
This creates a new connection entry underneath the *Lab Network* folder. In the *Config* pane, by default placed right below the Connections pane, a set of numerous options for the newly added entry is displayed.

We will go through the necessary settings for creating this first SSH connection (the SIEM VM) together. Afterwards, you will create the remaining two connections (for the Kali and IPS VMs, respectively) on your own.



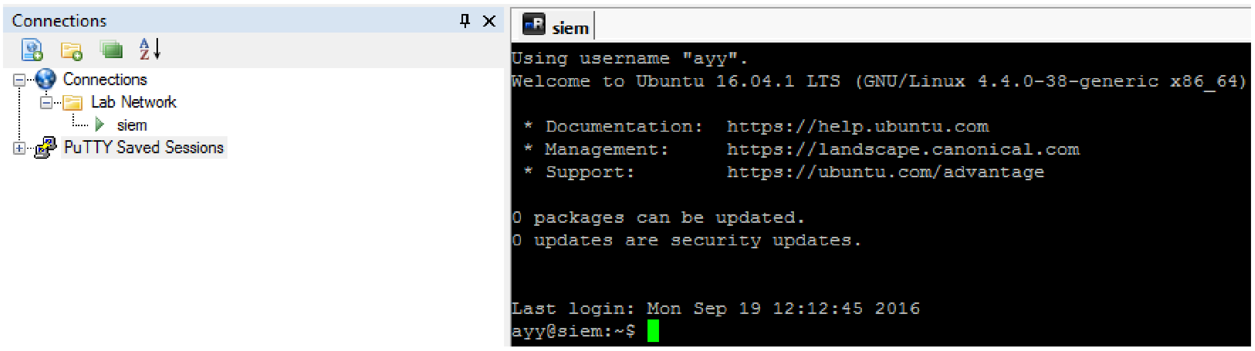
Display	
Name	New Connection
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	
Username	
Password	
Domain	
Protocol	
Protocol	RDP
Port	3389
Use Console Session	No
Server Authentication	Always connect, even if auth
Load Balance Info	
Use CredSSP	Yes
Gateway	
Use Gateway	Never
Appearance	
Resolution	Fit To Panel
Automatic Resize	Yes
Colors	65536 Colors (16-bit)
Cache Bitmaps	Yes

Starting in the *Display* section first, in the row labeled *Name*, double click on *New Connection* and insert "siem" in the box. Next, in the Protocol section, select *RDP* in the Protocol field. Click the drop-down arrow that appears, then select *SSH version 2* from the corresponding list. After that, in the *Connection* section, double click on the input field next to *Hostname/IP*, and enter the value "172.16.1.3". Then double click on the input field in the row named *Username*, and insert the username you used during the Ubuntu 16.04 installation of the SIEM VM (in my case, I named my first user "ayy"). Finally, double click on the input box to the right of *Password* to provide the password corresponding to the previously entered username.



Display	
Name	siem
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	172.16.1.3
Username	ayy
Password	●●●●●●●●●●●●
Protocol	
Protocol	SSH version 2
Port	22
PuTTY Session	Default Settings
Miscellaneous	
External Tool Before	
External Tool After	
MAC Address	
User Field	

When you're all done, double click on *siem* in the *Connections* pane. If the username and password were entered correctly, you should automatically authenticate and log in to the SIEM VM with no problems.



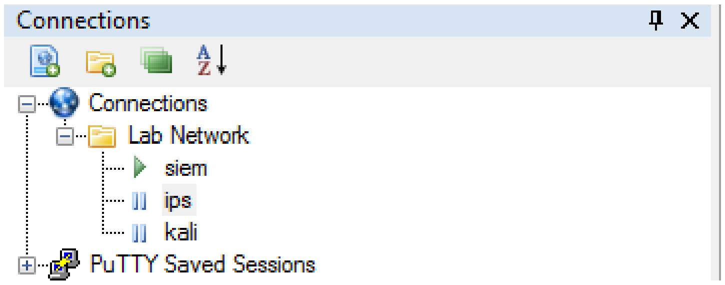
```
Using username "ayy".
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Mon Sep 19 12:12:45 2016
ayy@siem:~$
```

Continuing the above example, it’s now your turn. Create two more connections under the “Lab Network” folder, using the following settings:

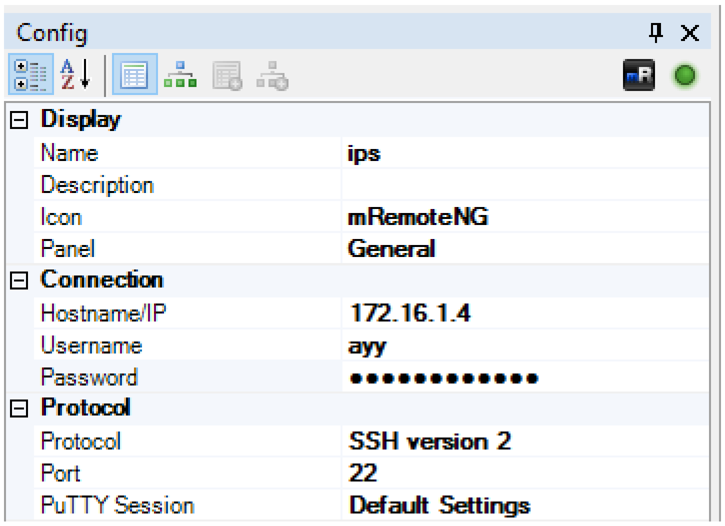
- Name: ips
- Protocol: SSH version 2
- Hostname/IP: 172.16.1.4
- Username: <username you entered during the setup of the IPS VM>
- Password: <password provided for the username above>

- Name: kali
- Protocol: SSH version 2
- Hostname/IP: 172.16.2.2
- Username: root
- Password: <password you assigned to the root account>
- Note: You will

If you configured everything correctly, the connections panel should host two new connections:

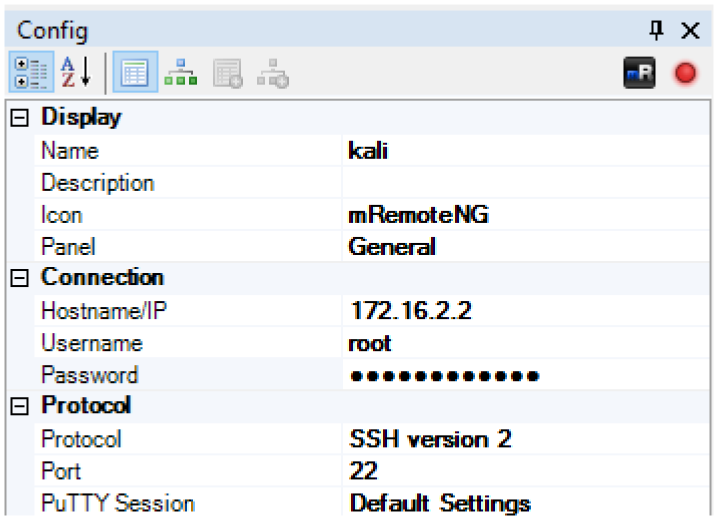


Their individual config options should look like this:



Config

Display	
Name	ips
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	172.16.1.4
Username	ayy
Password	●●●●●●●●●●●●
Protocol	
Protocol	SSH version 2
Port	22
PuTTY Session	Default Settings



Config

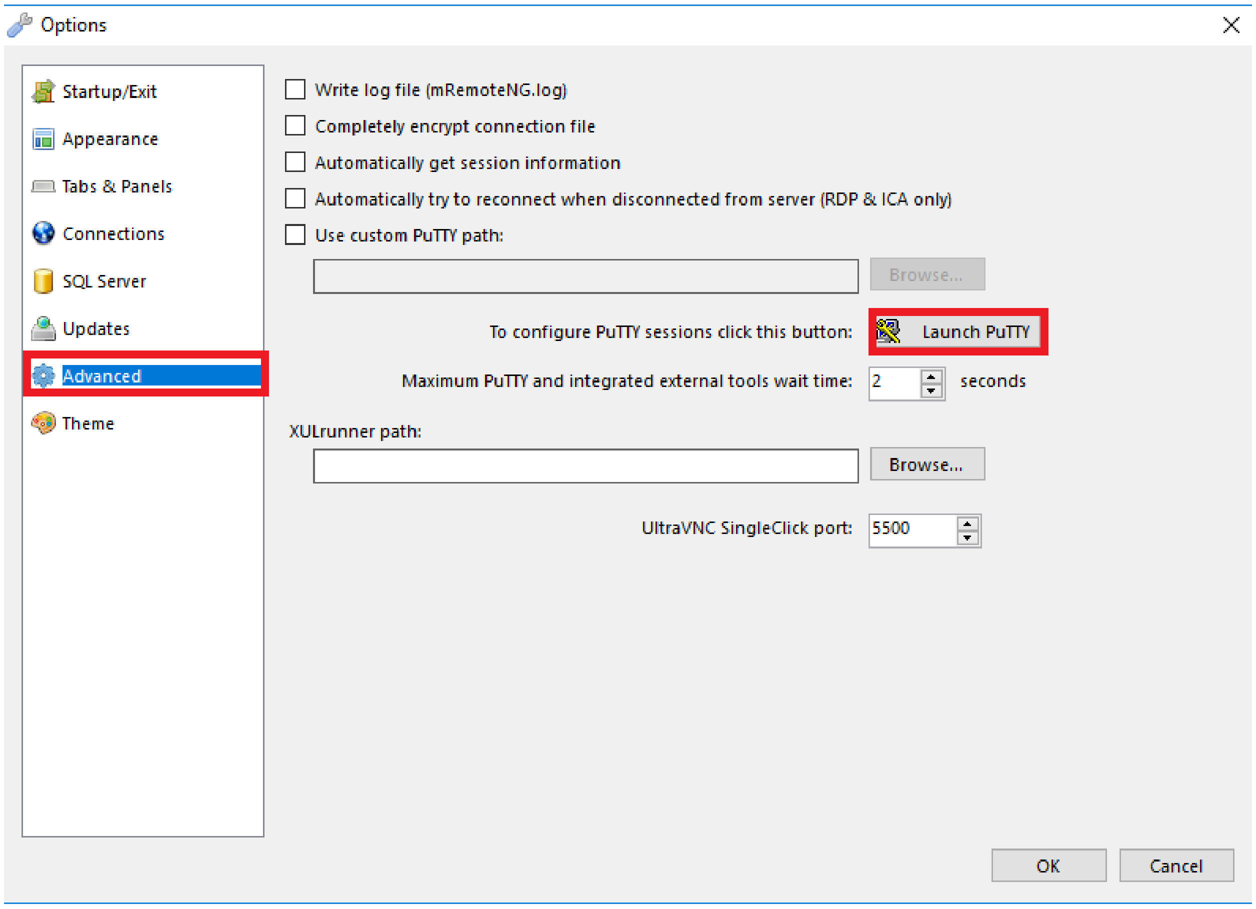
Display	
Name	kali
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	172.16.2.2
Username	root
Password	●●●●●●●●●●●●
Protocol	
Protocol	SSH version 2
Port	22
PuTTY Session	Default Settings

Feel free to test SSH connection to the IPS VM, which should already work just fine, but do note that for the time being, you will NOT be able to reach the Kali VM (Yes, even if you added a static route for the 172.16.2.0/24 network). Don't worry about this for right now, we're going to fix this shortly.

Using mRemoteNG - PuTTY Saved Sessions

Now that we have created the necessary SSH connections in mRemoteNG, the next step is to set up what is called a session profile. This will allow us to use the SSH key that we generated with PuTTYgen to authenticate to our lab VMs, instead of using a password. Or, if you password protected your private key earlier, it sets up a rudimentary form of two-factor authentication (e.g. you need to have the SSH key, and you must know the SSH key's password in order to connect).

From the main menu of mRemoteNG, choose *Tools* > *Options*. In the Options window that appears, select *Advanced* in the left pane, then click the *Launch PuTTY* button.


Options

Startup/Exit
Appearance
Tabs & Panels
Connections
SQL Server
Updates
Advanced
Theme

☐ Write log file (mRemoteNG.log)
☐ Completely encrypt connection file
☐ Automatically get session information
☐ Automatically try to reconnect when disconnected from server (RDP & ICA only)
☐ Use custom PuTTY path:

Browse...

To configure PuTTY sessions click this button: Launch PuTTY

Maximum PuTTY and integrated external tools wait time: 2 seconds

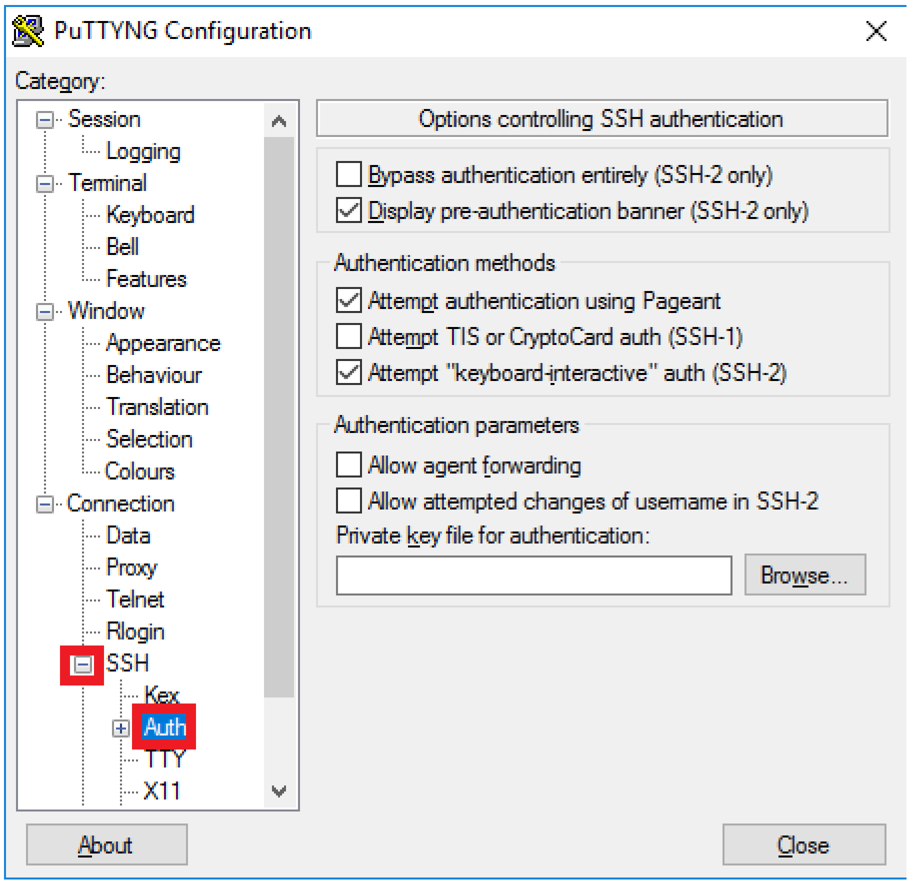
XULrunner path:

Browse...

UltraVNC SingleClick port: 5500

OK Cancel

This will open a window captioned *PuTTYNG Configuration*. On the left pane labeled *Category*, click on the + symbol next to *SSH* to expand the menu branch, then click on the *Auth* entry (this time the text itself, not the leading + sign).



Options controlling SSH authentication

- [ ] Bypass authentication entirely (SSH-2 only)
- [✓] Display pre-authentication banner (SSH-2 only)

Authentication methods

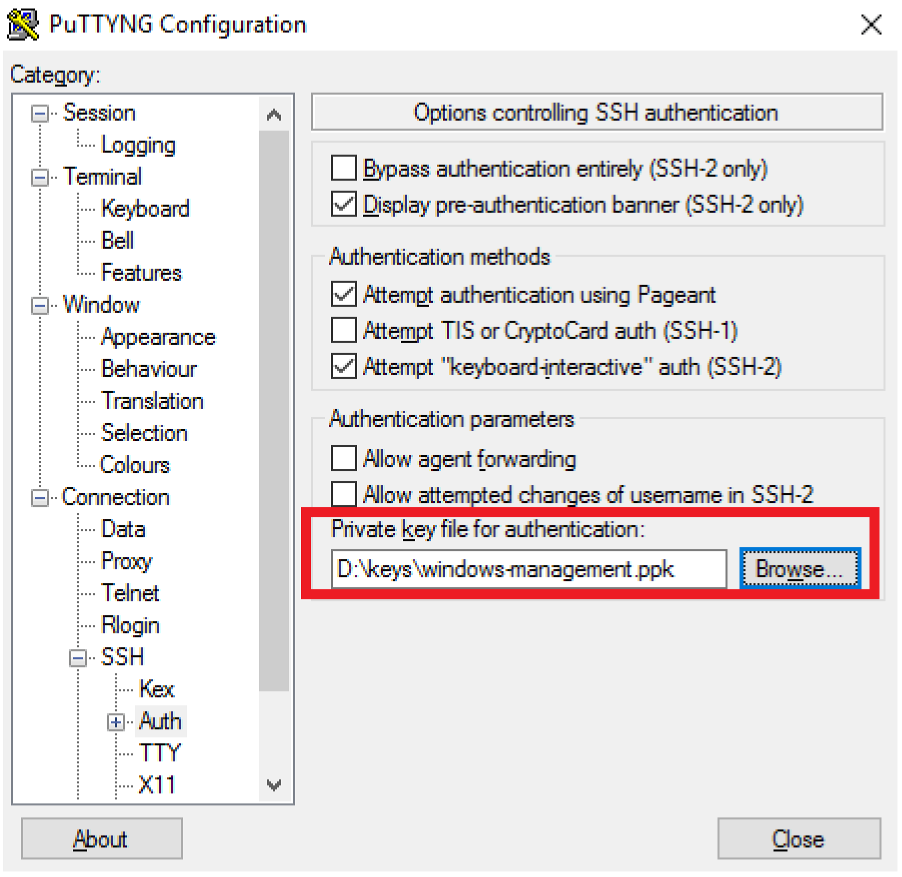
- [✓] Attempt authentication using Pageant
- [ ] Attempt TIS or CryptoCard auth (SSH-1)
- [✓] Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

- [ ] Allow agent forwarding
- [ ] Allow attempted changes of username in SSH-2

Private key file for authentication:

In the section headlined *Authentication parameters*, click the *Browse…* button next to *Private key file for authentication:*. An Explorer window pops up. Navigate to the folder where you stored the **PRIVATE** SSH key (the file with the `.ppk` extension) generated with PuTTYgen earlier, then double click to select it.

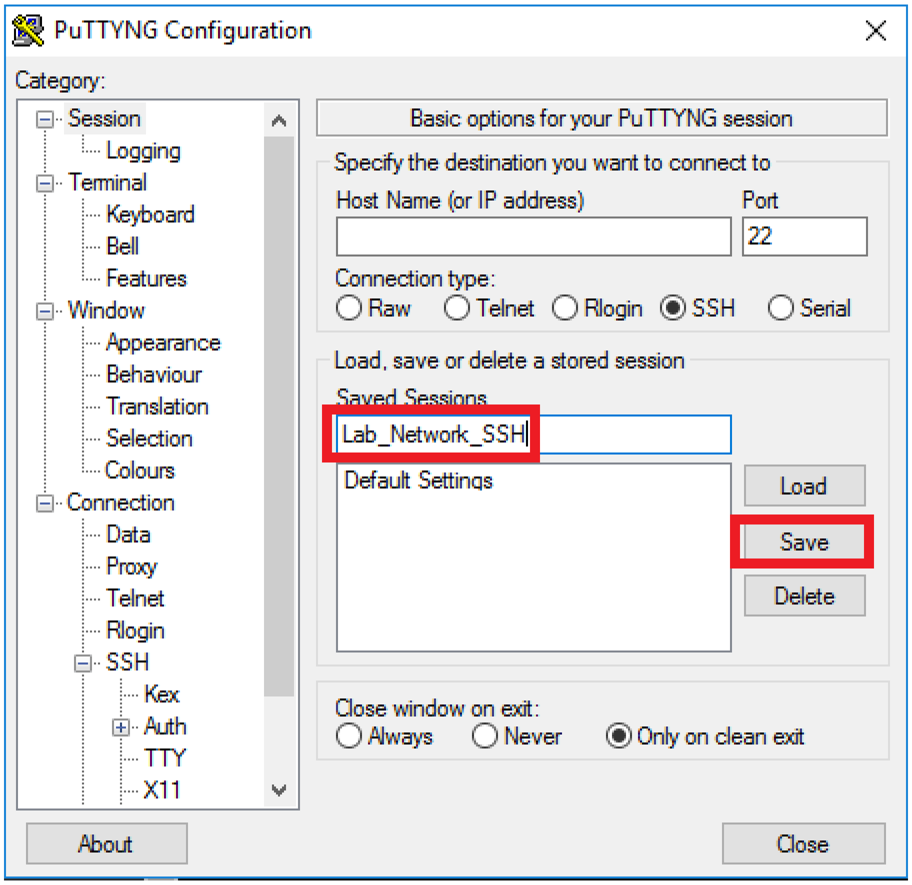


Bypass authentication entirely (SSH-2 only) [ ]
Display pre-authentication banner (SSH-2 only) [✓]

Authentication methods
Attempt authentication using Pageant [✓]
Attempt TIS or CryptoCard auth (SSH-1) [ ]
Attempt "keyboard-interactive" auth (SSH-2) [✓]

Authentication parameters
Allow agent forwarding [ ]
Allow attempted changes of username in SSH-2 [ ]
Private key file for authentication: D:\keys\windows-management.ppk

After selecting the file that contains the private SSH key, click on "*Session*" under the *Category:* pane. In the section titled *Load, save, or delete a stored session*, insert "Lab_Network_SSH" into the *Saved Sessions* input box, then click the *Save* button.



Port: 22

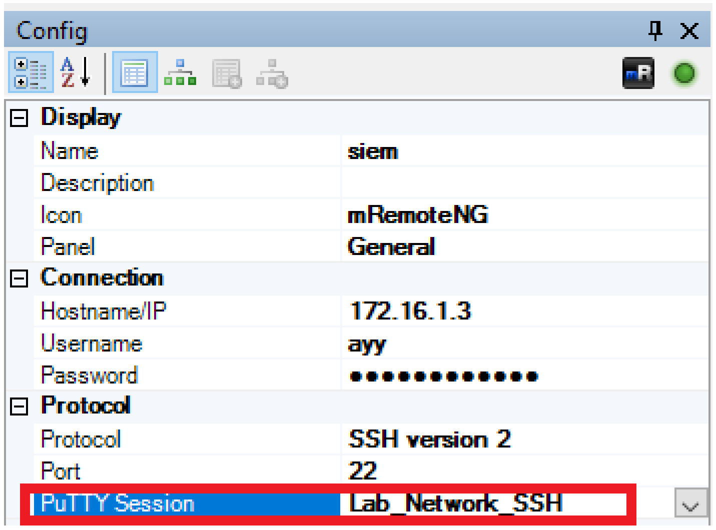
Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Saved Sessions: Lab_Network_SSH

Close window on exit: Always Never [✓ Only on clean exit]

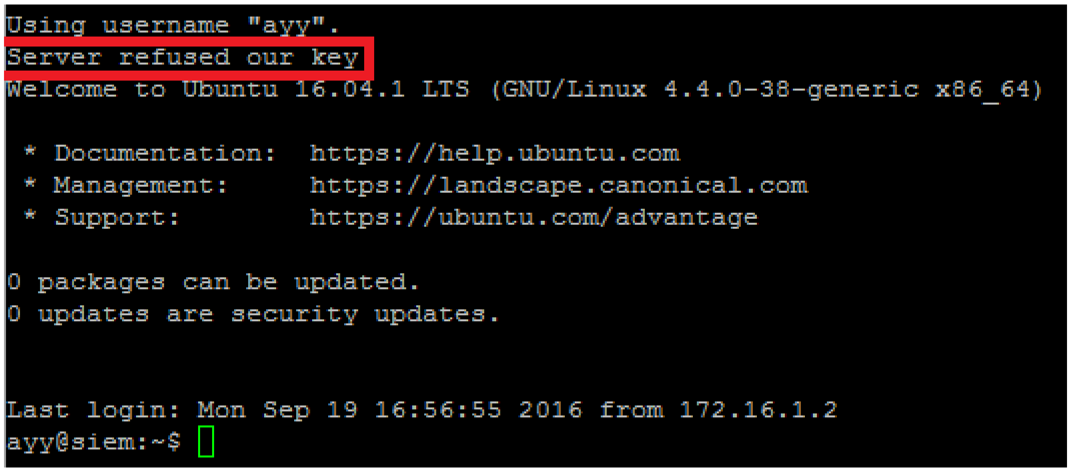
Click the Close button to exit *PuTTYNG Configuration*, then click the OK button in the mRemoteNG Options window. To return to the main screen.

Select the entry for "siem" in the *Connections* overview to bring up the corresponding settings in the *Config* pane. Under the Protocol heading, select the entry named *PuTTY Session*. Click on the drop-down arrow on the right and pick Lab_Network_SSH from the list.



Display	
Name	siem
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	172.16.1.3
Username	ayy
Password	●●●●●●●●●●●●
Protocol	
Protocol	SSH version 2
Port	22
PuTTY Session	Lab_Network_SSH

Return to the Connections pane, and double click on the "siem" entry to start an SSH connection to the respective VM. In the terminal window that appears, notice the text stating that the `Server refused our key`, yet the login attempt still succeeds.



```
Using username "ayy".
Server refused our key
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Mon Sep 19 16:56:55 2016 from 172.16.1.2
ayy@siem:~$
```

mRemoteNG establishes an SSH connection and to authenticate with the server using the SSH key we added to our PuTTY session. The problem is that the user attempting to log in doesn't have the public key for the used private key in its `authorized_keys` list on the remote server. So the server responds, "The user doesn't have the corresponding public key in their `authorized_keys` file." It immediately falls back to password-based authentication, accepting the password provided by mRemoteNG, and the user is logged in successfully. We'll fixing this problem in the following section.

Enabling Key-Based Authentication on Linux/Unix systems

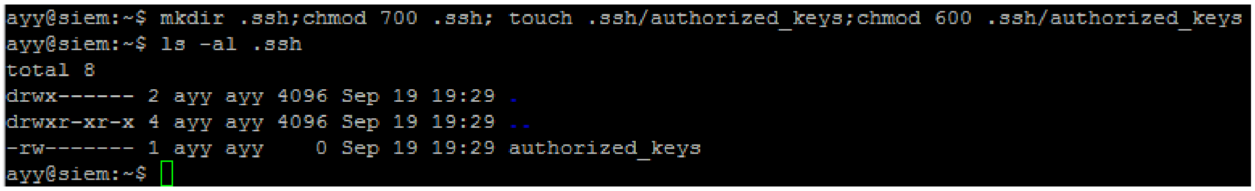
If you are continuing on from the previous section, you might still have an open SSH session to the SIEM VM, logged in with the user created during the initial system setup. In that case, you should already be in home directory of the connected user. If you are not, or if you just want to make sure that you are, run the command `cd ~/` inside the current SSH session. If you are not or no longer connected to the SIEM VM, open up mRemoteNG and create a new session. Once you are logged in, insert and execute the following:

```
mkdir ~/.ssh;chmod 700 ~/.ssh; touch ~/.ssh/authorized_keys;chmod 600
~/.ssh/authorized_keys
```

To verify everything has been processed properly, run the following command:

```
ls -al ~/.ssh
```

The output displayed should look similar to the one shown in the picture below:



```
ayy@siem:~$ mkdir .ssh;chmod 700 .ssh; touch .ssh/authorized_keys;chmod 600 .ssh/authorized_keys
ayy@siem:~$ ls -al .ssh
total 8
drwx------ 2 ayy ayy 4096 Sep 19 19:29 .
drwxr-xr-x 4 ayy ayy 4096 Sep 19 19:29 ..
-rw------- 1 ayy ayy    0 Sep 19 19:29 authorized_keys
ayy@siem:~$
```

The first command was actually made up of four instructions in rapid succession. It created the hidden directory `.ssh`, gave this directory file permissions that grant access only to the user who created it, then generated the empty file `authorized_keys` with the `touch` command (if the file does NOT already exist, touch creates it), and finally changed the permissions of this file to only allow the account currently used has read and write access.

The next part is tricky, in that we have to insert the contents of `windows-management-formatted.pub` on our local machine into ~/`.ssh/authorized_keys` in the user's home directory on the remote system. There are a several ways to do this. In the following sections, you will find three different approches

Key Copy Method 1: echo append to authorized_keys

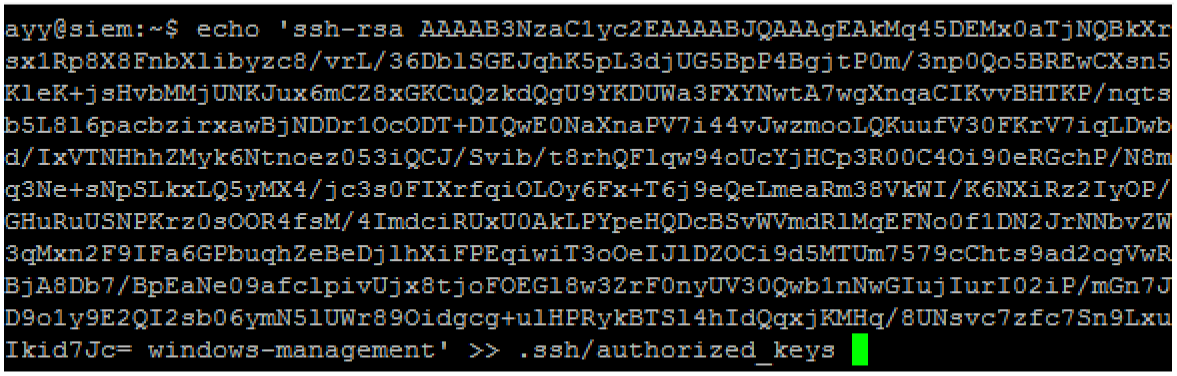
Open up the file `windows-management-formatted.pub` in Notepad++, highlight all of the text in the file, right click on it, then select *Copy*. The contents of the public key are now copied to the Windows clipboard.

Back in mRemoteNG, in the open SSH session to the SIEM VM, enter the following command:

`echo` ***'[right click here]'*** `>> .ssh/authorized_keys`

Be sure to include the single quotes in the command above!

By right-clicking inside the current SSH session (make sure to right click only once), the contents of the Windows clipboard were pasted into the terminal window. The text from the `windows-management-formatted.pub` file should now appear in between the single quotes. The final portion of the command in the terminal instructs the system to append this (echoed) key data to the `.ssh/authorized_keys` file. If your (somewhat extended) command line looks similar to the one you can see in the picture below, execute it.

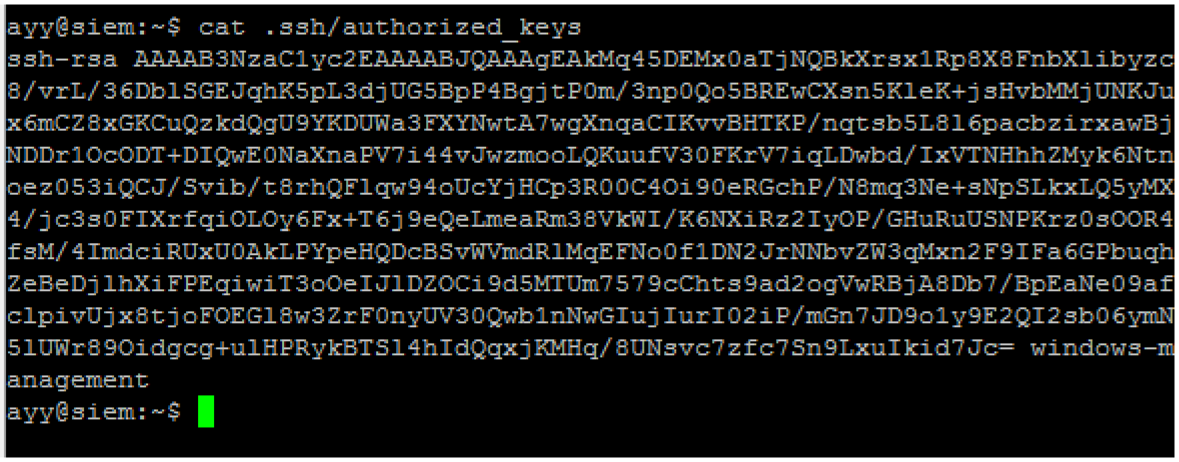


```
ayy@siem:~$ echo 'ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAgEAkMq45DEMx0aTjNQBkXr
sx1Rp8X8FnbXlibyzc8/vrL/36DblSGEJqhK5pL3djUG5BpP4BgjtP0m/3np0Qo5BREwCXsn5
KleK+jsHvbMMjUNKJux6mCZ8xGKCuQzkdQgU9YKDUWa3FXYNwtA7wgXnqaCIKvvBHTKP/nqts
b5L8l6pacbzirxawBjNDDr1OcODT+DIQwE0NaXnaPV7i44vJwzmooLQKuufV30FKrV7iqLDwb
d/IxVTNHhhZMyk6Ntnoez053iQCJ/Svib/t8rhQFlqw94oUcYjHCp3R00C4Oi90eRGchP/N8m
q3Ne+sNpSLkxLQ5yMX4/jc3s0FIXrfqiOLOy6Fx+T6j9eQeLmeaRm38VkWI/K6NXiRz2IyOP/
GHuRuUSNPKrz0sOOR4fsM/4ImdciRUxU0AkLPYpeHQDcBSvWVmdRlMqEFNo0f1DN2JrNNbvZW
3qMxn2F9IFa6GPbuqhZeBeDjlhXiFPEqiwiT3oOeIJlDZOCi9d5MTUm7579cChts9ad2ogVwR
BjA8Db7/BpEaNe09afclpivUjx8tjoFOEGl8w3ZrF0nyUV30Qwb1nNwGIujIurI02iP/mGn7J
D9o1y9E2QI2sb06ymN5lUWr89Oidgcg+ulHPRykBTSl4hIdQqxjKMHq/8UNsvc7zfc7Sn9Lxu
Ikid7Jc= windows-management' >> .ssh/authorized_keys
```

Next, run the following command:

```
cat .ssh/authorized_keys
```

The `cat` command reads the contents of a file and writes it to the terminal. Your output should look similar to this:



```
ayy@siem:~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAgEAkMq45DEMx0aTjNQBkXrsx1Rp8X8FnbXlibyzc
8/vrL/36Db1SGEJqhK5pL3djUG5BpP4BgjtP0m/3np0Qo5BREwCXsn5KleK+jsHvbMMjUNKJu
x6mCZ8xGKCuQzkdQgU9YKDUWa3FXYNwtA7wgXnqaCIKvvBHTKP/nqtsb5L816pacbzirxawBj
NDDr1OcODT+DIQwE0NaXnaPV7i44vJwzmooLQKuufV30FKrV7iqLDwbd/IxVTNHhhZMyk6Ntn
oez053iQCJ/Svib/t8rhQFlqw94oUcYjHCp3R00C4Oi90eRGchP/N8mq3Ne+sNpSLkxLQ5yMX
4/jc3s0FIXrfqiOLOy6Fx+T6j9eQeLmeaRm38VkWI/K6NXiRz2IyOP/GHuRuUSNPKrz0sOOR4
fsM/4ImdciRUxU0AkLPYpeHQDcBSvWVmdR1MqEFNo0f1DN2JrNNbvZW3qMxn2F9IFa6GPbuqh
ZeBeDj1hXiFPEqiwiT3oOeIJ1DZOCi9d5MTUm7579cChts9ad2ogVwRBjA8Db7/BpEaNe09af
clpivUjx8tjoFOEGl8w3ZrF0nyUV30Qwb1nNwGIujIurI02iP/mGn7JD9o1y9E2QI2sb06ymN
51UWr89Oidgcg+ulHPRykBTSl4hIdQqxjKMHq/8UNsvc7zfc7Sn9LxuIkid7Jc= windows-m
anagement
ayy@siem:~$
```

Key Copy Method 2: using `vi`

The vi text editor is older than dirt. It’s hard to use, hard to figure out, and has arcane invocations that nobody understands. And now you get to struggle with it as your elders did before you. Think of it as a rite of passage in the Unix/Linux world.

Open up the file `windows-management-formatted.pub` in Notepad++, highlight all of the text in the file, right click on it, then select *Copy*. The contents of the public key are now copied to the Windows clipboard.

Return to the open SSH session to the SIEM VM in mRemoteNG, and enter the following command:

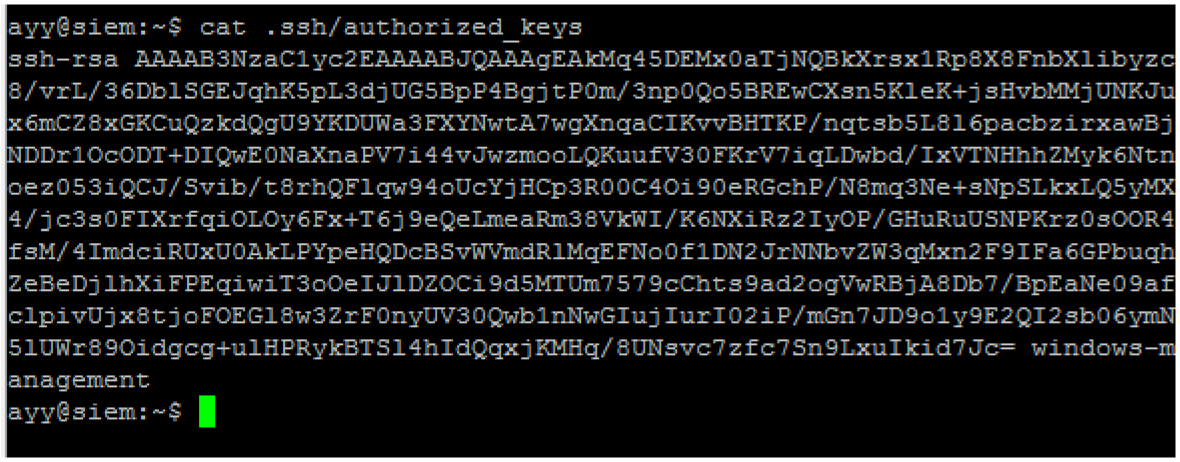
```
vi ~/.ssh/authorized_keys
```

This opens the `vi` editor in review mode. The `authorized_keys` file should be blank. No text, no other keys, nothing. Hit the `i` key to enter vi’s *insert mode*, then right click on the terminal window JUST ONCE. This fills in the contents of the Windows clipboard into the current editor session. After the contents have been copied into the open document, hit the `esc` key to exit the *insert mode*, then type:
`:wq!`

to save the file and exit `vi`.
Next, run the following command:

```
cat .ssh/authorized_keys
```

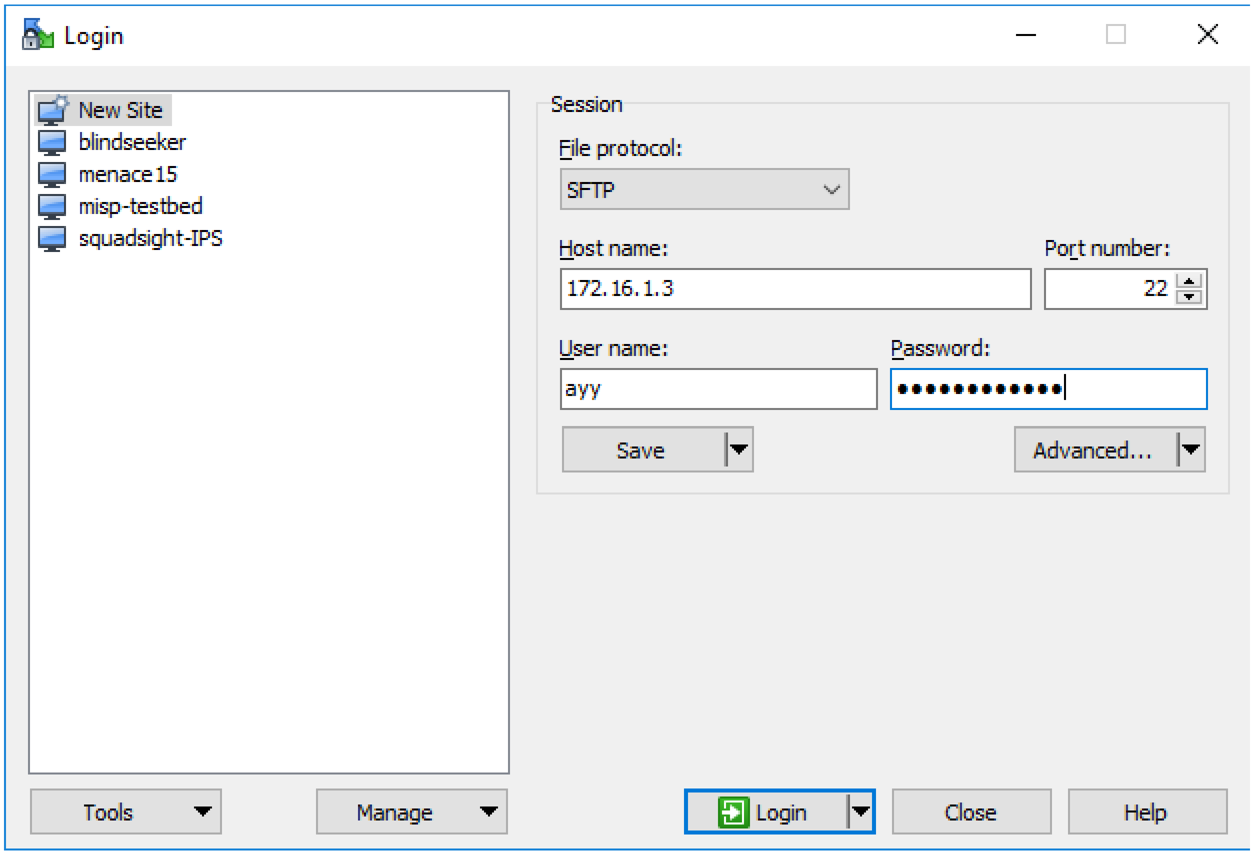
The `cat` command reads the contents of a file and writes it to the terminal. Your output should look similar to this:



```
ayy@siem:~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAgEAkMq45DEMx0aTjNQBkXrsx1Rp8X8FnbXlibyzc
8/vrL/36DblSGEJqhK5pL3djUG5BpP4BgjtP0m/3np0Qo5BREwCXsn5KleK+jsHvbMMjUNKJu
x6mCZ8xGKCuQzkdQgU9YKDUWa3FXYNwtA7wgXnqaCIKvvBHTKP/nqtsb5L8l6pacbzirxawBj
NDDr1OcODT+DIQwE0NaXnaPV7i44vJwzmooLQKuufV30FKrV7iqLDwbd/IxVTNHhhZMyk6Ntn
oez053iQCJ/Svib/t8rhQFlqw94oUcYjHCp3R00C4Oi90eRGchP/N8mq3Ne+sNpSLkxLQ5yMX
4/jc3s0FIXrfqiOLOy6Fx+T6j9eQeLmeaRm38VkWI/K6NXiRz2IyOP/GHuRuUSNPKrz0sOOR4
fsM/4ImdciRUxU0AkLPYpeHQDcBSvWVmdRlMqEFNo0f1DN2JrNNbvZW3qMxn2F9IFa6GPbuqh
ZeBeDjlhXiFPEqiwiT3oOeIJlDZOCi9d5MTUm7579cChts9ad2ogVwRBjA8Db7/BpEaNe09af
clpivUjx8tjoFOEGl8w3ZrF0nyUV30Qwb1nNwGIujIurI02iP/mGn7JD9o1y9E2QI2sb06ymN
5lUWr89Oidgcg+ulHPRykBTSl4hIdQqxjKMHq/8UNsvc7zfc7Sn9LxuIkid7Jc= windows-m
anagement
ayy@siem:~$ █
```

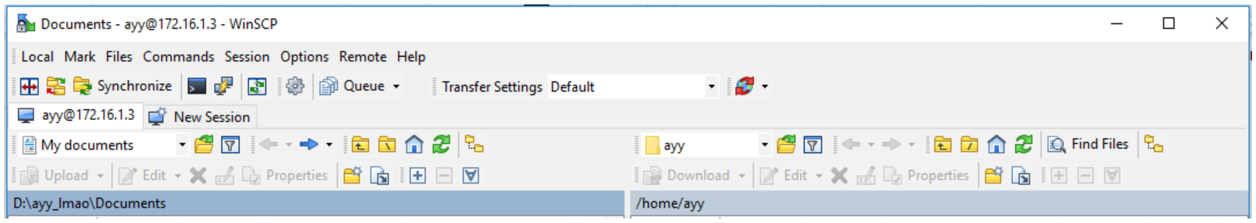
Key Copy Method 3: SCP

If you haven't done so yet, download the current version of WinSCP from https://winscp.net/eng/download.php, then install it on your Windows machine. Open the program, and select *New Site* in the left panel. On the right, in the section captioned Session, input the host information for the SIEM VM, along with the corresponding username and password,,then click the *Login* button.



File protocol: SFTP
Host name: 172.16.1.3
Port number: 22
User name: ayy
Password: ●●●●●●●●●●●●●

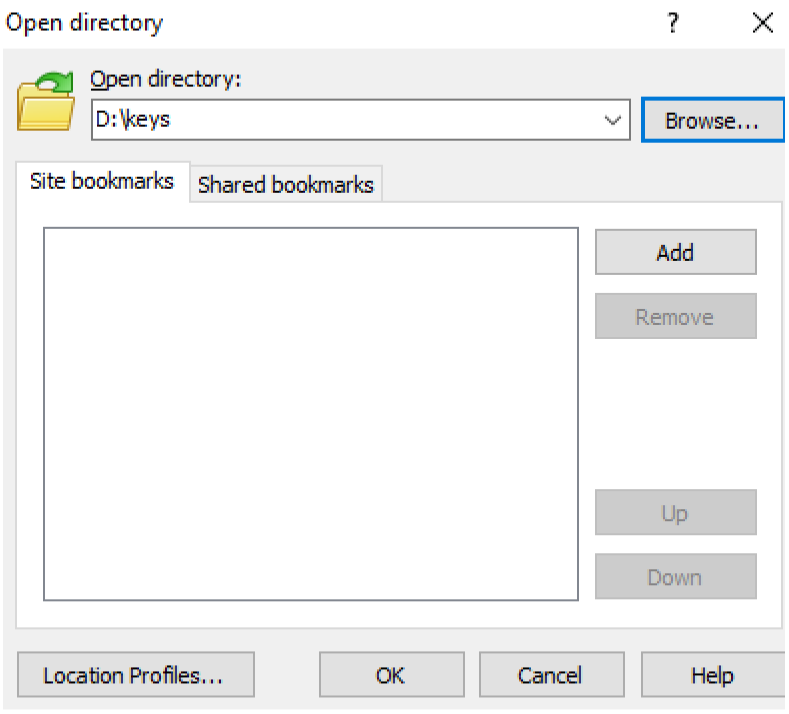
Once logged in successfully, you are greeted with with a screen made up of two panes: the left pane represents files and locations on the local computer, the right one those on the remote computer.


Documents - ayy@172.16.1.3 - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
ayy@172.16.1.3
New Session
My documents
ayy
Find Files
Upload
Edit
Properties
Download
Edit
Properties

D:\ayy_lmao\Documents

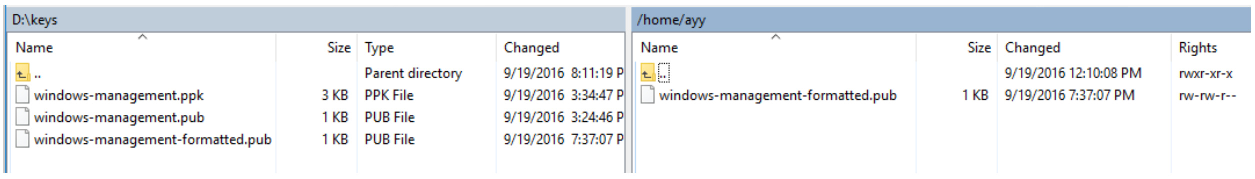
/home/ayy

Double click on the directory path on top of the left pane (the portion in blue, right below the menu toolbars). In the Open directory window that pops up, click the Browse button. Navigate to the directory that holds the SSH public and private keys, then click OK.



Open directory: D:\keys

The left pane of the WinSCP window changes to display the contents of the selected directory. Click on the file `windows-management-formatted.pub`, then drag it from the left pane to the right one. The file should appear in the right pane.



D:\keys

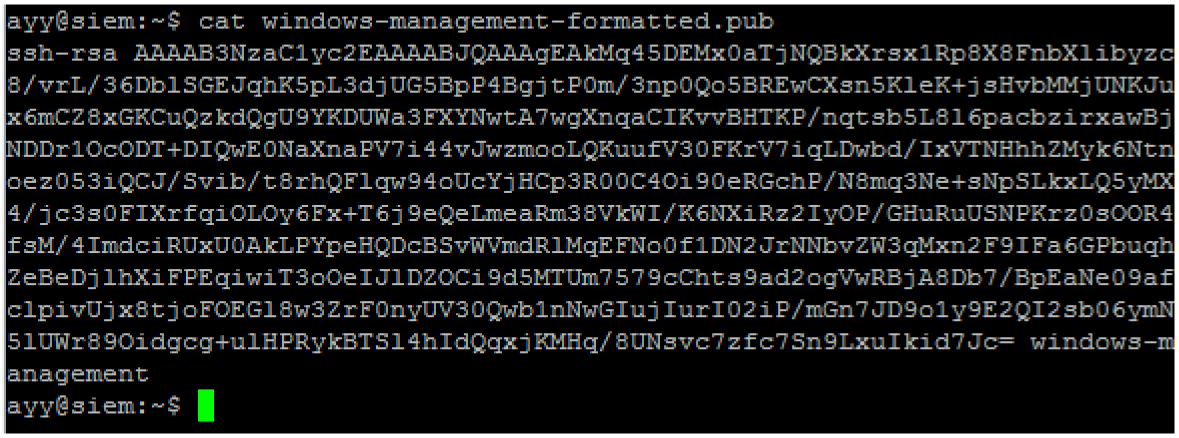
Name	Size	Type	Changed
..		Parent directory	9/19/2016 8:11:19 P
windows-management.ppk	3 KB	PPK File	9/19/2016 3:34:47 P
windows-management.pub	1 KB	PUB File	9/19/2016 3:24:46 P
windows-management-formatted.pub	1 KB	PUB File	9/19/2016 7:37:07 P

/home/ayy

Name	Size	Changed	Rights
..		9/19/2016 12:10:08 PM	rwxr-xr-x
windows-management-formatted.pub	1 KB	9/19/2016 7:37:07 PM	rw-rw-r--

Exit WinSCP, and start mRemoteNG if it isn't already running. In case the previously used session is closed, open an SSH connection to the SIEM VM. Once connected, run the command:

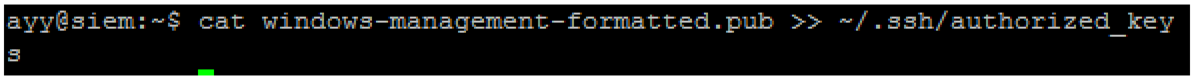
```
cat windows-management-formatted.pub
```



```
ayy@siem:~$ cat windows-management-formatted.pub
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAgEAkMq45DEMx0aTjNQBkXrsx1Rp8X8FnbXlibyzc
8/vrL/36DblSGEJqhK5pL3djUG5BpP4BgjtP0m/3np0Qo5BREwCXsn5KleK+jsHvbMMjUNKJu
x6mCZ8xGKCuQzkdQgU9YKDUWa3FXYNwtA7wgXnqaCIKvvBHTKP/nqtsb5L8l6pacbzirxawBj
NDDr1OcODT+DIQwE0NaXnaPV7i44vJwzmooLQKuufV30FKrV7iqLDwbd/IxVTNHhhZMyk6Ntn
oez053iQCJ/Svib/t8rhQFlqw94oUcYjHCp3R00C4Oi90eRGchP/N8mq3Ne+sNpSLkxLQ5yMX
4/jc3s0FIXrfqiOLOy6Fx+T6j9eQeLmeaRm38VkWI/K6NXiRz2IyOP/GHuRuUSNPKrz0sOOR4
fsM/4ImdciRUxU0AkLPYpeHQDcBSvWVmdRlMqEFNo0f1DN2JrNNbvZW3qMxn2F9IFa6GPbugh
ZeBeDjlhXiFPEqiwiT3oOeIJlDZOCi9d5MTUm7579cChts9ad2ogVwRBjA8Db7/BpEaNe09af
clpivUjx8tjoFOEGl8w3ZrF0nyUV30Qwb1nNwGIujIurI02iP/mGn7JD9o1y9E2QI2sb06ymN
5lUWr89Oidgcg+ulHPRykBTSl4hIdQqxjKMHq/8UNsvc7zfc7Sn9LxuIkid7Jc= windows-m
anagement
ayy@siem:~$
```

If the output in the terminal window resembles the one shown in the image above, the file made it over in one piece. Next, enter and execute the following:

```
cat windows-management-formatted.pub >> ~/.ssh/authorized_keys
```

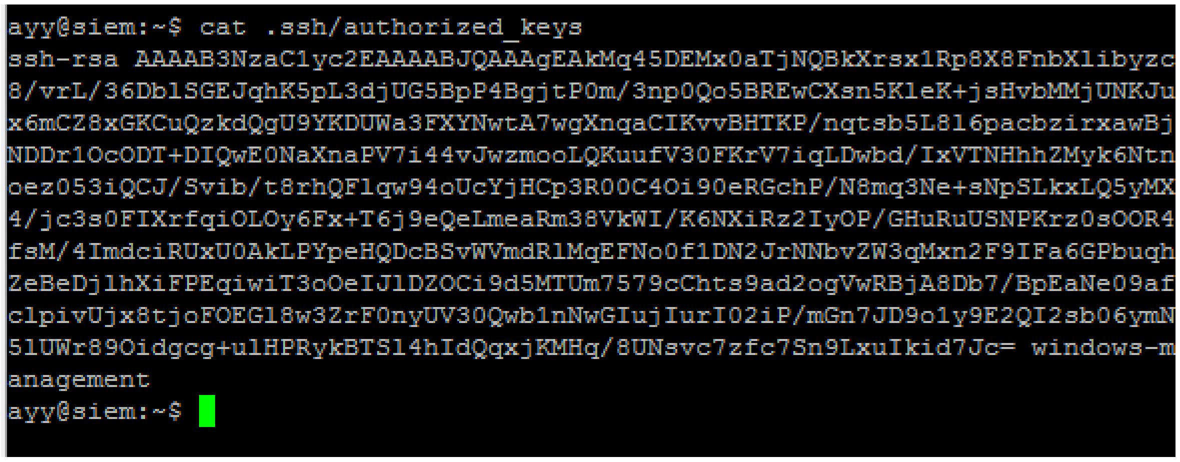


```
ayy@siem:~$ cat windows-management-formatted.pub >> ~/.ssh/authorized_key
s
```

Next, run the following command:

```
cat .ssh/authorized_keys
```

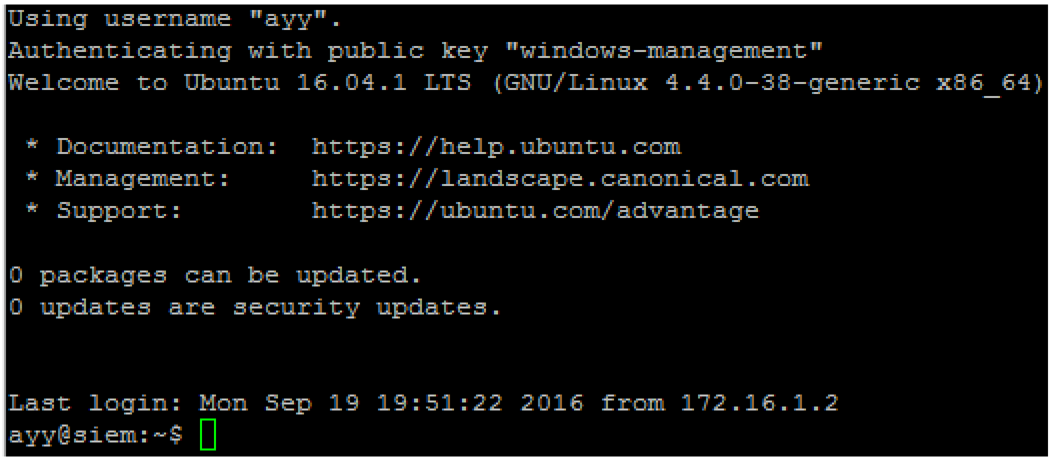
The `cat` command reads the contents of a file and writes it to the terminal. Your output should look similar to this:



```
ayy@siem:~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAgEAkMq45DEMx0aTjNQBkXrsx1Rp8X8FnbXlibyzc
8/vrL/36DblSGEJqhK5pL3djUG5BpP4BgjtP0m/3np0Qo5BREwCXsn5KleK+jsHvbMMjUNKJu
x6mCZ8xGKCuQzkdQgU9YKDUWa3FXYNwtA7wgXnqaCIKvvBHTKP/nqtsb5L8l6pacbzirxawBj
NDDr1OcODT+DIQwE0NaXnaPV7i44vJwzmooLQKuufV30FKrV7iqLDwbd/IxVTNHhhZMyk6Ntn
oez053iQCJ/Svib/t8rhQFlqw94oUcYjHCp3R00C4Oi90eRGchP/N8mq3Ne+sNpSLkxLQ5yMX
4/jc3s0FIXrfqiOLOy6Fx+T6j9eQeLmeaRm38VkWI/K6NXiRz2IyOP/GHuRuUSNPKrz0sOOR4
fsM/4ImdciRUxU0AkLPYpeHQDcBSvWVmdRlMqEFNo0f1DN2JrNNbvZW3qMxn2F9IFa6GPbugh
ZeBeDjlhXiFPEqiwiT3oOeIJlDZOCi9d5MTUm7579cChts9ad2ogVwRBjA8Db7/BpEaNe09af
clpivUjx8tjoFOEGl8w3ZrF0nyUV30Qwb1nNwGIujIurI02iP/mGn7JD9o1y9E2QI2sb06ymN
5lUWr89Oidgcg+ulHPRykBTSl4hIdQqxjKMHq/8UNsvc7zfc7Sn9LxuIkid7Jc= windows-m
anagement
ayy@siem:~$
```

Making sure it worked

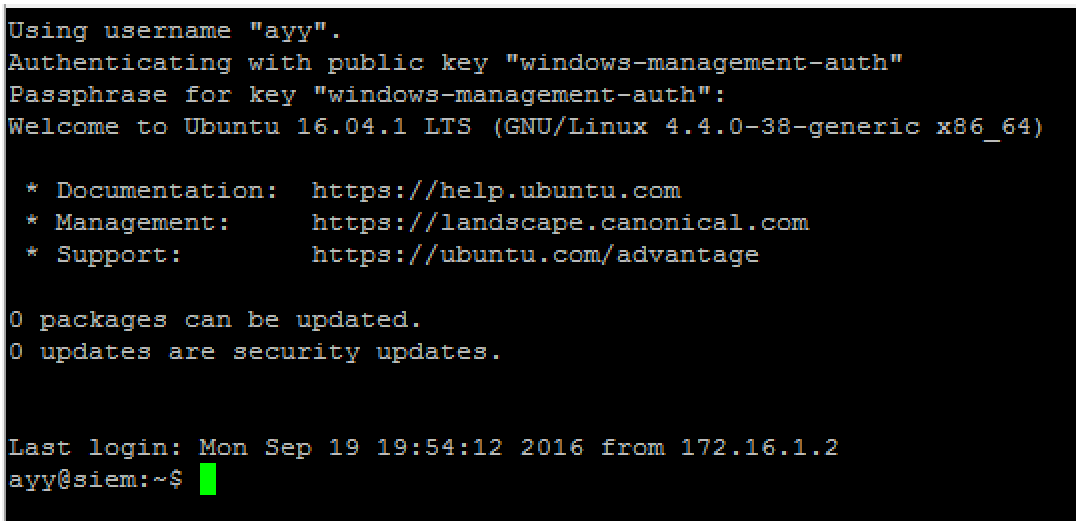
After verifying that the public key has been successfully copied to `.ssh/authorized_keys`, run `exit` inside the terminal window to leave the SSH session and log off of the SIEM VM. In mRemoteNG, on the *Config* pane, click on the Password entry for the siem connection. Clear the corresponding input field Then double click on siem in the Connections pane to reconnect to the VM.



```
Using username "ayy".
Authenticating with public key "windows-management"
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Mon Sep 19 19:51:22 2016 from 172.16.1.2
ayy@siem:~$ 
```

If the setup of the private SSH key for the user is correct, and if you didn’t put a passphrase on the SSH key we generated with PuTTYgen, you should be greeted with a login prompt with no need to enter a password. The server is comparing the public and private key to authenticate the user to the SIEM VM. If the comparison is successful (e.g. the public key on the system is related to the private key that has been sent to authenticate), the user will be logged in.

If you DID set a key passphrase during generation of the SSH keys earlier, you are prompted to enter the password for that key now.


```
Using username "ayy".
Authenticating with public key "windows-management-auth"
Passphrase for key "windows-management-auth":
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Mon Sep 19 19:54:12 2016 from 172.16.1.2
ayy@siem:~$ 
```

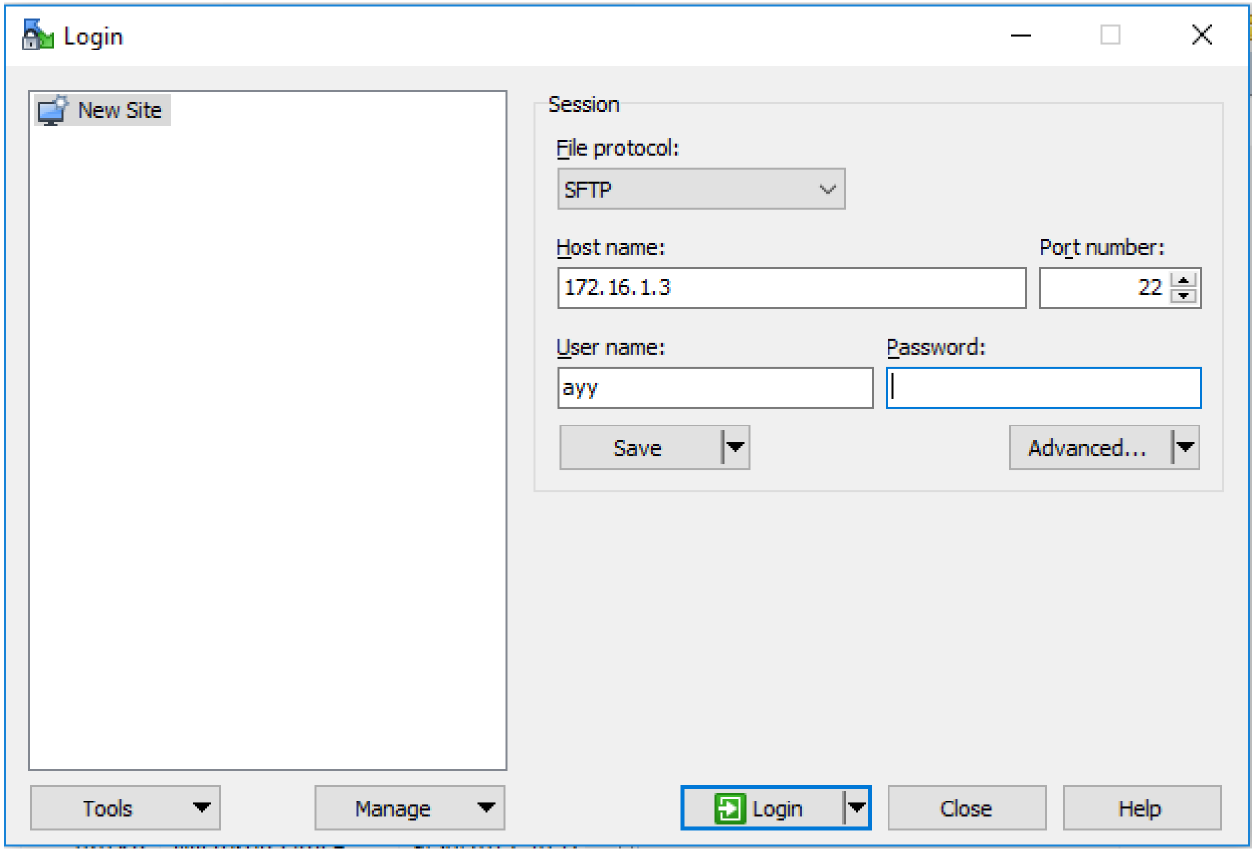
Once you have verified that key-based authentication is working on the SIEM virtual machine, you can repeat the process to enable key-based auth on the IPS VM. You may also wish to jump to the section How to Enable SSH on Kali Linux, if you planned on enabling remote access for the Kali LInux virtual machine.

How to use Key-Based Authentication with WinSCP

Assuming that you followed the instructions in the sections above, and furthermore assuming that you're using a Windows system, you have configured (or should be able to configure) key-based authentication for SSH. Another common usecase for this kind of authentication is SCP. SCP is a protocol that relies on using SSH as a transport which makes it a (more) secure alternative to FTP. In fact, one of the abovementioned methods for adding your SSH public key to the Linux servers in our lab makes use of SCP.

Just like with the key-based authentication guides above, I will instruct you on how to enable this functionality on the SIEM virtual machine, and leave it as an exercise to you, if you would like to enable this functionality on the remaining lab VMs.

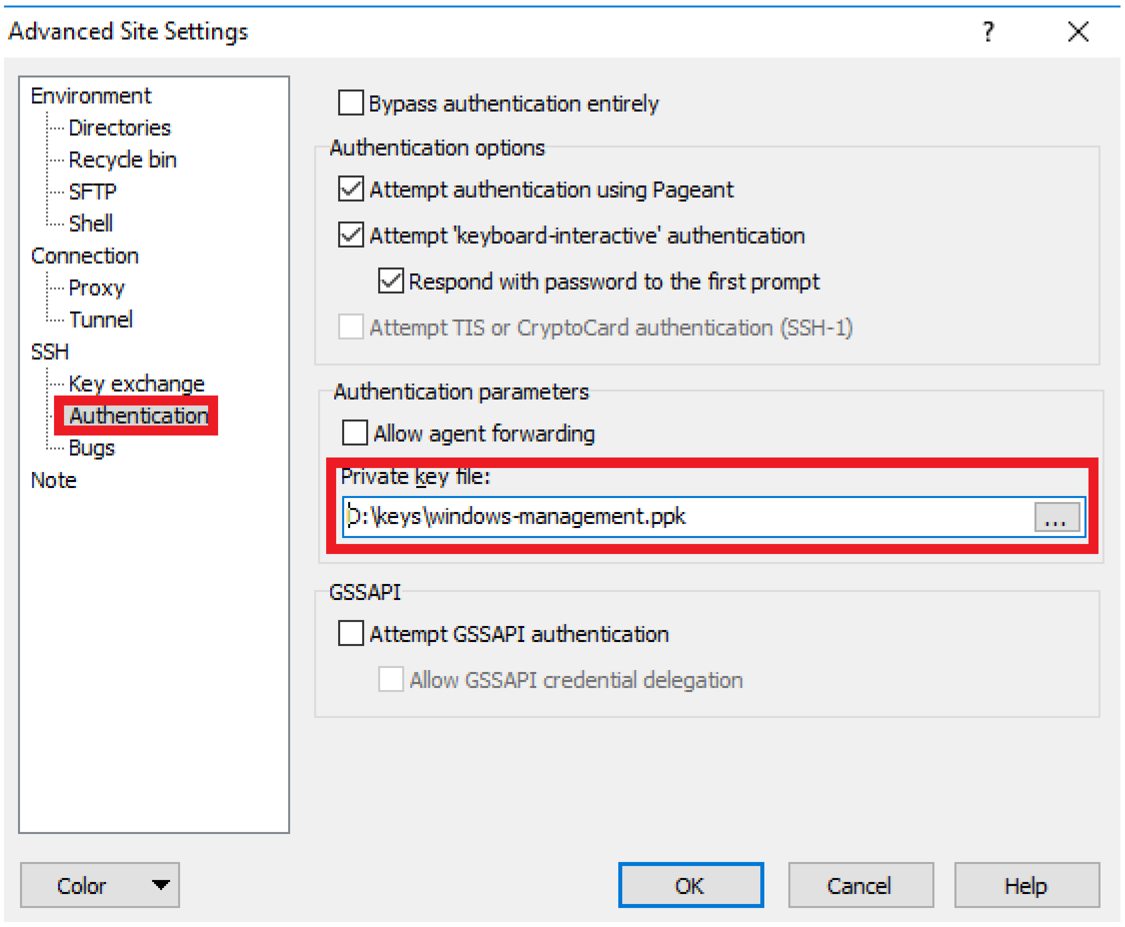
First, open WinSCP. You are greeted with a window captioned *Login*. The pane on the left shows a highlighted entry with a computer monitor icon labeled *New Site*, on the right side of the window is a section headlined *Session*. As we are creating a session for the SIEM VM, enter 172.16.1.3 into the input box below *Host name:* . Leave the default value of 22 for the *Port number*. Into the field titled *User name:*, insert the name of the account you configured SSH key-based authentication for on the SIEM VM (in my case, the user's name was "ayy").



File protocol: SFTP
Host name: 172.16.1.3
Port number: 22
User name: ayy
Password:

You may have noticed that I haven't mentioned the *Password:* field in WinSCP yet. That's because we aren't going to use it. Instead, after entering the name of the user you enabled key-based authentication for, click on the *Advanced…* button. The *Advanced Site Settings* window pops up.

On the left pane, in the section titled *SSH*, select the entry named *Authentication*. In the middle of the window, find the *Authentication parameters* section. On the right side of the input box below *Private key file:*, click on the gray button with the ellipses in it. In the Explorer window that appears, navigate to the folder containing the PRIVATE SSH key that corresponds to the public key already set up on the host. Select this private key file to return to the Advanced Site Settings window, then click OK to close it.


Advanced Site Settings

- Environment
  - Directories
  - Recycle bin
  - SFTP
  - Shell
- Connection
  - Proxy
  - Tunnel
- SSH
  - Key exchange
  - Authentication
  - Bugs
- Note

[ ] Bypass authentication entirely

Authentication options

[✓] Attempt authentication using Pageant

[✓] Attempt 'keyboard-interactive' authentication

[✓] Respond with password to the first prompt

[ ] Attempt TIS or CryptoCard authentication (SSH-1)

Authentication parameters

[ ] Allow agent forwarding

Private key file:

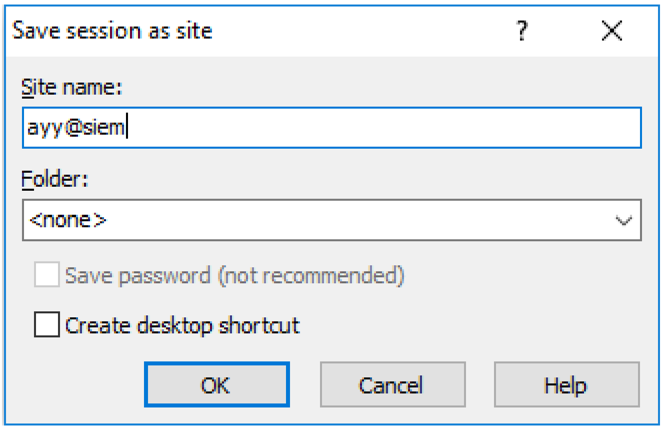
D:\keys\windows-management.ppk

GSSAPI

[ ] Attempt GSSAPI authentication

[ ] Allow GSSAPI credential delegation

Back in the Login window, click the Save button. In the window captioned *Save session as site* that appears, change the suggested value in the box underneath *Site name:* to "*[username]*@siem". Using this pattern for the site names can make it easier to differentiate which session maps to which lab virtual machine. When you are done, click *OK* to finish.


Save session as site

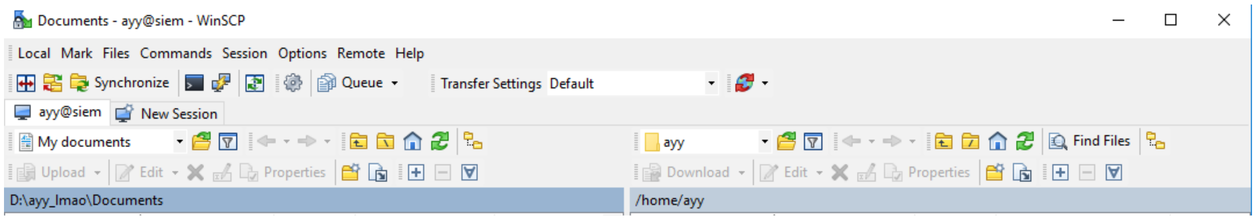
Site name: ayy@siem

Folder: <none>

Save password (not recommended)

Create desktop shortcut

A new saved session appears on the left pane in the Login window. Double click on this new entry. If everything works as expected, an SCP session to the SIEM VM is established.


Documents - ayy@siem - WinSCP

D:\ayy_lmao\Documents

/home/ayy

Linux, BSD, and OS X Remote Access

This guide is dedicated to those running a hosted hypervisor on Linux, BSD, or OS X, or planning to use a workstation running one of the aforementioned operating systems to establish network access to the lab environment as required (e.g. for both, bare-metal and/or hosted hypervisor networking).

The following sections include instructions on how to configure static routes and how to make them persist between reboots, on applications and commands I recommend to make remote management even easier, and finally on how to generate and use SSH keys for fast, easy and secure connectivity to the Linux lab VMs.

Static Routes in Linux and OS X

Almost all Linux and BSD systems (yes, that includes OS X) come with the `route` command pre-installed for managing the system's network routing table. However, the syntax differs slightly between Linux and BSD variants. Additionally, Linux distros in general are deprecating the `route` command in favor of the `ip` command, specifically `ip route add`.

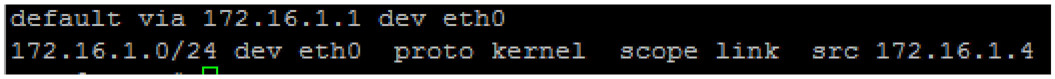
For this reason, we'll discuss using the `ip` command for adding static routes on Linux systems, while the guide for BSD and OS X on this matter is going to focus on using the respective version of the `route` command.

Adding Routes to Linux with the `ip` Command

The `ip` command is considered the preferred way to request network information on modern Linux systems, as well as add new routes to the system's network routing table. Entering the command:

```
ip route show
```

displays the current contents of the OS' routing table.

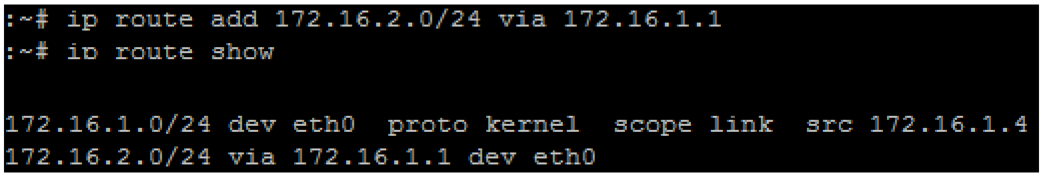


```
default via 172.16.1.1 dev eth0
172.16.1.0/24 dev eth0  proto kernel  scope link  src 172.16.1.4
```

The output in the image above indicates that the default gateway is 172.16.1.1 (the IP address of the *LAN* interface of the pfSense VM, connected to the *Management* network). The default gateway is the address a host will forward any traffic to that isn't sent to target machines on its local network, in hopes that the gateway knows where to relay this data to. Another thing that can be learned from the output is that the machine is connected directly to the 172.16.1.0/24 network through its `eth0` interface.

Let's assume we are using a Linux workstation as our hypervisor host, and we want to access the 172.16.2.0/24 network (the *IPS* network) from our virtual network adapter connected to the 172.16.1.0/24 network (the *Management* network). To update the system's routing informations accordingly and verify the configuration, enter the following commands:

```
ip route add 172.16.2.0/24 via 172.16.1.1
ip route show
```



```
:~# ip route add 172.16.2.0/24 via 172.16.1.1
:~# ip route show

172.16.1.0/24 dev eth0  proto kernel  scope link  src 172.16.1.4
172.16.2.0/24 via 172.16.1.1 dev eth0
```

As you can see in the image above, the output produced by `ip route show` includes `172.16.2.0/24 via 172.16.1.1 dev eth0`, indicating that the routing information has been added successfully. The device name (`eth0`) may differ, depending on the version and distribution of Linux being run, but other than that, the output should be consistent.

Please bear in mind, that while you may have a static route to the `172.16.2.0/24` network in the routing table, that in order to establish a connection to systems on that network, you also have to have firewalls rules configured on pfSense VM to allow traffic from the *Management* network to the *LAN* network, and that the system you want to connect to must also be running the network service you want to interact with.

Adding Routes to OS X/BSD with the `route` command

OS X provides the `route` command to add new static routes on-demand. Compared to the version formerly used on Linux however, the adaption that comes with OS X behaves a little bit differently, mostly because it has been inherited from BSD. With that in mind, let's run through how to add a static route to OS X or BSD using the `route` command.

Let's assume we are using a OS X or BSD workstation as our hypervisor host, and we want to access the 172.16.2.0/24 network (the *IPS* network) from our virtual network adapter connected to the 172.16.1.0/24 network (the *Management* network). To update the system's routing informations accordingly and verify the configuration, enter the following commands

```
route add 172.16.2.0/24 172.16.1.1
```

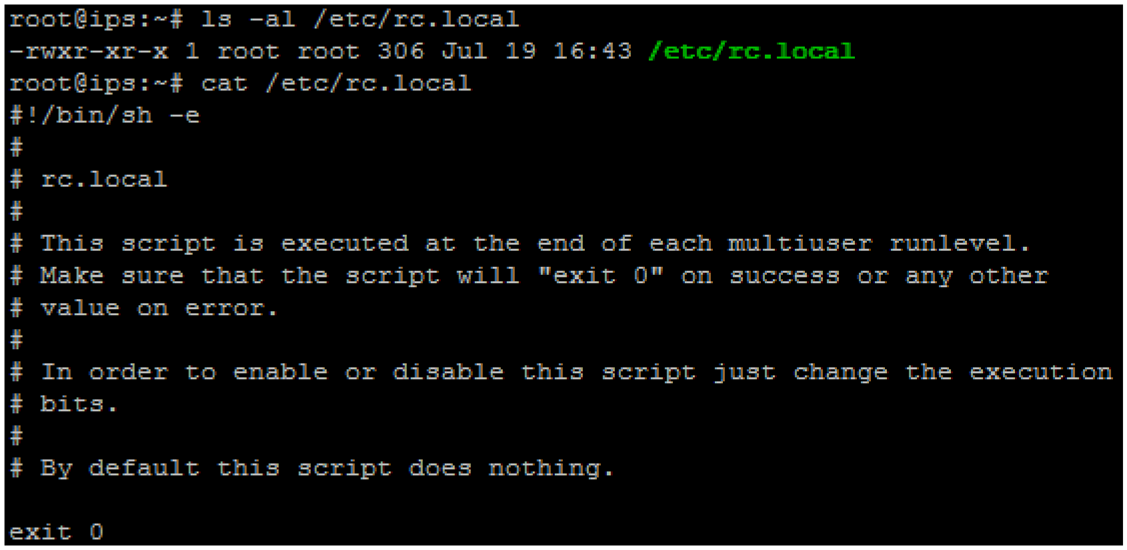
This tells the system to send any packets destined for the 172.16.2.0/24 network to 172.16.1.1, in order to have them forwarded correctly. Please bear in mind, that while you may have a static route to the `172.16.2.0/24` network in the routing table, that in order to establish a connection to systems on that network, you also have to have firewalls rules configured on pfSense VM to allow traffic from the *Management* network to the *LAN* network, and that the system you want to connect to must also be running the network service you want to interact with.

Making Static Routes Persistent

It's one thing to know how to add a static route to Linux, BSD or OS X, but as soon as the system is rebooted, the manually created entries in the routing table are lost. This section details ways to make the static routes we created persistent, as well as some workarounds for quirks that I discovered in VMware Fusion on OS X.

Linux and BSD Route Persistence via `/etc/rc.local`

On most Linux and BSD systems there is a file in `/etc` called `rc.local`. This file is a way to instruct the OS, "On startup, After all the other services on the system are executed, I want you to run these commands." We can use this file to add our static routes every time the system is (re)booted. While using `rc.local` as a persistence mechanism is considered lazy, it serves our purpose perfectly.



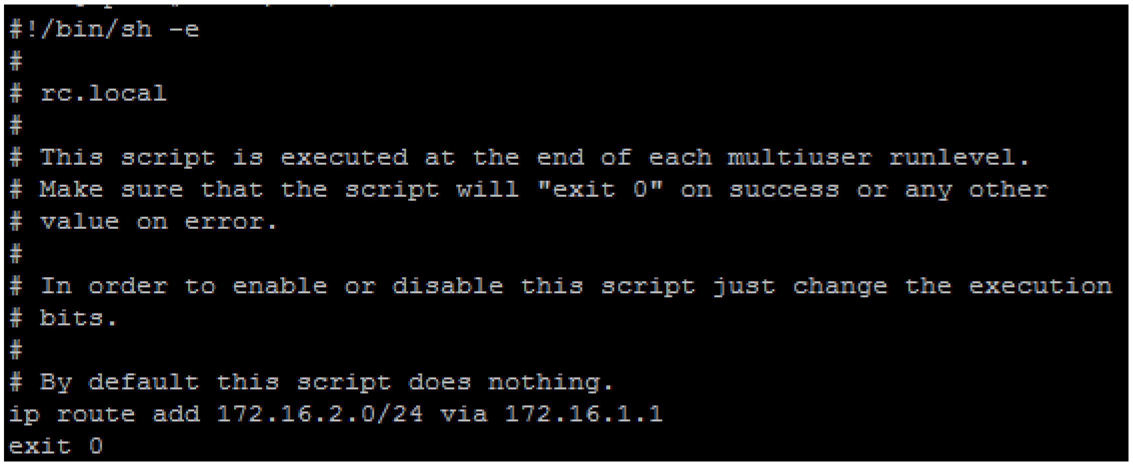
```
root@ips:~# ls -al /etc/rc.local
-rwxr-xr-x 1 root root 306 Jul 19 16:43 /etc/rc.local
root@ips:~# cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

exit 0
```

In the illustration above, I ran `ls -al /etc/rc.local`, then I used the `cat` command to display the contents of `rc.local`. This is to highlight a couple of important features of this file:

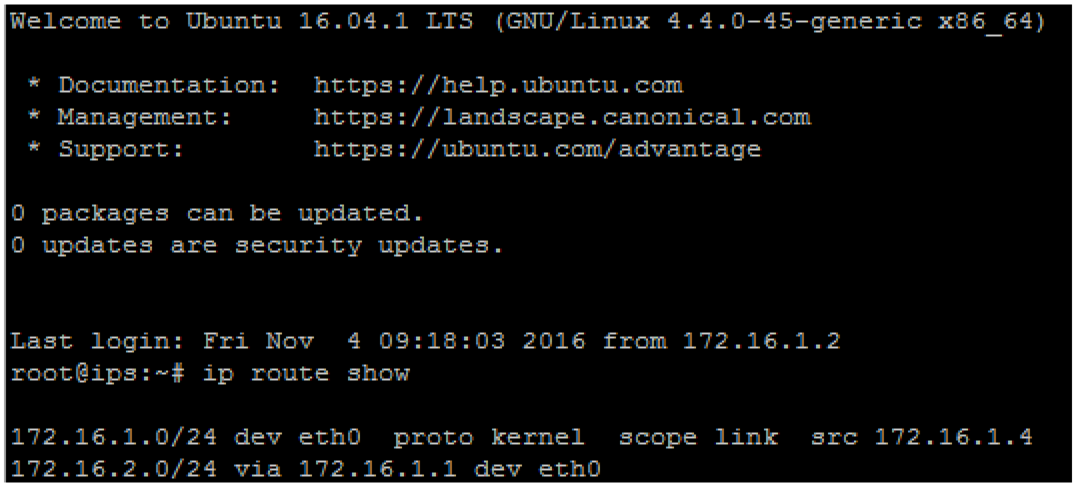
1. The file should be owned by root.

2. The file needs execute permissions for the root user at a minimum, in order for the system to run it and perform the commands contained within.
3. Given the conditions listed above are met, the script does nothing by default, but run “`exit 0`” to exit. Any commands you want `rc.local` to execute has to be placed ABOVE the `exit 0` line.
4. If you have to create the file from scratch, it must begin with the `#!/bin/sh` line, and end with the `exit 0` statement.

By default, on Ubuntu Linux, `rc.local` exists, has the execute permissions set, and looks exactly like the output in the image above. To equip the system with persistent static routes, all you have to do is insert any “`ip route add`” statements above the “`exit 0`” line. After editing it, the file should look similar to the example displayed in the picture below:


```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
ip route add 172.16.2.0/24 via 172.16.1.1
exit 0
```

What if you need to access your virtual networks hosted on a bare-metal hypervisor, and you need to add more than one route? Simply add the necessary instructions to the `/etc/rc.local` file. Make sure to add them above the `exit 0` line, one command per line. When finished, save the updated file, then reboot the Linux or BSD host. If you entered the `ip route` command into `rc.local` correctly, you should have a static route to 172.16.2.0/24 in your routing table.

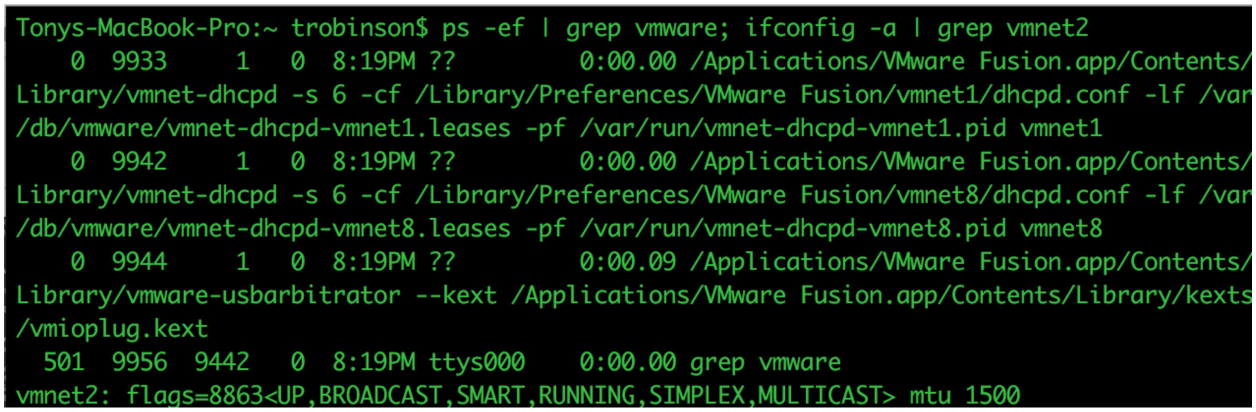


```
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Fri Nov  4 09:18:03 2016 from 172.16.1.2
root@ips:~# ip route show

172.16.1.0/24 dev eth0  proto kernel  scope link  src 172.16.1.4
172.16.2.0/24 via 172.16.1.1 dev eth0
```

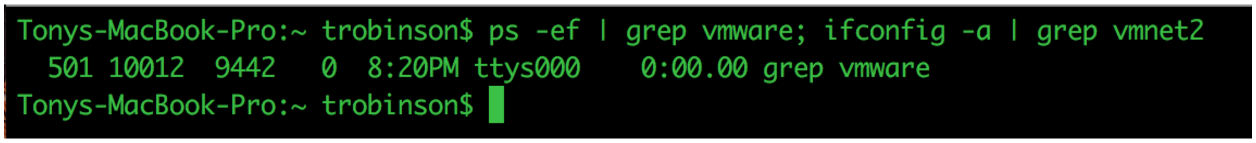
Note: For BSD users, the process of using `/etc/rc.local` to add persistent static routes to the systems is more or less the same as the one described above for a (current) Linux distro. The only difference is that you will have to use BSD's `route` command instead of `ip route add`. Also be aware that the approach of simply editing the `/etc/rc.local` file doesn't exactly work for OS X systems. You can find out more about this in the following section.

OS X Route Persistence with Hosted Hypervisors

While I was working on this guide, I discovered a unique issue running hosted hypervisors on OS X (e.g. VMware Fusion Pro and/or Oracle Virtualbox). The problem is, that if the hosted hypervisor is not running, the virtual interfaces and networks that one creates in the application simply cease to exist in OS X – until the hypervisor is started again. This is why every time you restart VMware Fusion Pro, or reboot your Mac, you have to run `sudo ifconfig vmnet2` in order to re-set the IP address on this network interface for the *Management* virtual network. Don't understand? Pay attention:



```
Tonys-MacBook-Pro:~ trobinson$ ps -ef | grep vmware; ifconfig -a | grep vmnet2
    0  9933     1   0  8:19PM ??         0:00.00 /Applications/VMware Fusion.app/Contents/
Library/vmnet-dhcpd -s 6 -cf /Library/Preferences/VMware Fusion/vmnet1/dhcpd.conf -lf /var
/db/vmware/vmnet-dhcpd-vmnet1.leases -pf /var/run/vmnet-dhcpd-vmnet1.pid vmnet1
    0  9942     1   0  8:19PM ??         0:00.00 /Applications/VMware Fusion.app/Contents/
Library/vmnet-dhcpd -s 6 -cf /Library/Preferences/VMware Fusion/vmnet8/dhcpd.conf -lf /var
/db/vmware/vmnet-dhcpd-vmnet8.leases -pf /var/run/vmnet-dhcpd-vmnet8.pid vmnet8
    0  9944     1   0  8:19PM ??         0:00.09 /Applications/VMware Fusion.app/Contents/
Library/vmware-usbarbitrator --kext /Applications/VMware Fusion.app/Contents/Library/kexts
/vmioplug.kext
  501  9956  9442   0  8:19PM ttys000    0:00.00 grep vmware
vmnet2: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
```



```
Tonys-MacBook-Pro:~ trobinson$ ps -ef | grep vmware; ifconfig -a | grep vmnet2
  501 10012  9442   0  8:20PM ttys000    0:00.00 grep vmware
Tonys-MacBook-Pro:~ trobinson$ 
```

These two illustrations above are showing the results of running the following set of commands:

```
ps -ef | grep vmware;
ifconfig -a | grep vmnet2
```

The first command helps us to find out if there are any processes running that are related to `vmware`. From the first image we can tell that VMware Fusion is running. The second part of the command set shows us any lines in `ifconfig -a` that have the text `vmnet2`, which is the interface we used to have our hypervisor host connect to the *Management* virtual network. We can see in that very last line of output displayed in the first picture that `vmnet2` does exist, and that OS X recognizes the virtual network interface.

From what is shown in the second image,, we confirm that VMware Fusion is not running, as there appear to be no processes running related to `vmware`. The query `ifconfig -a | grep vmnet2` returns no output either; `vmnet2` has simply disappeared.

There is no way creating routes for networks and interfaces that do not exist. The `route` command requires OS X to know how to reach a particularly gateway before we can send packets to it destined for other networks. So, how do we work around this issue? We're going to create a simple shell script that is meant to be run each time after starting VMware Fusion Pro.

`flightcheck.sh`

I have created a shell script that helps us to deal with VMware Fusion's quirk that virtual network interfaces are removed from the system on exiting the program and recreated with default settings on startup. It actually takes care of two settings: adjusting the IP address of the `vmnet2` interface, and creating a route to the 172.16.2.0 network. Please note that this script could easily be adapted to Oracle Virtualbox (e.g. `vboxnet0` vs. `vmnet2` with VMware Fusion)

Log in to OS X and open up your favorite text editor. (I prefer to open up the terminal application and use vi.) Create a file in your home directory (`~/`) called `flightcheck.sh`, and input the following:

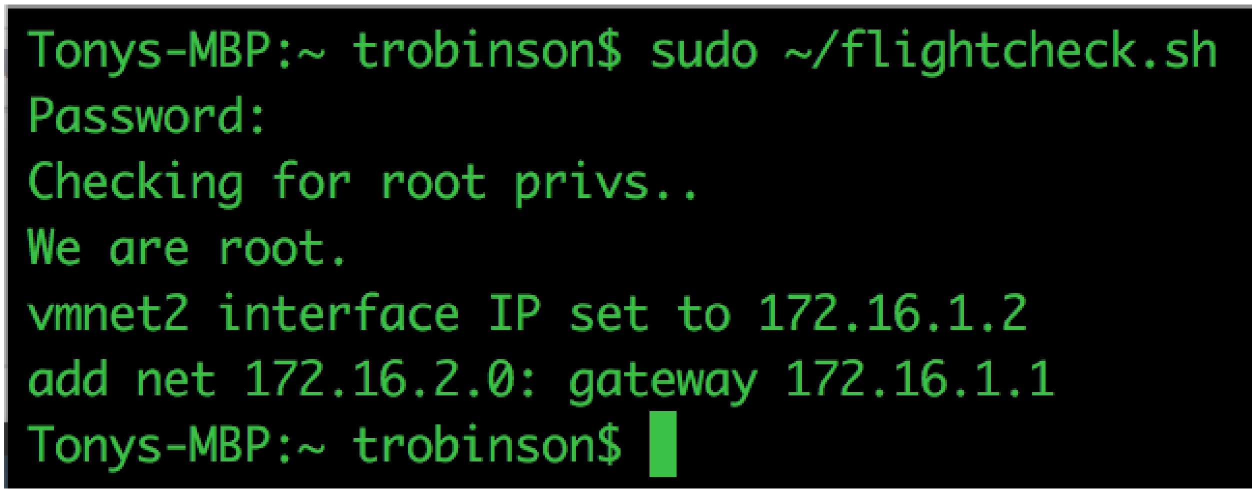
```
#!/bin/bash
#This script is meant for VMware Fusion Professional users.
```

```
#This script assumes that vmnet2 exists and is configured to be our
management virtual network.
echo "Checking for root privs.."
if [ $(whoami) != "root" ]; then
      echo "This script must be ran with sudo or root privileges."
      exit 1
else
      echo "We are root."
fi
ifconfig vmnet2 172.16.1.2 netmask 255.255.255.0
if [ $? -eq 0 ]; then
      echo “vmnet2 interface IP set to 172.16.1.2”
else
      echo “Could not set IP address of vmnet2 interface. Does vmnet2 exist?
Is VMware Fusion running?”
      exit 1
fi
route add 172.16.2.0/24 172.16.1.1
if [ $? -ne 0 ]; then
      echo “Could not create route to 172.16.2.0/24. Does the network exist?
Is VMWare Fusion running? Is the pfSense/Gateway VM running?”
      exit 1
fi
exit 0
```

Save the script when you are done, and change the file to have execution permissions at a minimum (e.g `chmod 700 ~/flightcheck.sh`). Start up VMware Fusion, and then boot up all of your lab’s virtual machines. As soon as all the VMs are ready, open a terminal on your Mac, and run the following command:

```
sudo ~/flightcheck.sh
```

You should be prompted to enter your password. We are using `sudo` to call the script to make sure it is executed with root privileges. If we wouldn’t do so, none of the commands inside the file would work. In the picture below you can see what it looks like when the script completes successfully:



```
Tonys-MBP:~ trobinson$ sudo ~/flightcheck.sh
Password:
Checking for root privs..
We are root.
vmnet2 interface IP set to 172.16.1.2
add net 172.16.2.0: gateway 172.16.1.1
Tonys-MBP:~ trobinson$
```

OS X route persistence for Bare-metal Hypervisors

As we know from years and years of Apple marketing, they like to *think different*. That different thinking doesn't necessarily result in better solutions, but just researching the Apple way of doing things. Apple uses BSD-like core components, but a lot of the functionality built into OS X is *entirely different* from BSD or Unix-like operating systems, leading to some difficulty.

I tried a number of tricks, picked up over the years of doing systems administration of Unix-like and Linux systems, that I was led to believe might actually work in OS X for providing route persistence in some sort of a fully automated manner on reboot. As previously established, using `rc.local` doesn't work on OS X. I tried creating a crontab entry as root to run a script at boot, but that failed. THEN I tried making a `launchd` (Launch Daemon) script to run a shell script on boot, but THAT failed too. It turns out that running commands as root to rebuild a routing table in OS X on boot time is complicated. Along with the fact that I don't know nearly enough about the dark inner workings of OS X to try and fix this, I decided to settle for the following workaround.

I made a modified version of the `flightcheck.sh` script we used for setting up networking in the case of hosted hypervisors running locally on OS X. This modified script won't automatically run when the system is booted, but I've provided instructions on how to execute it on-demand to fix your static routes as necessary. If you can figure out how to either run this script or adjust the system's routing information automatically at boot time as the root user, with no need to enter a password, feel free to do so.

`flightcheckBM.sh`

Use your favorite text editor to create the file `flightcheckBM.sh` in your home directory (e.g. `~/flightcheckBM.sh`), with the following content:

```
#!/bin/bash
#this script can help automate building static routes to virtual networks
hosted on a bare-metal hypervisor.
echo "Checking for root privs.."
if [ $(whoami) != "root" ]; then
      echo "This script must be ran with sudo or root privileges."
      exit 1
else
      echo "We are root."
fi
route add 172.16.1.0/24 192.168.1.22
if [ $? -ne 0 ]; then
      echo “Could not create route to 172.16.1.0/24. Does the network exist?
Is ESXi running? Is the pfSense/Gateway VM running?”
      exit 1
fi
route add 172.16.2.0/24 192.168.1.22
if [ $? -ne 0 ]; then
      echo “Could not create route to 172.16.2.0/24. Does the network exist?
Is ESXi running? Is the pfSense/Gateway VM running?”
      exit 1
fi
exit 0
```

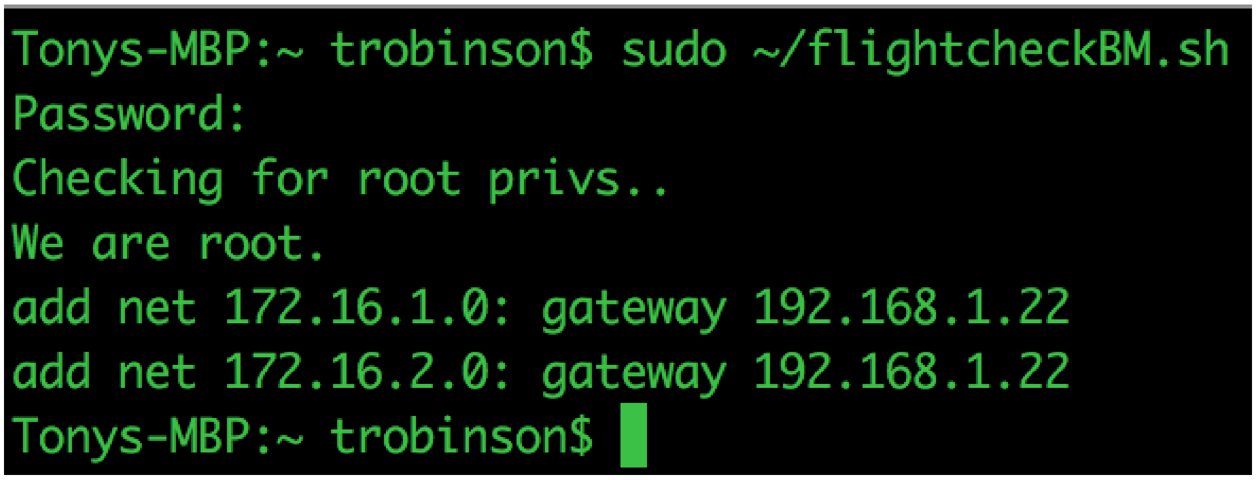
Save the script when you are done, and change the file to have execution permissions at a minimum (e.g `chmod 700 ~/flightcheckBM.sh`).

Note: 192.168.1.22 was the IP address of the pfSense WAN interface on my local network. You will (almost certainly) need to adjust the respective `route` commands in your version of the script to meet the conditions of your home or office network.

When you are ready, open a terminal, and run the command below:

```
sudo ~/flightcheckBM.sh
```

You should be prompted to enter your password. We are using `sudo` to call the script to make sure it is executed as the root user; only root can modify the routing table on our system. The picture below shows what it looks like when the script completes successfully:



```
Tonys-MBP:~ trobinson$ sudo ~/flightcheckBM.sh
Password:
Checking for root privs..
We are root.
add net 172.16.1.0: gateway 192.168.1.22
add net 172.16.2.0: gateway 192.168.1.22
Tonys-MBP:~ trobinson$ 
```

The ssh and scp terminal Applications

Most Unix/Linux systems come with the command line applications `ssh` and `scp,` installed along with the operating system's core utilities by default. Being a minimalist at heart, I like to live off the land and use a system's standard applications when and if possible, so employing the `ssh` and `scp` commands suits me just fine. For the novice user, unfamiliar with the Unix/Linux command-line environment, the syntax for `ssh` and/or `scp` may take some getting used to, but like with everything else, practice (and time) makes perfect.

A program's `man` pages usually provide all the information needed for using the tool, including all of the unique options available for advanced operations. For sake of completion however, I will show you how to interact with the lab's SIEM VM using the `ssh` and `scp` command line clients. For this exercise, we are going to assume that you are using a hosted hypervisor on Linux, BSD, or OS X system, with a virtual network card connected to the Management network, its IP address set to 172.16.1.2. If you are working with a lab based on a hosted hypervisor, the examples below will work too, provided you have static routes and firewall rules configured accordingly.

To start an SSH session to the SIEM VM, all you need to do is to open a shell (or terminal), then type (and execute):

```
ssh 172.16.1.3
```

Upon connection, the server will ask for the name of the user you wish to login as, and/or what password you'd like to use. To make things slightly faster, your basic `ssh` command can be modified to:

```
ssh [username]@172.16.1.3
```

In the command above, replace [username] with the name of the user you wish to log in as. For instance, if I want to log in as the user "ayy", the command is:

```
ssh ayy@172.16.1.3
```

If done correctly, you should be prompted to enter a password for the user you are trying to log in as. If you entered the username and password correctly, you're greeted with a shell session on the remote system.

The `scp` program allows you to transfer files to and from a distant system. The syntax varies slightly, depending on whether you're attempting to download or to send a file. Let's have a look at the following instruction:

```
scp /Users/lmao/data.txt ayy@172.16.1.3:/home/ayy/data.txt
```

In the example above, we are attempting to send the file `data.txt`, located in the local directory `/Users/lmao`, as the user "ayy" to the remote host at 172.16.1.3, and store it in `/home/ayy/data.txt`. If the connection to 172.16.1.3 is successful, `scp` will ask for the password for the user "ayy". If entered correctly, the file will be transferred. Now that you have an idea how to upload a file, here's an example how to do a download:

```
scp ayy@172.16.1.3:/home/ayy/results.pdf /Users/lmao/results.pdf
```

With the command above, we're attempting to copy the remote file `results.pdf` file from the directory `/home/ayy` on the host at 172.16.1.3, authenticating as the user "ayy", and save it to `/Users/lmao/results.pdf` on the local machine.

iTerm2 and Terminator

While typically the `scp` and `ssh` commands are enough to suit my needs when working on Unix/Linux systems, there is a pair of terminal programs -- one for OS X, another one for Linux systems -- that I highly recommend, mostly for the wonderful features they have.

The OS X application iTerm2 is meant to replace the Mac's built-in Terminal utility. While there are a lot of neat features that I don't work with on a regular basis, the ones I use (and love) include the capacity to create multiple tabs for different terminal sessions, and the ability to split a single window/tab vertically and/or horizontally, to fit multiple SSH sessions into a single application window. Visit https://www.iterm2.com/ if you're interested.

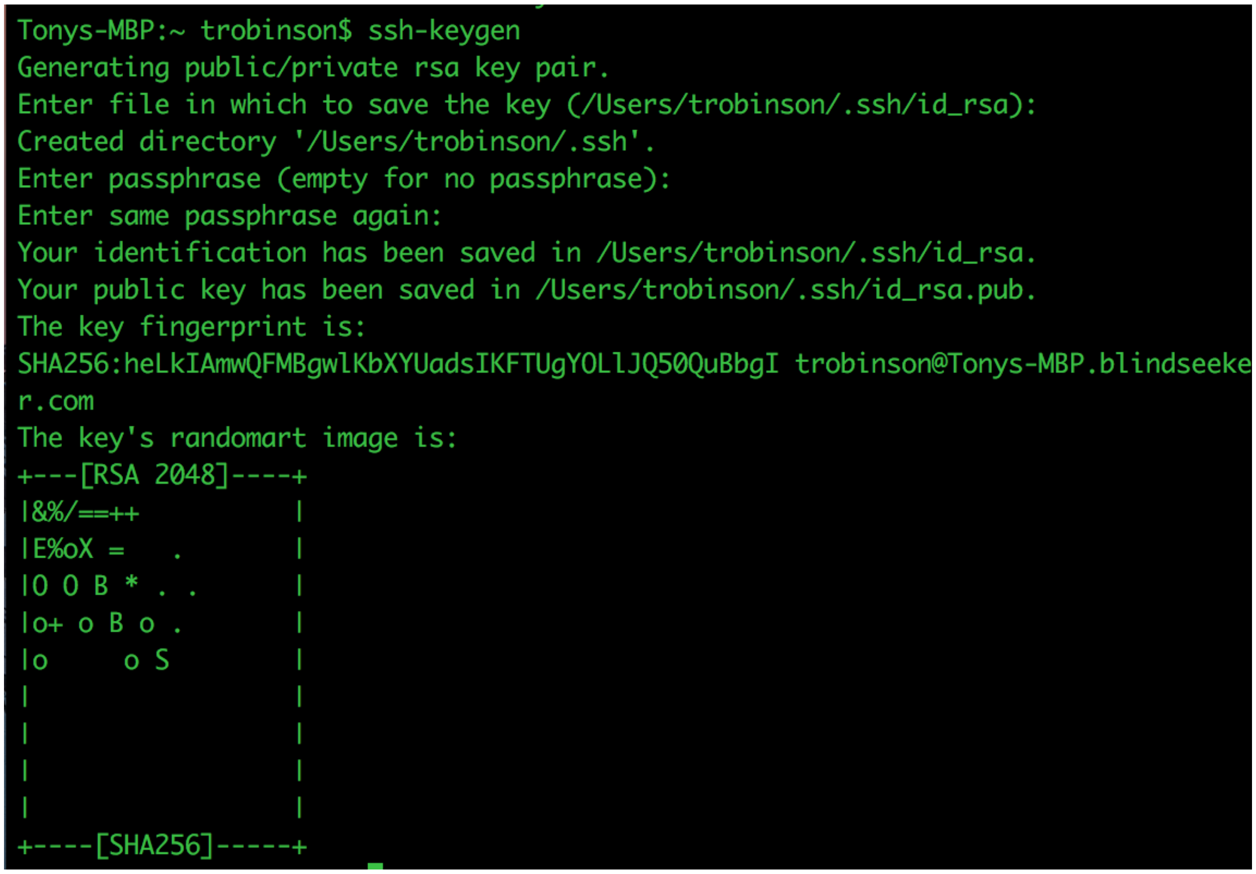
Terminator is essentially the Unix/Linux equivalent for iTerm2 on OS X, with a comparable featureset. Most current distros come with Terminator as a part of their included packages (I know for certain that Kali Linux and Ubuntu provide it via apt/aptitude by default, if nothing else). Find out more at https://gnometerminator.blogspot.com/p/introduction.html.

Generating ssh keys using ssh-keygen

In addition to having SSH and SCP clients available as core functionality, most Linux, Unix, and BSD variants also provide the key generation program `ssh-keygen`. This utility is used to generate public and private keys, which can serve as a method of authentication for SSH connections. Given the SSH server has been configured accordingly, it would allow and try to authenticate the user attempting to log in against the corresponding local copy of the public key t instead of prompting for a password. Additionally, putting a password on your SSH private key, this password would serve as a second factor of authentication when connecting to the SSH server in question.

Running `ssh-keygen` is as simple as just typing and executing

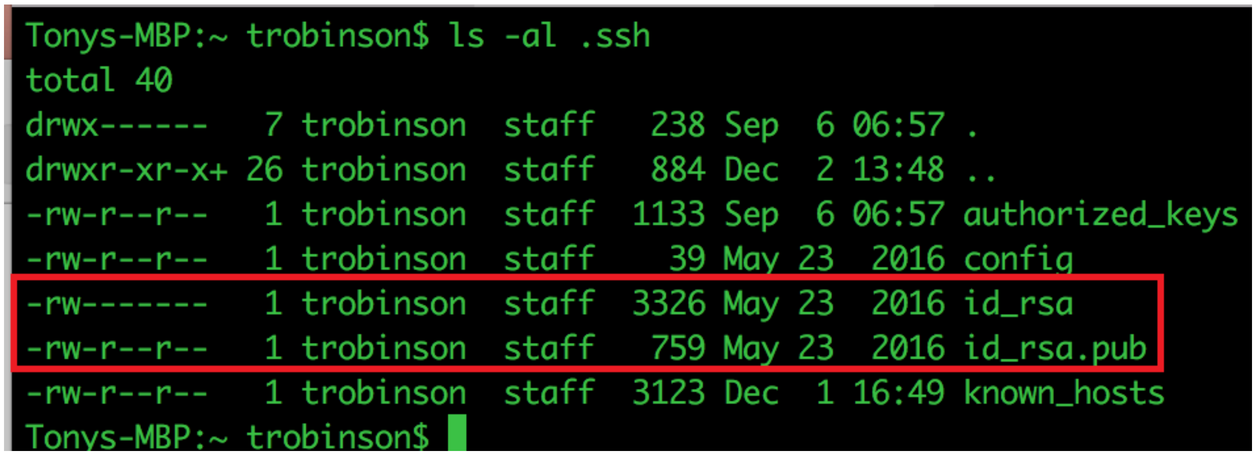
```
ssh-keygen
```



```
Tonys-MBP:~ trobinson$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/trobinson/.ssh/id_rsa):
Created directory '/Users/trobinson/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/trobinson/.ssh/id_rsa.
Your public key has been saved in /Users/trobinson/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:heLkIAmwQFMBgwlKbXYUadsIKFTUgYOLlJQ50QuBbgI trobinson@Tonys-MBP.blindseeke
r.com
The key's randomart image is:
+---[RSA 2048]----+
|&%/==++          |
|E%oX =    .      |
|O O B * . .      |
|o+ o B o .       |
|o     o S        |
|                 |
|                 |
|                 |
|                 |
+----[SHA256]-----+
```

By default, when running `ssh-keygen`, the folder `.ssh` is created in the current user's home directory (if it doesn't already exist), and the two newly generated files `id_rsa` and `id_rsa.pub` are stored there. The file `id_rsa` holds the private key, while the file `id_rsa.pub` on the other hand, has the public key. Unless specified otherwise, `ssh-keygen` will generate RSA keys that are 2048 bits in length. If you would like to generate stronger keys, increase this value, e.g. to 4096, by using the option "`-b 4096`" when running `ssh-keygen`.

The public key information contained in `id_rsa.pub` has to be copied to any system you want the respective user to log on to using SSH key-based authentication. The contents of this file need to be copied to to file "`authorized_keys`" in the `.ssh` directory of the user you want to use key-based authentication for. I'll be demonstrating a couple of ways you can do this in just a moment, but for now, run `ls -al ~/.ssh,` and verify that both files, `id_rsa` and `id_rsa.pub,` have been generated on your workstation.

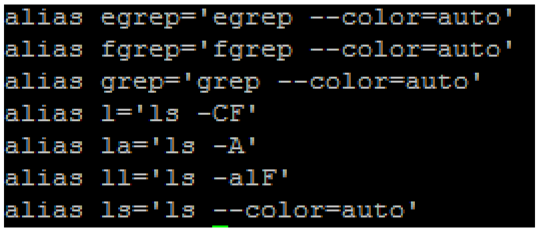


```
Tonys-MBP:~ trobinson$ ls -al .ssh
total 40
drwx------   7 trobinson  staff   238 Sep  6 06:57 .
drwxr-xr-x+ 26 trobinson  staff   884 Dec  2 13:48 ..
-rw-r--r--   1 trobinson  staff  1133 Sep  6 06:57 authorized_keys
-rw-r--r--   1 trobinson  staff    39 May 23  2016 config
-rw-------   1 trobinson  staff  3326 May 23  2016 id_rsa
-rw-r--r--   1 trobinson  staff   759 May 23  2016 id_rsa.pub
-rw-r--r--   1 trobinson  staff  3123 Dec  1 16:49 known_hosts
Tonys-MBP:~ trobinson$
```

In contrast to the public key, the data of the private one, stored in the `id_rsa` file, must be kept secure and private at all costs. When you run `ssh-keygen`, the application will ask you if you wish to set a password for your SSH private key. Leaving the password blank is an option, with the result that you don't have to enter a password each time you use this key for authentication. If you don't set a password for the private key, however, anyone who acquires your `id_rsa` file could use it to successfully log in via SSH to any system you set up key-based authentication on for the corresponding user. In other words, if someone manages to get a hold of said file, from a computer system's point of view, they can essentially become you. Please keep this in mind when considering not setting a password on your `id_rsa` file.

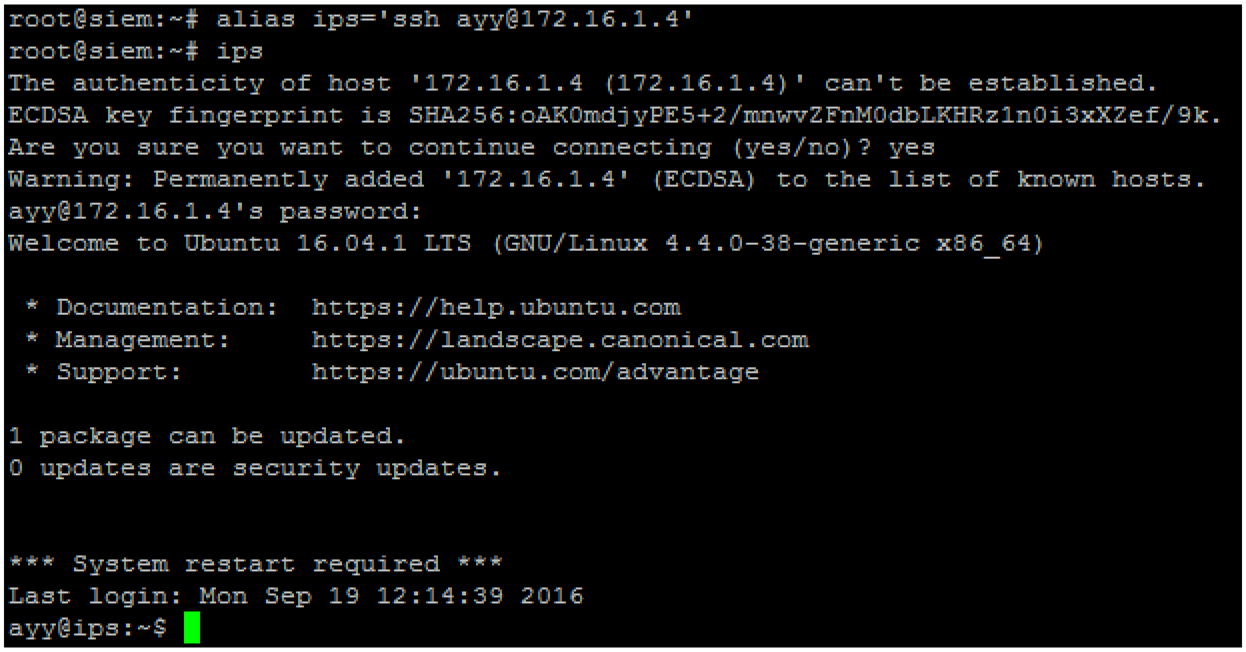
The alias Command

The `alias` command, like most the commands mentioned so far, is available by default in most Linux, Unix and BSD systems. What `alias` allows you to do is to define a sort of "shortcut" for running commands with certain options. For example, entering the command `alias` in an SSH session on the SIEM VM returns the list of aliases already configured on the remote system by default.



```
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
```

So, what could be so interesting about this command? The `alias` command saves the user (and the administrator, of course) from typing long and thus error-prone lines of commands and corresponding parameters. In our case, for instance, we can create aliased commands, telling the terminal/shell to open an SSH session to certain remote hosts.



```
root@siem:~# alias ips='ssh ayy@172.16.1.4'
root@siem:~# ips
The authenticity of host '172.16.1.4 (172.16.1.4)' can't be established.
ECDSA key fingerprint is SHA256:oAK0mdjyPE5+2/mnwvZFnM0dbLKHRz1n0i3xXZef/9k.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.1.4' (ECDSA) to the list of known hosts.
ayy@172.16.1.4's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

1 package can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Mon Sep 19 12:14:39 2016
ayy@ips:~$
```

In this way, we are able to build some sort of connection profiles, allowing us to more easily connect to our lab VMs. The problem with the alias above is that it is lost as soon as the terminal session is closed.

The way around this behavior, in order to set up persistent aliases, is to use the shell's resource files, part of which can be found in the user's home directory.

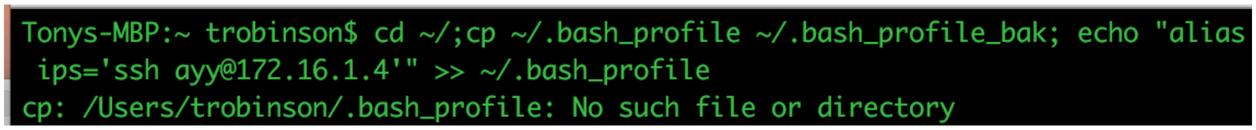
On many Linux and Unix systems, the BASH (Bourne Again SHell) is the default shell used for command-line interactions, whereas the standard on BSD derivatives usually is the KSH (KornSHell). This time, OS X is an exception, as it is normally configured to use the BASH. If you are running a Linux/Unix/BSD system where BASH is not the default, I highly recommend installing it, and making it the default shell for your user account. Consult documentation for the

OS of your choice on how to do this. As BASH is widely used, and for reasons of simplicity, I am going to show you how to create persistent aliases for this shell.

Usually, in each user's home directory, there is file called `.bashrc` or `.bash_profile`. (If neither of them exists in the home folder of the respective account, feel free to create either `.bashrc` or `.bash_profile`, using a text editor of your choosing.) These files are used to customize various settings in the BASH to a user's liking. Given that the BASH is the default shell for the system , and a user successfully logs in (or starts a terminal application within the graphical desktop environment), said files are automatically read (if present in the corresponding home folder), and their settings are applied accordingly.

As displayed in the screen capture above (which was taken on a system running Ubuntu Linux), there are already several aliases set up by default, most likely configured through `.bashrc` or `.bash_profile`. Consequently, aliases that are meant to be persistent can be defined in either file. One possible way to achieve this is by appending the alias to the end of either `.bashrc` or `.bash_profile`. Try entering and running the following command line:

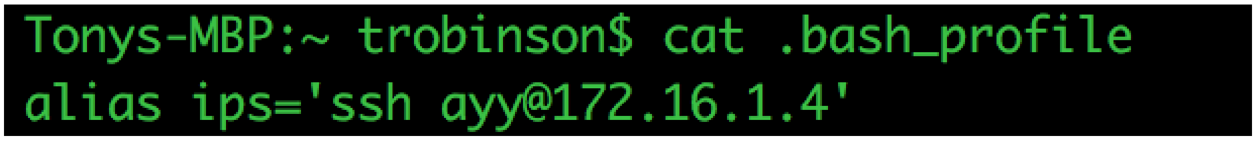
```
cd ~/;cp ~/.bash_profile ~/.bash_profile_bak;echo “alias ips=’ssh
ayy@172.16.1.4’” >> ~/.bash_profile
```



```
Tonys-MBP:~ trobinson$ cd ~/;cp ~/.bash_profile ~/.bash_profile_bak; echo "alias
 ips='ssh ayy@172.16.1.4'" >> ~/.bash_profile
cp: /Users/trobinson/.bash_profile: No such file or directory
```

This series of commands starts with a change of directories to the current user's home directory, then a copy of `.bash_profile` (if it exists) named `.bash_profile_bak is created.` (The latter done for the case we do something that “breaks” `.bash_profile`, allowing us to restore the file our backup.) Next, our alias command is appended to `.bash_profile`: “ips” is set to open an SSH session to 172.16.1.4 as the user “ayy”. (If “ayy” is not a user on your IPS VM, substitute the name with the one of the account you created.) In case the file `.bash_profile` doesn't exist yet, it is created and the alias is inserted. If everything went right, a line similar to the one shown in the picture below should be printed when running the following command (from the respective user's home directory):

```
cat .bash_profile
```



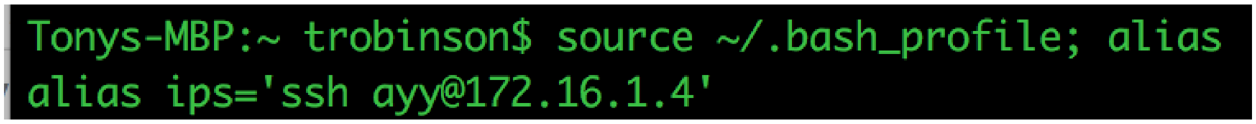
```
Tonys-MBP:~ trobinson$ cat .bash_profile
alias ips='ssh ayy@172.16.1.4'
```

Note: if `~/.bash_profile` does NOT exist, the command `cp ~/.bash_profile ~/.bash_profile_bak;` will print an error message stating `No such file or directory`.

There is no need to worry about this, especially if `cat ~/.bash_profile` displays the alias correctly.

To test the alias and the updated `.bash_profile`, try the following:

```
source ~/.bash_profile; alias
```



```
Tonys-MBP:~ trobinson$ source ~/.bash_profile; alias
alias ips='ssh ayy@172.16.1.4'
```

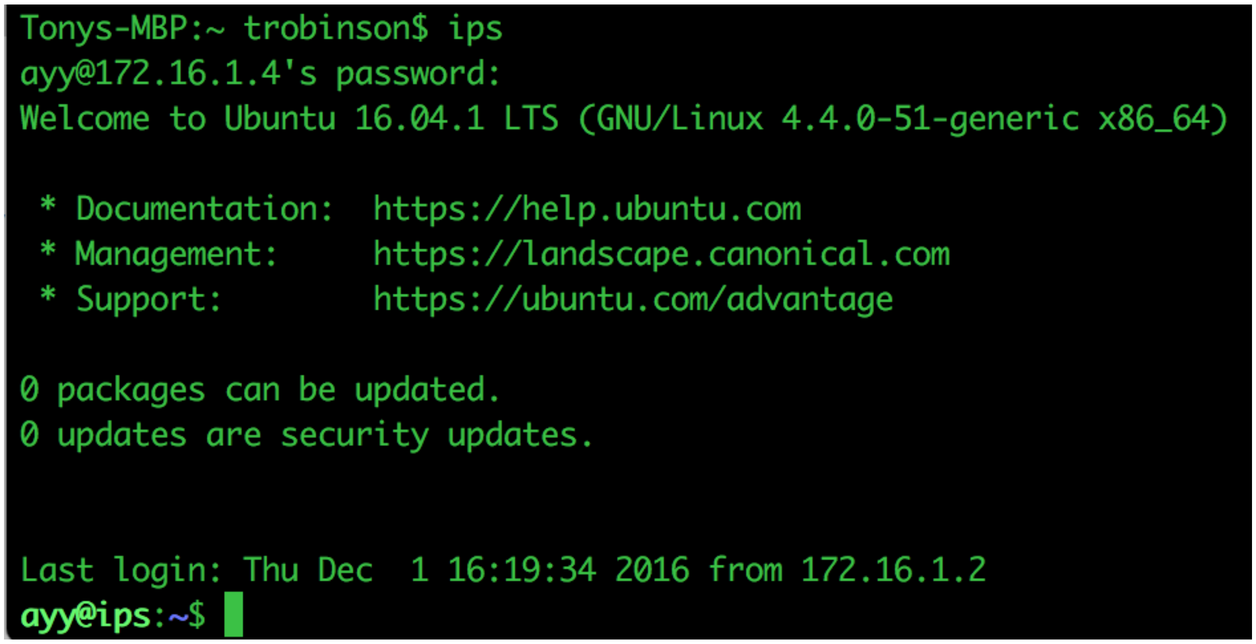
If you don't get any errors, and you see the new alias listed (among the other system default aliases, if there are any defined), then everything worked fine.

The `source` command tells the shell to read the contents of the file that has been provided as a parameter, and apply the settings found within to the current session. In a nutshell, this applies our configuration settings without forcing us to logout, if we don't want to.

To verify that the new aliased command actually establishes a proper SSH connection, simply type and execute

```
ips
```

As a result, you should be prompted for the password of the user that is meant to log in to the IPS VM. If entered correctly, you are supposed to be connected to the remote system via SSH.



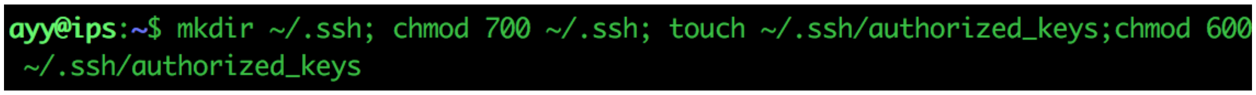
```
Tonys-MBP:~ trobinson$ ips
ayy@172.16.1.4's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Thu Dec  1 16:19:34 2016 from 172.16.1.2
ayy@ips:~$ 
```

Creating the corresponding aliases for the other virtual machines is fairly simple. Just re-use the instructions above, and, within the quotation marks, adjust the label of the alias, the username and the IP address of the remote system as required. The way the original command above is formatted, new aliases are always appended to the bottom of `.bash_profile`, if the file exists.

Enabling Key-Based Authentication in Unix/Linux Systems

If you haven't run `ssh-keygen` on your Linux, Unix, or OS X host yet, please do so now. In case that you're continuing on from the previous section, you should have an established SSH session to the IPS VM, logged in with the corresponding user account. Make sure that you are in said user's home directory; if you are not or when in doubt, run the command `cd ~/`. If you are not or if you are no longer connected to the remote, open up an SSH session to the IPS VM at 172.16.1.4. Once you are logged in, enter and run the following command:

```
mkdir ~/.ssh;chmod 700 ~/.ssh; touch ~/.ssh/authorized_keys;chmod 600
~/.ssh/authorized_keys
```

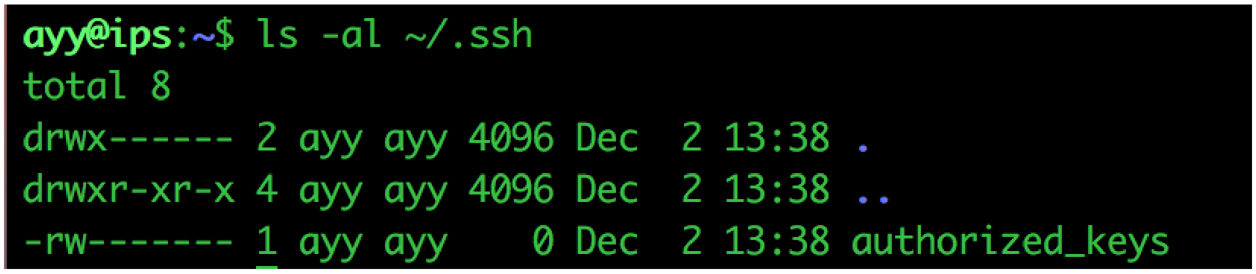


```
ayy@ips:~$ mkdir ~/.ssh; chmod 700 ~/.ssh; touch ~/.ssh/authorized_keys;chmod 600
 ~/.ssh/authorized_keys
```

To verify everything executed properly, execute the following command:

```
ls -al ~/.ssh
```

The resulting output should look similar to the one in the picture below:



```
ayy@ips:~$ ls -al ~/.ssh
total 8
drwx------ 2 ayy ayy 4096 Dec  2 13:38 .
drwxr-xr-x 4 ayy ayy 4096 Dec  2 13:38 ..
-rw------- 1 ayy ayy    0 Dec  2 13:38 authorized_keys
```

The first command was actually made up of four instructions in rapid succession. It created the hidden directory `.ssh`, gave this directory file permissions that grant access only to the user who created it, then generated the empty file `authorized_keys` with the `touch` command (if the file does NOT already exist, touch creates it), and finally changed the permissions of this file to only allow the account currently used read and write access.

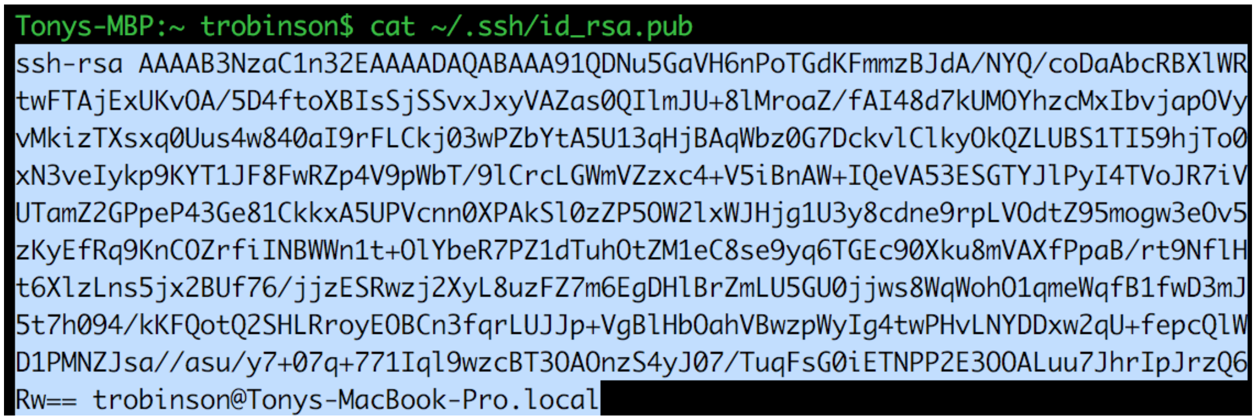
The next part is tricky, in that we have to insert the contents of `~/.ssh/id_rsa.pub` on our local machine into `~/.ssh/authorized_keys` in the user's home directory on the remote system. There are a several ways to do this. In the following, you can find three possible approaches.

Key Copy Method 1: echo append to authorized_keys

On your local machine, open `~/.ssh/id_rsa.pub` in your favorite graphical text editor, or, using the command line (from another terminal, or from a new tab,if you're using iTerm2/Terminator), run

```
cat ~/.ssh/id_rsa.pub
```

Highlight all of the text from the file, beginning to end, and hit Ctrl+C (or on a Mac, Command+C) to copy the selection to your clipboard.



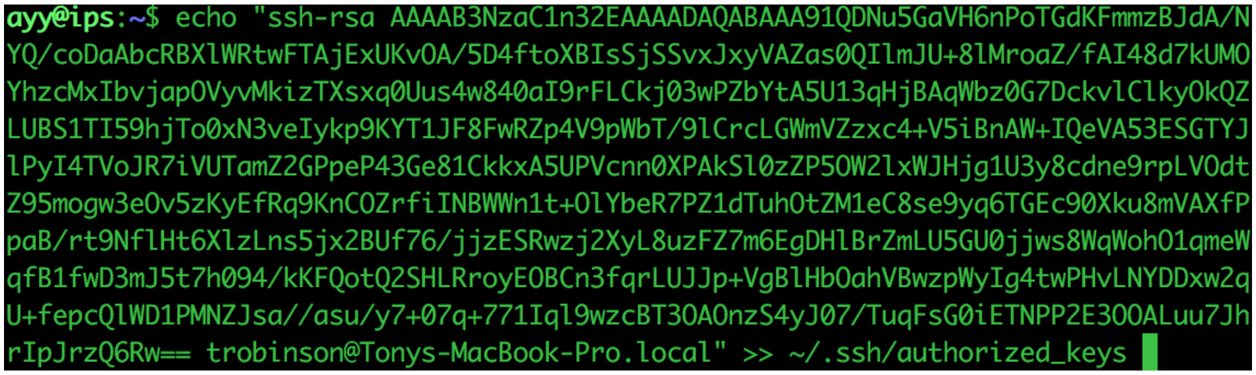
```
Tonys-MBP:~ trobinson$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1n32EAAAADAQABAAA91QDNu5GaVH6nPoTGdKFmmzBJdA/NYQ/coDaAbcRBXlWR
twFTAjExUKvOA/5D4ftoXBIsSjSSvxJxyVAZas0QIlmJU+8lMroaZ/fAI48d7kUMOYhzcMxIbvjapOVy
vMkizTXsxq0Uus4w840aI9rFLCkj03wPZbYtA5U13qHjBAqWbz0G7DckvlClkyOkQZLUBS1TI59hjTo0
xN3veIykp9KYT1JF8FwRZp4V9pWbT/9lCrcLGWmVZzxc4+V5iBnAW+IQeVA53ESGTYJlPyI4TVoJR7iV
UTamZ2GPpeP43Ge81CkkxA5UPVcnn0XPAkSl0zZP5OW2lxWJHjg1U3y8cdne9rpLVOdtZ95mogw3eOv5
zKyEfRq9KnCOZrfiINBWWn1t+OlYbeR7PZ1dTuhOtZM1eC8se9yq6TGEc90Xku8mVAXfPpaB/rt9NflH
t6XlzLns5jx2BUf76/jjzESRwzj2XyL8uzFZ7m6EgDHlBrZmLU5GU0jjws8WqWohO1qmeWqfB1fwD3mJ
5t7h094/kKFQotQ2SHLRroyEOBCn3fqrLUJJp+VgBlHbOahVBwzpWyIg4twPHvLNYDDxw2qU+fepcQlW
D1PMNZJsa//asu/y7+07q+771Iql9wzcBT3OAOnzS4yJ07/TuqFsG0iETNPP2E3OOALuu7JhrIpJrzQ6
Rw== trobinson@Tonys-MacBook-Pro.local
```

Back in our SSH session to the IPS VM, enter the following command:

```
echo "[ctrl + v  or meta + v (on a mac)]" >> ~/.ssh/authorized_keys
```

Be sure to include the double quotes in the command above!

When hitting ctrl+v (or in OS X meta + v), the current contents of your system's clipboard are (normally) inserted in the active session at the cursor's position. In our case, the content of `id_rsa.pub` should be filled in between the double quotes. The final portion of the command in the terminal instructs the system to append this (echoed) key data to the `~/.ssh/authorized_keys` file. If your (somewhat extended) command line looks similar to the one you can see in the picture below, execute it.

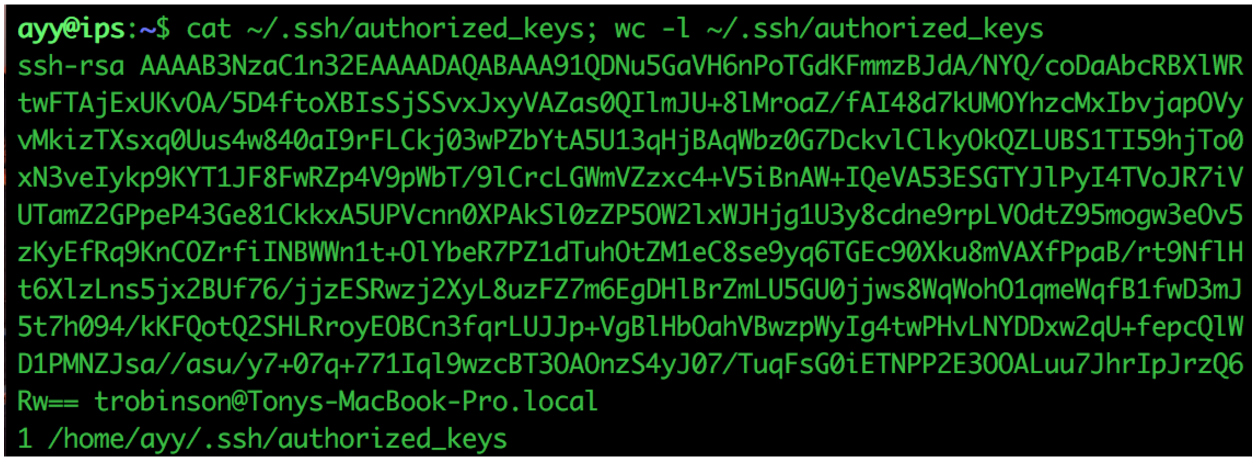


```
ayy@ips:~$ echo "ssh-rsa AAAAB3NzaC1n32EAAAADAQABAAA91QDNu5GaVH6nPoTGdKFmmzBJdA/N
YQ/coDaAbcRBXlWRtwFTAjExUKvOA/5D4ftoXBIsSjSSvxJxyVAZas0QIlmJU+8lMroaZ/fAI48d7kUMO
YhzcMxIbvjapOVyvMkizTXsxq0Uus4w840aI9rFLCkj03wPZbYtA5U13qHjBAqWbz0G7DckvlClkyOkQZ
LUBS1TI59hjTo0xN3veIykp9KYT1JF8FwRZp4V9pWbT/9lCrcLGWmVZzxc4+V5iBnAW+IQeVA53ESGTYJ
lPyI4TVoJR7iVUTamZ2GPpeP43Ge81CkkxA5UPVcnn0XPAkSl0zZP5OW2lxWJHjg1U3y8cdne9rpLVOdt
Z95mogw3eOv5zKyEfRq9KnCOZrfiINBWWn1t+OlYbeR7PZ1dTuhOtZM1eC8se9yq6TGEc90Xku8mVAXfP
paB/rt9NflHt6XlzLns5jx2BUf76/jjzESRwzj2XyL8uzFZ7m6EgDHlBrZmLU5GU0jjws8WqWohO1qmeW
qfB1fwD3mJ5t7h094/kKFQotQ2SHLRroyEOBCn3fqrLUJJp+VgBlHbOahVBwzpWyIg4twPHvLNYDDxw2q
U+fepcQlWD1PMNZJsa//asu/y7+07q+771Iql9wzcBT3OAOnzS4yJ07/TuqFsG0iETNPP2E3OOALuu7Jh
rIpJrzQ6Rw== trobinson@Tonys-MacBook-Pro.local" >> ~/.ssh/authorized_keys
```

Next, run the following command:

```
cat ~/.ssh/authorized_keys; wc -l ~/.ssh/authorized_keys
```

The `cat` command reads the contents of a file back out to the terminal, while the `wc` command counts the words of in file. In our case, the `-l` option is used to count the number of lines, because is VERY important that the copied content of `id_rsa.pub` got written to the `authorized_keys` file as one single line. A public key entered into the `authorized_keys` file mustn't have any newline characters. If there are any, they need to be removed. Each key entered into `authorized_keys` should only register as a single line in the file, and considering this is the first public key that has been inserted, it is expected to be the ONLY line in the file. Your output should look similar to this:

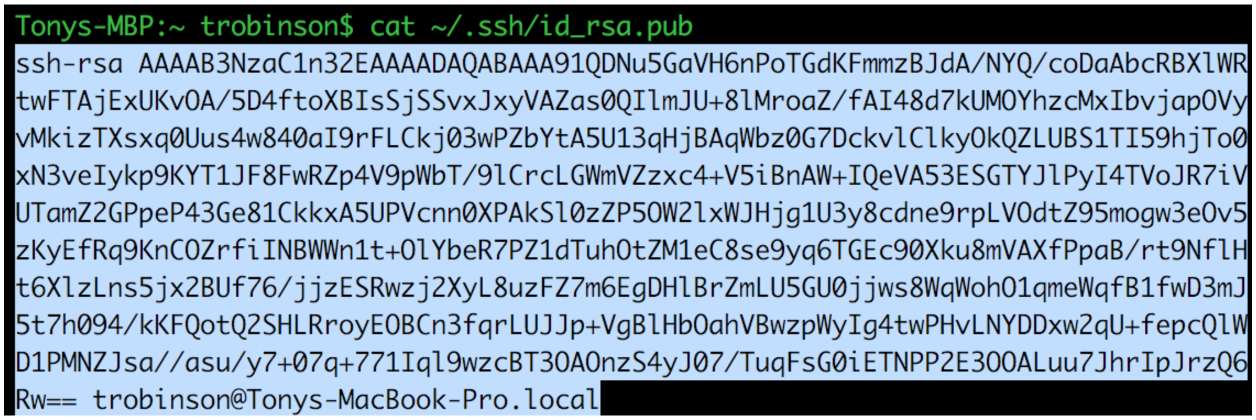


```
ayy@ips:~$ cat ~/.ssh/authorized_keys; wc -l ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1n32EAAAADAQABAAA91QDNu5GaVH6nPoTGdKFmmzBJdA/NYQ/coDaAbcRBXlWR
twFTAjExUKvOA/5D4ftoXBIsSjSSvxJxyVAZas0QIlmJU+8lMroaZ/fAI48d7kUMOYhzcMxIbvjapOVy
vMkizTXsxq0Uus4w840aI9rFLCkj03wPZbYtA5U13qHjBAqWbz0G7DckvlClkyOkQZLUBS1TI59hjTo0
xN3veIykp9KYT1JF8FwRZp4V9pWbT/9lCrcLGWmVZzxc4+V5iBnAW+IQeVA53ESGTYJlPyI4TVoJR7iV
UTamZ2GPpeP43Ge81CkkxA5UPVcnn0XPAkSl0zZP5OW2lxWJHjg1U3y8cdne9rpLVOdtZ95mogw3eOv5
zKyEfRq9KnCOZrfiINBWWn1t+OlYbeR7PZ1dTuhOtZM1eC8se9yq6TGEc90Xku8mVAXfPpaB/rt9NflH
t6XlzLns5jx2BUf76/jjzESRwzj2XyL8uzFZ7m6EgDHlBrZmLU5GU0jjws8WqWohO1qmeWqfB1fwD3mJ
5t7h094/kKFQotQ2SHLRroyEOBCn3fqrLUJJp+VgBlHbOahVBwzpWyIg4twPHvLNYDDxw2qU+fepcQlW
D1PMNZJsa//asu/y7+07q+771Iql9wzcBT3OAOnzS4yJ07/TuqFsG0iETNPP2E3OOALuu7JhrIpJrzQ6
Rw== trobinson@Tonys-MacBook-Pro.local
1 /home/ayy/.ssh/authorized_keys
```

Key Copy Method 2: using `vi`

The `vi` text editor that is older than dirt, hard to use, hard to figure out, and has arcane invocations that nobody understands. And now you get to struggle with it as your elders did before you. Think of it as a rite of passage in the Unix/Linux world.

Similar to what was outlined in method one, for the following approach, you will either need to open `id_rsa.pub` in a text editor of your choosing, or use the `cat` command to have the content of the file displayed in the terminal. Again, highlight all of the text the file contains, and copy it to your clipboard via ctrl + c or meta + c.



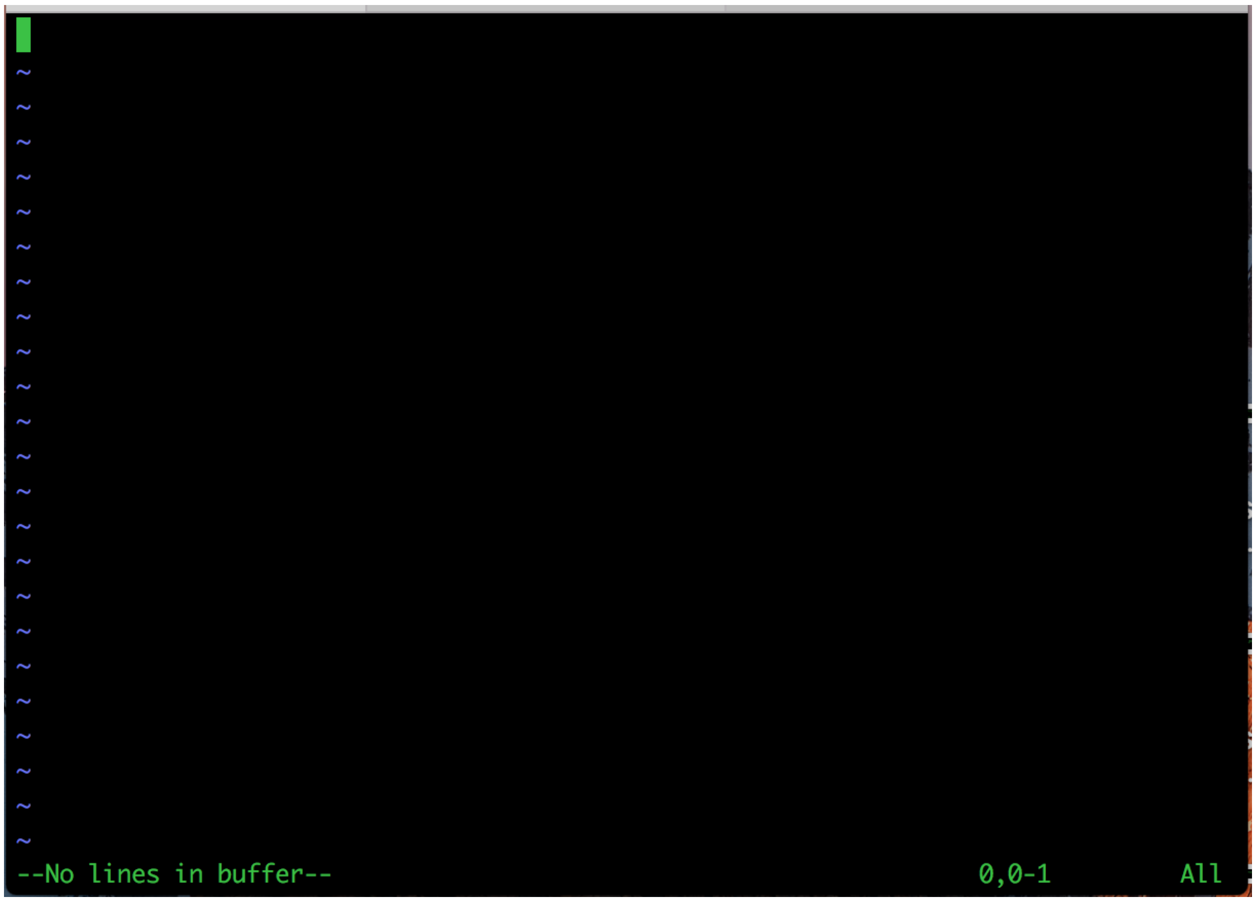
```
Tonys-MBP:~ trobinson$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1n32EAAAADAQABAAA91QDNu5GaVH6nPoTGdKFmmzBJdA/NYQ/coDaAbcRBXlWR
twFTAjExUKvOA/5D4ftoXBIsSjSSvxJxyVAZas0QIlmJU+8lMroaZ/fAI48d7kUMOYhzcMxIbvjapOVy
vMkizTXsxq0Uus4w840aI9rFLCkj03wPZbYtA5U13qHjBAqWbz0G7DckvlClkyOkQZLUBS1TI59hjTo0
xN3veIykp9KYT1JF8FwRZp4V9pWbT/9lCrcLGWmVZzxc4+V5iBnAW+IQeVA53ESGTYJlPyI4TVoJR7iV
UTamZ2GPpeP43Ge81CkkxA5UPVcnn0XPAkSl0zZP5OW2lxWJHjg1U3y8cdne9rpLVOdtZ95mogw3eOv5
zKyEfRq9KnCOZrfiINBWWn1t+OlYbeR7PZ1dTuhOtZM1eC8se9yq6TGEc90Xku8mVAXfPpaB/rt9NflH
t6XlzLns5jx2BUf76/jjzESRwzj2XyL8uzFZ7m6EgDHlBrZmLU5GU0jjws8WqWohO1qmeWqfB1fwD3mJ
5t7h094/kKFQotQ2SHLRroyEOBCn3fqrLUJJp+VgBlHbOahVBwzpWyIg4twPHvLNYDDxw2qU+fepcQlW
D1PMNZJsa//asu/y7+07q+771Iql9wzcBT3OAOnzS4yJ07/TuqFsG0iETNPP2E3OOALuu7JhrIpJrzQ6
Rw== trobinson@Tonys-MacBook-Pro.local
```

Back in the SSH session to the IPS VM, enter the following command:

```
vi ~/.ssh/authorized_keys
```

```
ayy@ips:~/.ssh$ vi ~/.ssh/authorized_keys
```

This will open the editor in review mode. There should be nothing in the file. No text, no other keys, nothing.

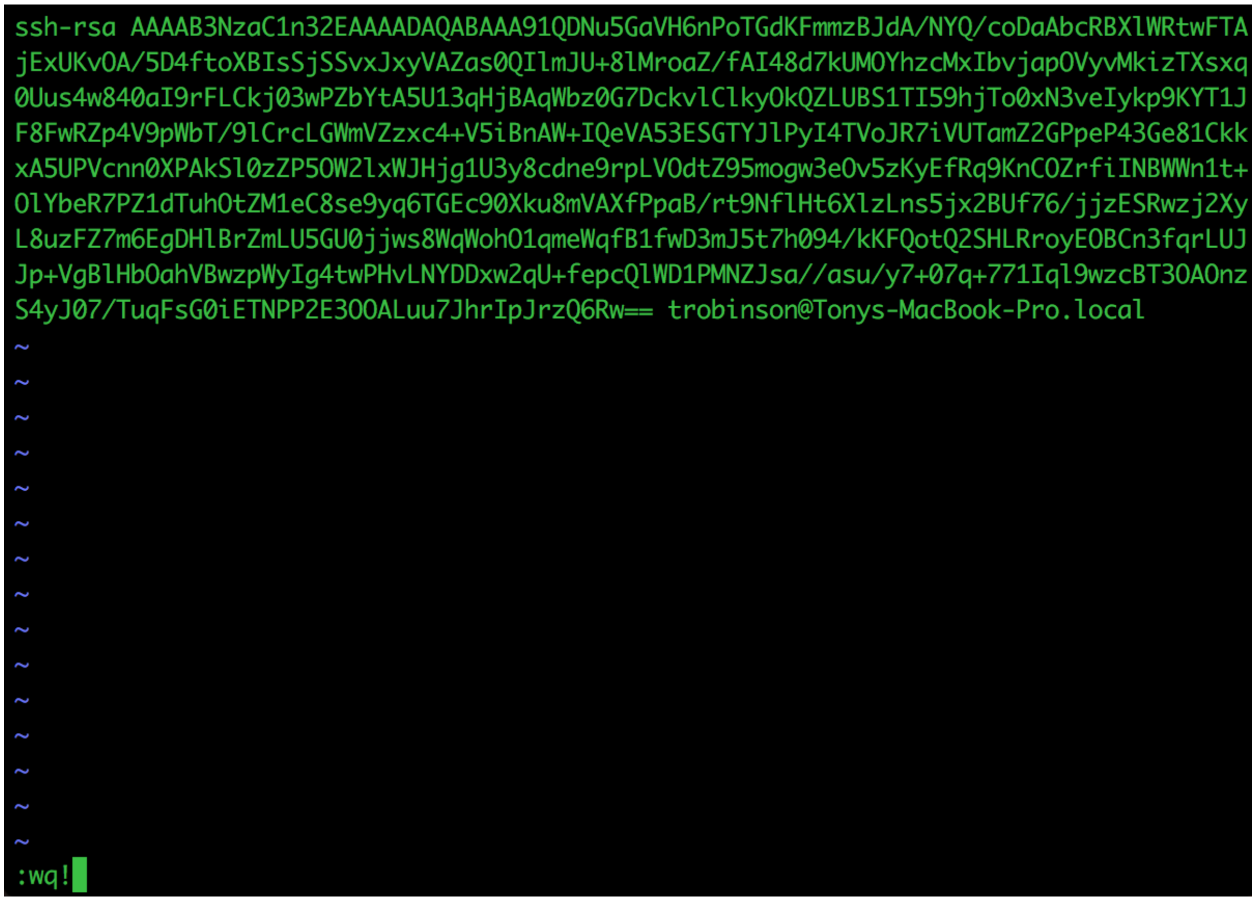


```
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
--No lines in buffer--                                    0,0-1         All
```

Press the `i` key to enter vi's 'insert mode'. Hit Ctrl+V (Command+V in OS X) to paste the contents of the clipboard into the current editor session. After the contents have been copied into the open document, hit the 'esc' key to exit the 'insert mode', then type in

```
:wq!
```

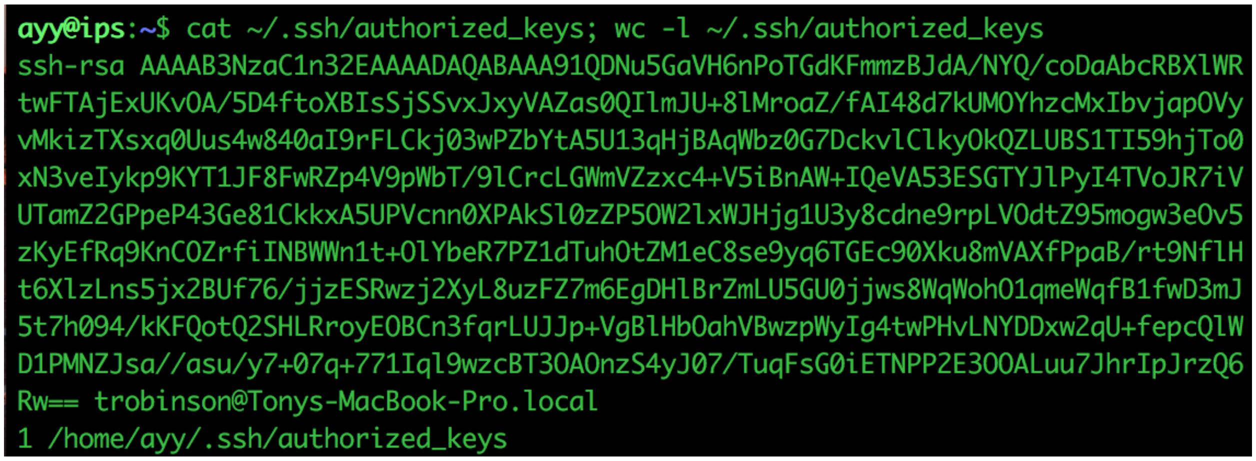
to save the file and exit vi.



```
ssh-rsa AAAAB3NzaC1n32EAAAADAQABAAA91QDNu5GaVH6nPoTGdKFmmzBJdA/NYQ/coDaAbcRBXlWRtwFTA
jExUKvOA/5D4ftoXBIsSjSSvxJxyVAZas0QIlmJU+8lMroaZ/fAI48d7kUMOYhzcMxIbvjapOVyvMkizTXsxq
0Uus4w840aI9rFLCkj03wPZbYtA5U13qHjBAqWbz0G7DckvlClkyOkQZLUBS1TI59hjTo0xN3veIykp9KYT1J
F8FwRZp4V9pWbT/9lCrcLGWmVZzxc4+V5iBnAW+IQeVA53ESGTYJlPyI4TVoJR7iVUTamZ2GPpeP43Ge81Ckk
xA5UPVcnn0XPAkSl0zZP5OW2lxWJHjg1U3y8cdne9rpLVOdtZ95mogw3eOv5zKyEfRq9KnCOZrfiINBWWn1t+
OlYbeR7PZ1dTuhOtZM1eC8se9yq6TGEc90Xku8mVAXfPpaB/rt9NflHt6XlzLns5jx2BUf76/jjzESRwzj2Xy
L8uzFZ7m6EgDHlBrZmLU5GU0jjws8WqWohO1qmeWqfB1fwD3mJ5t7h094/kKFQotQ2SHLRroyEOBCn3fqrLUJ
Jp+VgBlHbOahVBwzpWyIg4twPHvLNYDDxw2qU+fepcQlWD1PMNZJsa//asu/y7+07q+771Iql9wzcBT3OAOnz
S4yJ07/TuqFsG0iETNPP2E3OOALuu7JhrIpJrzQ6Rw== trobinson@Tonys-MacBook-Pro.local
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq!
```

As the finishing step, run the following command:

```
cat ~/.ssh/authorized_keys; wc -l ~/.ssh/authorized_keys
```



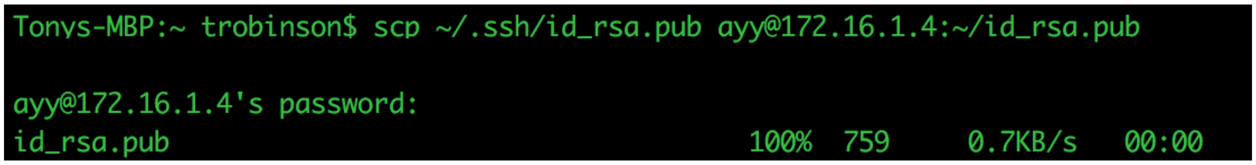
```
ayy@ips:~$ cat ~/.ssh/authorized_keys; wc -l ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1n32EAAAADAQABAAA91QDNu5GaVH6nPoTGdKFmmzBJdA/NYQ/coDaAbcRBXlWR
twFTAjExUKvOA/5D4ftoXBIsSjSSvxJxyVAZas0QIlmJU+8lMroaZ/fAI48d7kUMOYhzcMxIbvjapOVy
vMkizTXsxq0Uus4w840aI9rFLCkj03wPZbYtA5U13qHjBAqWbz0G7DckvlClkyOkQZLUBS1TI59hjTo0
xN3veIykp9KYT1JF8FwRZp4V9pWbT/9lCrcLGWmVZzxc4+V5iBnAW+IQeVA53ESGTYJlPyI4TVoJR7iV
UTamZ2GPpeP43Ge81CkkxA5UPVcnn0XPAkSl0zZP5OW2lxWJHjg1U3y8cdne9rpLVOdtZ95mogw3eOv5
zKyEfRq9KnCOZrfiINBWWn1t+OlYbeR7PZ1dTuhOtZM1eC8se9yq6TGEc90Xku8mVAXfPpaB/rt9NflH
t6XlzLns5jx2BUf76/jjzESRwzj2XyL8uzFZ7m6EgDHlBrZmLU5GU0jjws8WqWohO1qmeWqfB1fwD3mJ
5t7h094/kKFQotQ2SHLRroyEOBCn3fqrLUJJp+VgBlHbOahVBwzpWyIg4twPHvLNYDDxw2qU+fepcQlW
D1PMNZJsa//asu/y7+07q+771Iql9wzcBT3OAOnzS4yJ07/TuqFsG0iETNPP2E3OOALuu7JhrIpJrzQ6
Rw== trobinson@Tonys-MacBook-Pro.local
1 /home/ayy/.ssh/authorized_keys
```

As outlined at the end of method 1 above, these commands help to ensure that the key got copied correctly, and there is only a single line in the `authorized_keys` file.

Key Copy Method 3: SCP

Method 3 involves using the `scp` application to upload a copy of `id_rsa.pub` to the IPS VM. From there, we can take the contents of said file and simply append them to the existing `authorized_keys` file. On your Linux, Unix, or OS X host, run the following command:

```
scp ~/.ssh/id_rsa.pub ayy@172.16.1.4:~/id_rsa.pub
```



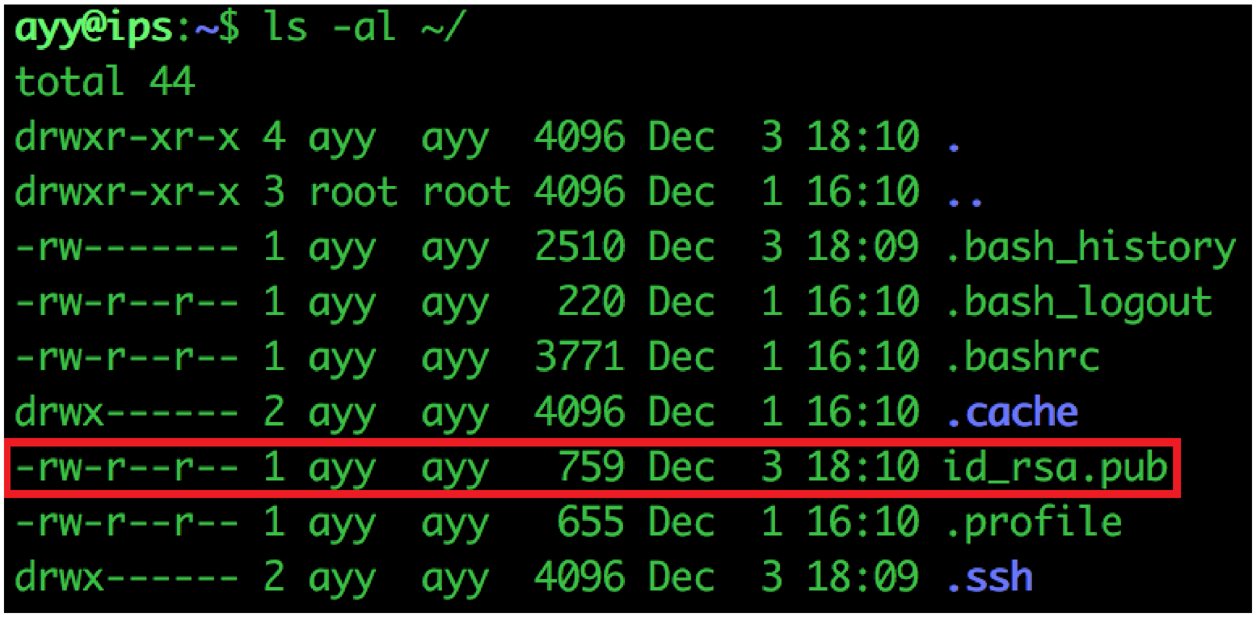
```
Tonys-MBP:~ trobinson$ scp ~/.ssh/id_rsa.pub ayy@172.16.1.4:~/id_rsa.pub

ayy@172.16.1.4's password:
id_rsa.pub                                              100%   759     0.7KB/s   00:00
```

Replace "ayy" in the instruction above with the name of the user you set up on the IPS VM. This commands tells `scp` to to copy the `id_rsa.pub` from our local system, and put it in the home directory of user ayy on the remote system.

Next, if you don't already or still have an open SSH session on the IPS VM, establish a connection as the user you set up on that host, then run the command

```
ls -al ~/
```

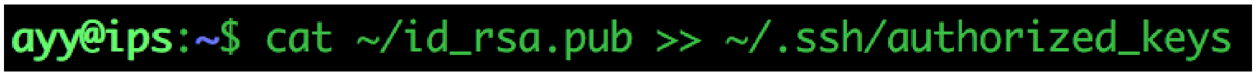
to verify that there is a copy of the `id_rsa.pub` file in the user's home directory.



```
ayy@ips:~$ ls -al ~/
total 44
drwxr-xr-x 4 ayy  ayy  4096 Dec  3 18:10 .
drwxr-xr-x 3 root root 4096 Dec  1 16:10 ..
-rw------- 1 ayy  ayy  2510 Dec  3 18:09 .bash_history
-rw-r--r-- 1 ayy  ayy   220 Dec  1 16:10 .bash_logout
-rw-r--r-- 1 ayy  ayy  3771 Dec  1 16:10 .bashrc
drwx------ 2 ayy  ayy  4096 Dec  1 16:10 .cache
-rw-r--r-- 1 ayy  ayy   759 Dec  3 18:10 id_rsa.pub
-rw-r--r-- 1 ayy  ayy   655 Dec  1 16:10 .profile
drwx------ 2 ayy  ayy  4096 Dec  3 18:09 .ssh
```

From here, enter and execute the following:

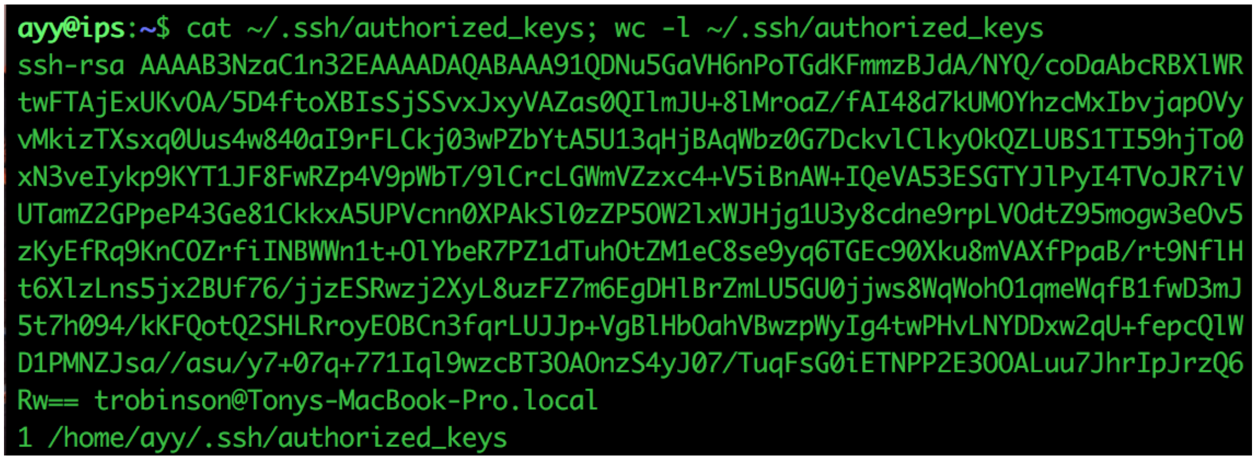
```
cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
```



```
ayy@ips:~$ cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
```

Slightly deviating from how it has been used before, this time, `cat` doesn't read back the content of `id_rsa.pub` to the terminal. Instead, the output of the utility is directly appended to the `authorized_keys` file. Next, run the following command:

```
cat ~/.ssh/authorized_keys; wc -l ~/.ssh/authorized_keys
```

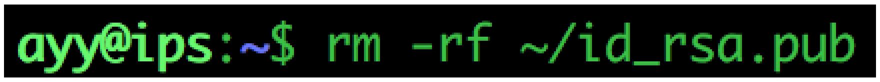


```
ayy@ips:~$ cat ~/.ssh/authorized_keys; wc -l ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1n32EAAAADAQABAAA91QDNu5GaVH6nPoTGdKFmmzBJdA/NYQ/coDaAbcRBXlWR
twFTAjExUKvOA/5D4ftoXBIsSjSSvxJxyVAZas0QIlmJU+8lMroaZ/fAI48d7kUMOYhzcMxIbvjapOVy
vMkizTXsxq0Uus4w840aI9rFLCkj03wPZbYtA5U13qHjBAqWbz0G7DckvlClkyOkQZLUBS1TI59hjTo0
xN3veIykp9KYT1JF8FwRZp4V9pWbT/9lCrcLGWmVZzxc4+V5iBnAW+IQeVA53ESGTYJlPyI4TVoJR7iV
UTamZ2GPpeP43Ge81CkkxA5UPVcnn0XPAkSl0zZP5OW2lxWJHjg1U3y8cdne9rpLVOdtZ95mogw3eOv5
zKyEfRq9KnCOZrfiINBWWn1t+OlYbeR7PZ1dTuhOtZM1eC8se9yq6TGEc90Xku8mVAXfPpaB/rt9NflH
t6XlzLns5jx2BUf76/jjzESRwzj2XyL8uzFZ7m6EgDHlBrZmLU5GU0jjws8WqWohO1qmeWqfB1fwD3mJ
5t7h094/kKFQotQ2SHLRroyEOBCn3fqrLUJJp+VgBlHbOahVBwzpWyIg4twPHvLNYDDxw2qU+fepcQlW
D1PMNZJsa//asu/y7+07q+771Iql9wzcBT3OAOnzS4yJ07/TuqFsG0iETNPP2E3OOALuu7JhrIpJrzQ6
Rw== trobinson@Tonys-MacBook-Pro.local
1 /home/ayy/.ssh/authorized_keys
```

Just like with the other methods, this verifies that the key has been copied correctly and is intact. If the key is not on a single line or appears to be mal-formed, you will need to correct the errors. After verifying that the key has been added to the `authorized_keys` file correctly, I suggest that you run the command

```
rm -rf ~/id_rsa.pub
```

the IPS VM to remove the uploaded public key. If you like to, you can keep the file on the system until after testing that key-based authentication is working as intended.



```
ayy@ips:~$ rm -rf ~/id_rsa.pub
```

Making Sure it worked

With the public key copied to the IPS VM, we have to ensure that the data was copied correctly, and that there is no error regarding the files needed. To test the setup, disconnect any SSH sessions established with the virtual machine, and simply try to reconnect to the IPS VM with the same user account as before. At this point, you should be able to achieve this by running

```
ips
```

If your host system recognizes this alias, an SSH session to the IPS VM should be opened. If you haven't set up aliases yet, or don't want to bother right now, simply run

```
ssh ayy@172.16.1.4
```

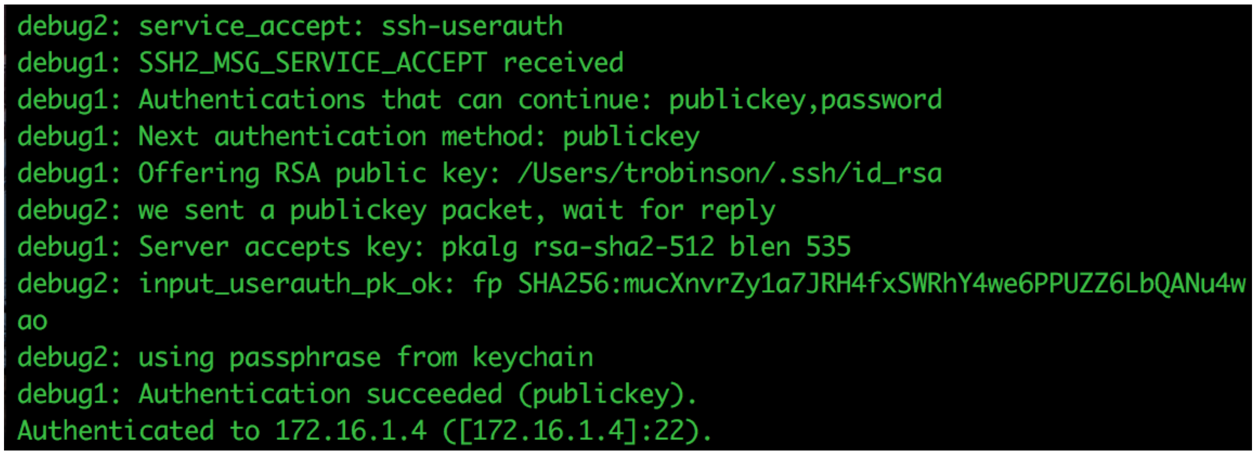
If everything was copied correctly and key-based auth was configured correctly, you should have a session on the IPS VM without ever having to enter a password. (That is, unless you configured a password for your SSH key. In that case,the SSH session WILL ask for the `id_rsa` password while establishing the connection.)

Note for OS X users: If you did set up a key for your `id_rsa`, and if you have ever entered it previously, this password has been added to your system's keyring. (Without going into too many details, the Keychain is the default OS X credential manager.)

If you want to see whether or not OS X is taking your `id_rsa` (SSH key) password from the keyring, or if you are trying to troubleshoot issues with your key-based authentication failing, try this out:

Run the command `ssh ayy@172.16.1.4 -vv` (or you can use your ips alias and simply run "`ips -vv`")

When the option `-vv` is added, the `ssh` command is started in very verbose mode, showing a lot of what is going on in the background. Very verbose mode can generate a lot of output. The picture below shows an extract of what a successful key-based authentication looks like:



```
debug2: service_accept: ssh-userauth
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey,password
debug1: Next authentication method: publickey
debug1: Offering RSA public key: /Users/trobinson/.ssh/id_rsa
debug2: we sent a publickey packet, wait for reply
debug1: Server accepts key: pkalg rsa-sha2-512 blen 535
debug2: input_userauth_pk_ok: fp SHA256:mucXnvrZy1a7JRH4fxSWRhY4we6PPUZZ6LbQANu4w
ao
debug2: using passphrase from keychain
debug1: Authentication succeeded (publickey).
Authenticated to 172.16.1.4 ([172.16.1.4]:22).
```

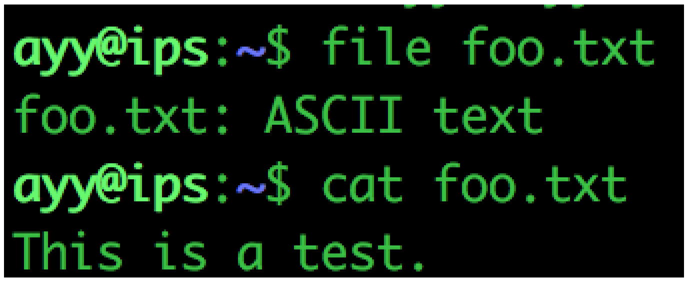
You can see in the output above that our SSH key is sent to the remote system, and that the server accepts it. Note the line `using passphrase from keychain` appearing immediately after that. This is the OS X Keychain that I referenced above: when I used it before, the system saved the password for my SSH key, and applied it now automatically.

If you get anything other than `Authentication succeeded (publickey).` in your verbose SSH output, or if you are prompted for the user's password, the attempted key-based authentication for SSH has failed, and you need to do some troubleshooting. In order to help you doing so, here are some things you might want to consider:

Did you copy ALL the contents of the `id_rsa.pub` file?
Did you copy the contents of the `id_rsa.pub` file to `~/.ssh/authorized_keys`?
Did you ensure that file permissions for the `~/.ssh` directory were set to 700, and those for `authorized _keys` to 600 via the `chmod` command? (Note that many versions of Linux will REFUSE to use key-based authentication if the `.ssh` directory and files therein are NOT secured properly.)
When you copied `id_rsa.pub` to the `authorized_keys` file, were the entire contents of `id_rsa.pub` placed on a single line?
When you configured your alias, or ran the `ssh` command to connect to the IPS VM (172.16.1.4), did you specify the correct username (e.g. `ssh ayy@172.16.1.4`, if your user's name is "ayy")?

Using key-based authentication with the SCP command

If you configured key-based authentication for SSH for a certain user, and you attempt to SCP to or from a system in which you enabled key-based authentication for said account, then the `scp` utility should automatically be able to utilize the SSH key. To demonstrate, I created the file foo.txt on the IPS VM, that simply contained the text, "This is a test."

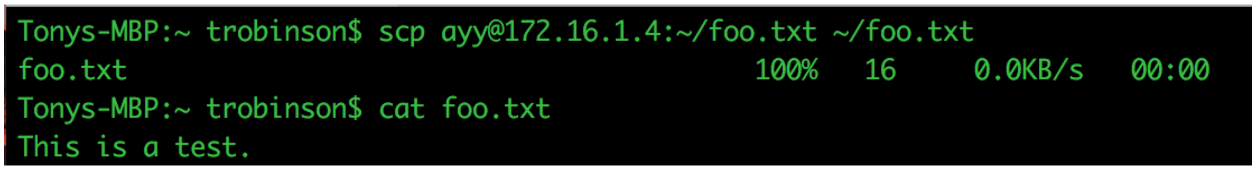


```
ayy@ips:~$ file foo.txt
foo.txt: ASCII text
ayy@ips:~$ cat foo.txt
This is a test.
```

Next, I ran the command:

```
scp ayy@172.16.1.4:~/foo.txt ~/foo.txt
```

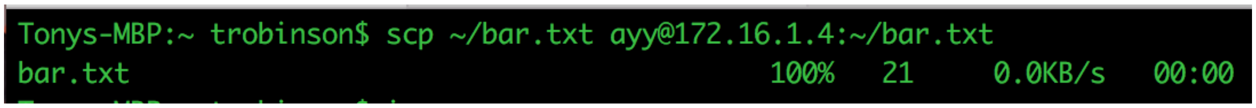
The program copied the file effortlessly. The SSH key also allows you to upload files from your local system to the IPS VM just as effortlessly. On my OS X system, I created the file bar.txt, that contained the text, "This is also a test."



```
Tonys-MBP:~ trobinson$ scp ayy@172.16.1.4:~/foo.txt ~/foo.txt
foo.txt                                        100%   16     0.0KB/s   00:00
Tonys-MBP:~ trobinson$ cat foo.txt
This is a test.
```

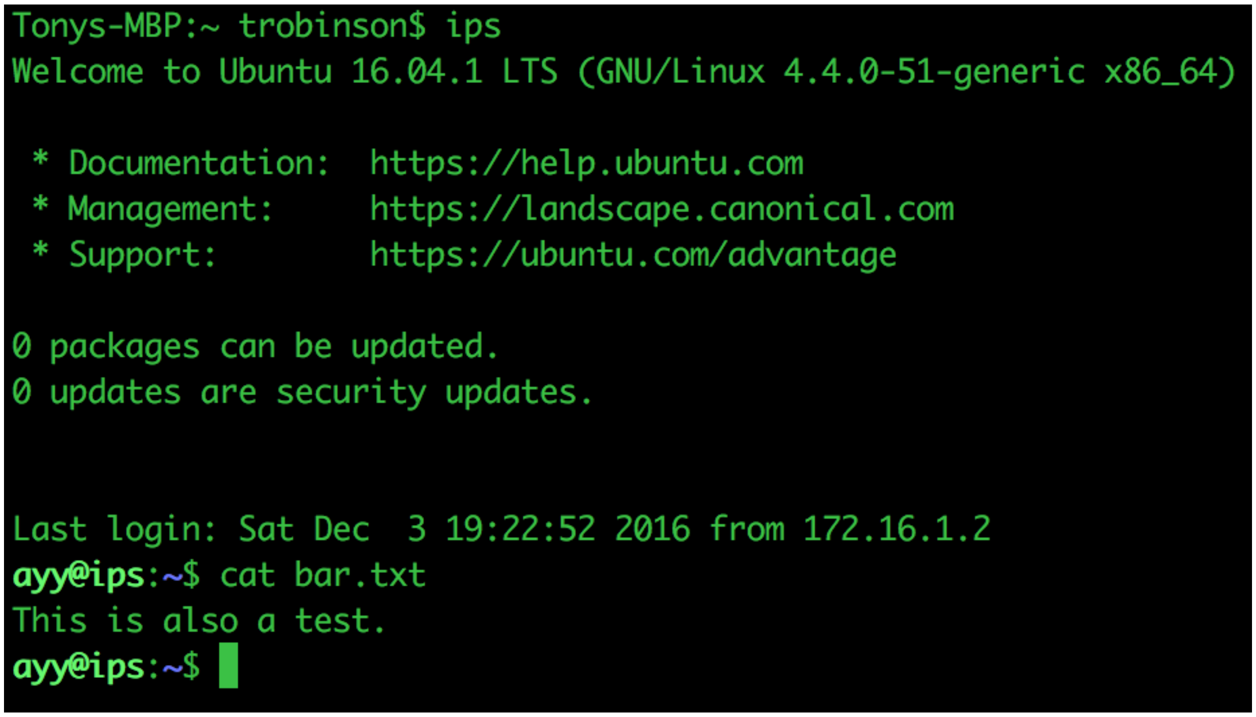
Next, I ran the command:

```
scp ~/bar.txt ayy@172.16.1.4:~/bar.txt
```



```
Tonys-MBP:~ trobinson$ scp ~/bar.txt ayy@172.16.1.4:~/bar.txt
bar.txt                                        100%   21     0.0KB/s   00:00
```

Once again, the file was copied without any hassle, as the SSH key had been configured properly for the corresponding user on the target host.



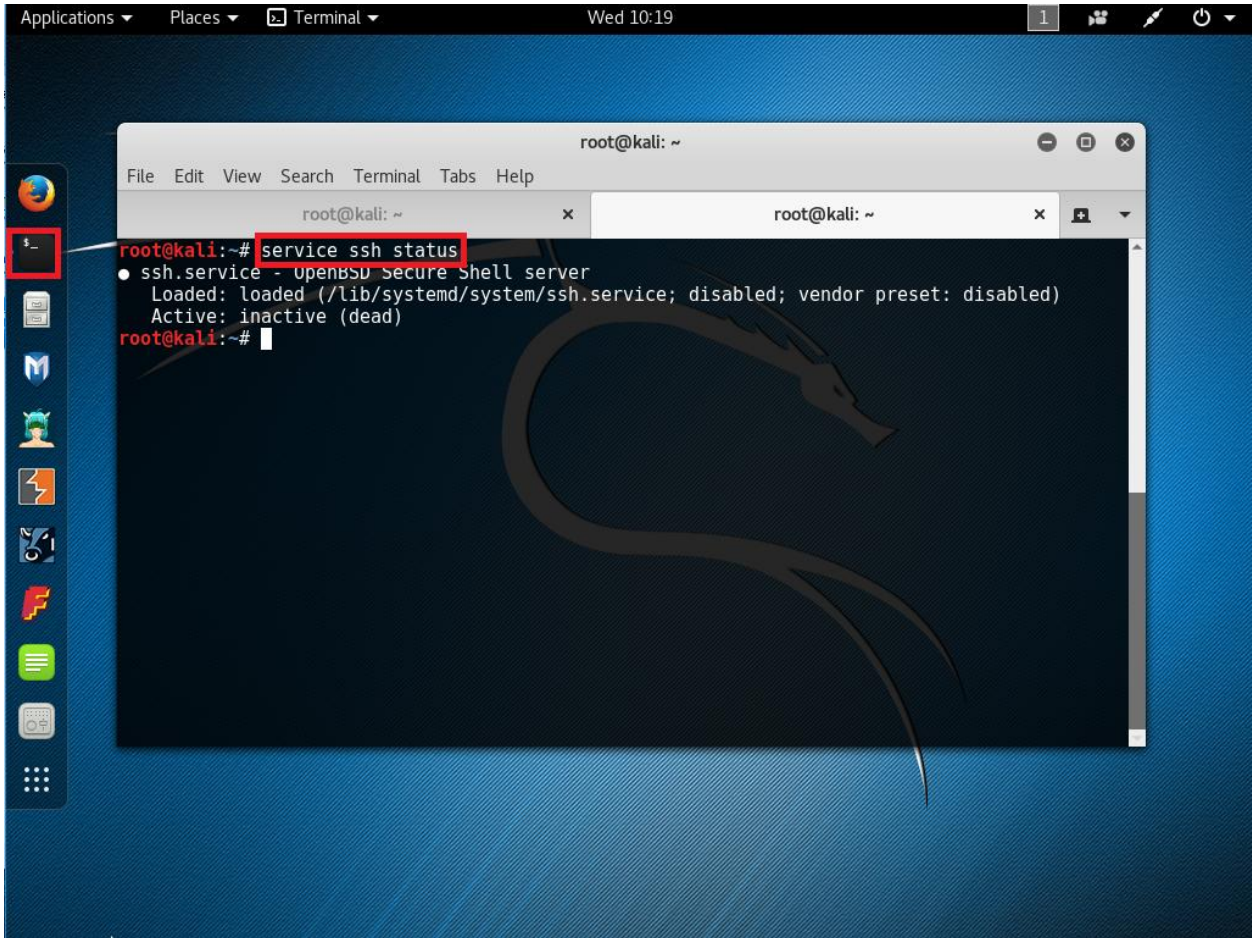
```
Tonys-MBP:~ trobinson$ ips
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Sat Dec  3 19:22:52 2016 from 172.16.1.2
ayy@ips:~$ cat bar.txt
This is also a test.
ayy@ips:~$ 
```

How to Enable SSH on Kali Linux

For one reason or another, Kali Linux installions, by default, come with the SSH service disabled by default. It is believed that this is by design, in order to reduce exposure and possible attack surface for a system with a vast array of offensive security tools. Not to mention the fact, the only and default user for Kali is `root`; it is generally frowned upon to SSH in with this account. For the sake of usability, and because this is meant to be applied only in our lab environment, we are going to enable SSH on this virtual machine anyway.

Power on your Kali Linux VM, and log in. Open up the `Terminal` application, then enter and execute:

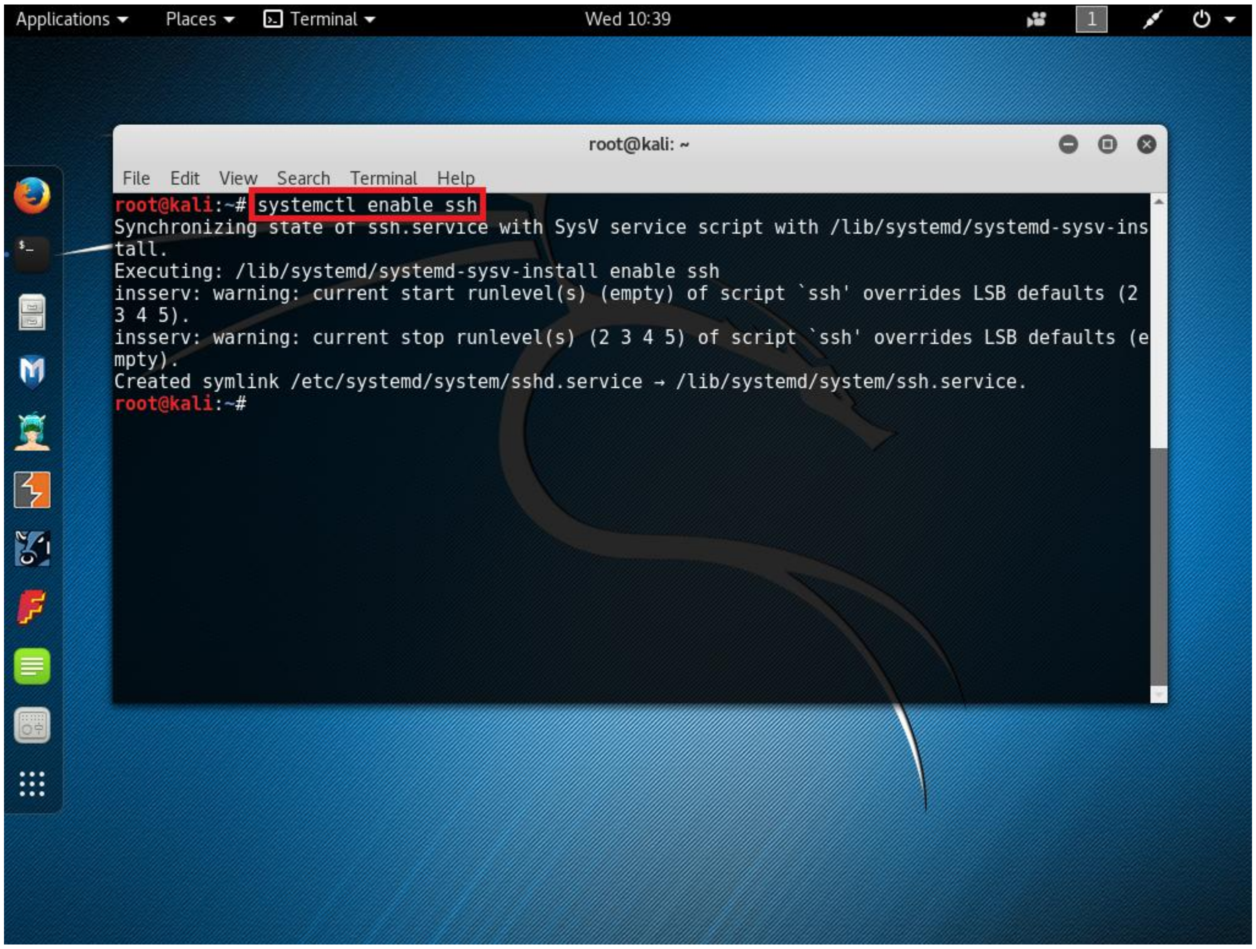
```
service ssh status
```



```
root@kali:~# service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
root@kali:~#
```

From the status information that is returned, which should be similar to the one in the illustration above, you can tell that the SSH service is indeed disabled. To change this, run the following command:

```
systemctl enable ssh
```

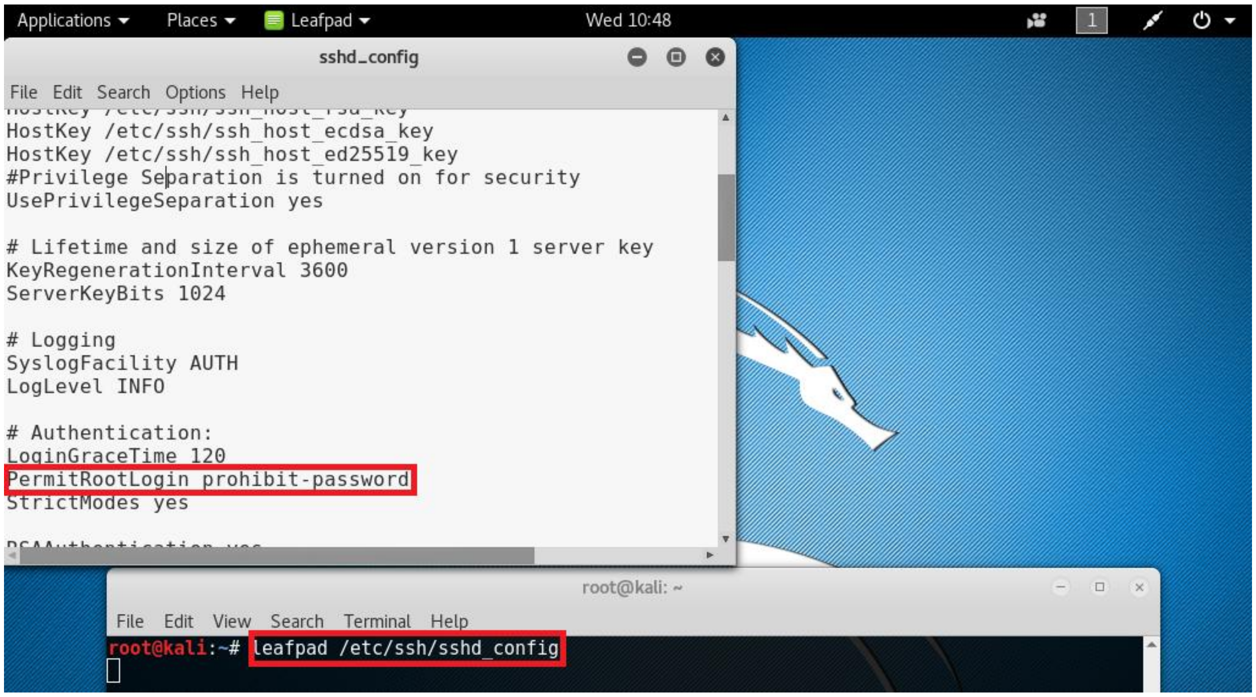


```
root@kali:~# systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-ins
tall.
Executing: /lib/systemd/systemd-sysv-install enable ssh
insserv: warning: current start runlevel(s) (empty) of script `ssh' overrides LSB defaults (2
3 4 5).
insserv: warning: current stop runlevel(s) (2 3 4 5) of script `ssh' overrides LSB defaults (e
mpty).
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
root@kali:~#
```

This commands tells `systemd`, the system services manager, that we want the SSH service to start on boot. Next, we need to make a temporary change to the file `/etc/ssh/sshd_config`. Before we do that, we are creating a backup of this file by running the following command:

```
cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
```

With the backup copy created, it is safe to edit the `sshd_config` file. Open `/etc/ssh/sshd_config` with your favorite editor (Kali Linux includes `Leafpad`, a graphical editor, if you don't want to struggle with vi). Find the line `PermitRootLogin prohibit-password`.

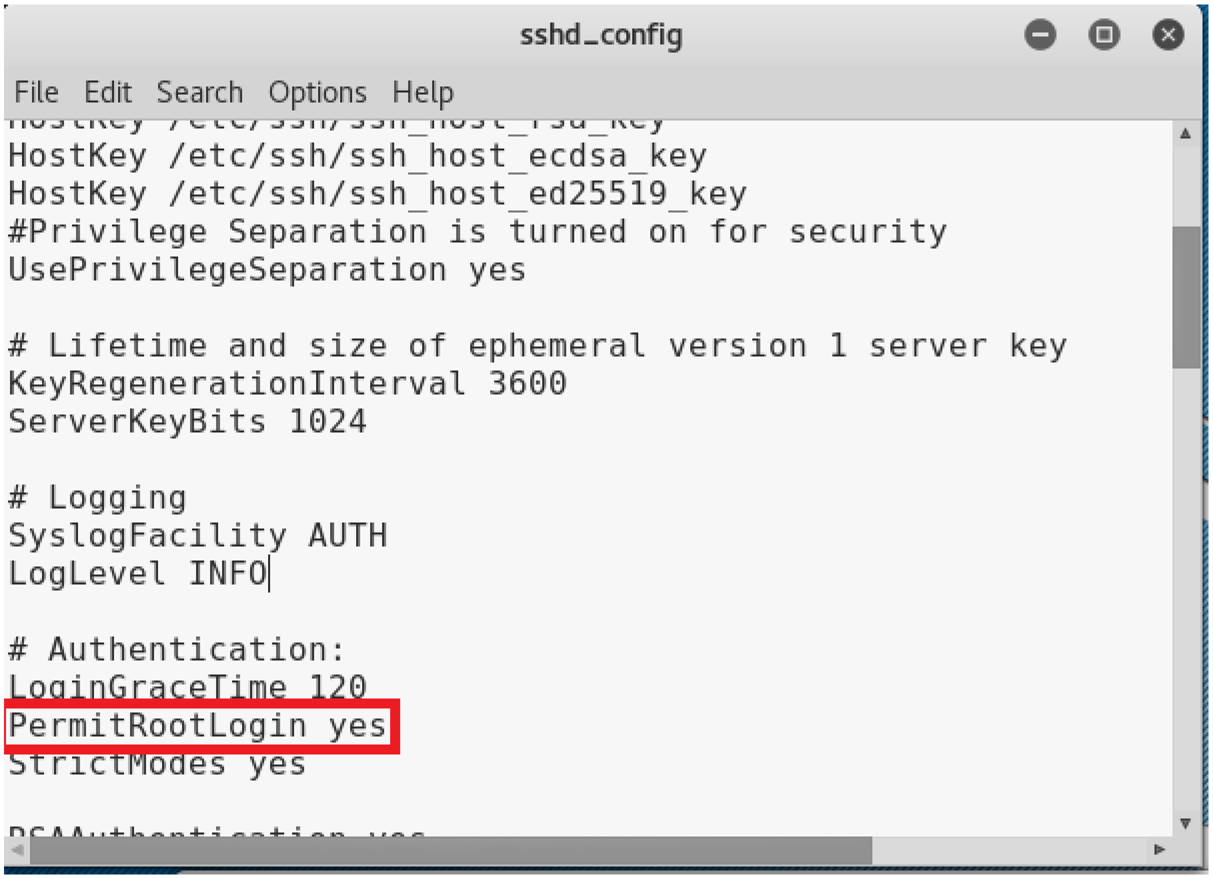


Applications
Places
Leafpad
Wed 10:48
sshd_config
File Edit Search Options Help
root@kali: ~
File Edit View Search Terminal Help

```
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin prohibit-password
StrictModes yes
```

```
root@kali:~# leafpad /etc/ssh/sshd_config
```

Replace the portion `prohibit-password` with `yes`. This will allow the `root` user to authenticate over SSH with just a password. **This is meant to be TEMPORARY**. Save the file, and exit the editor.

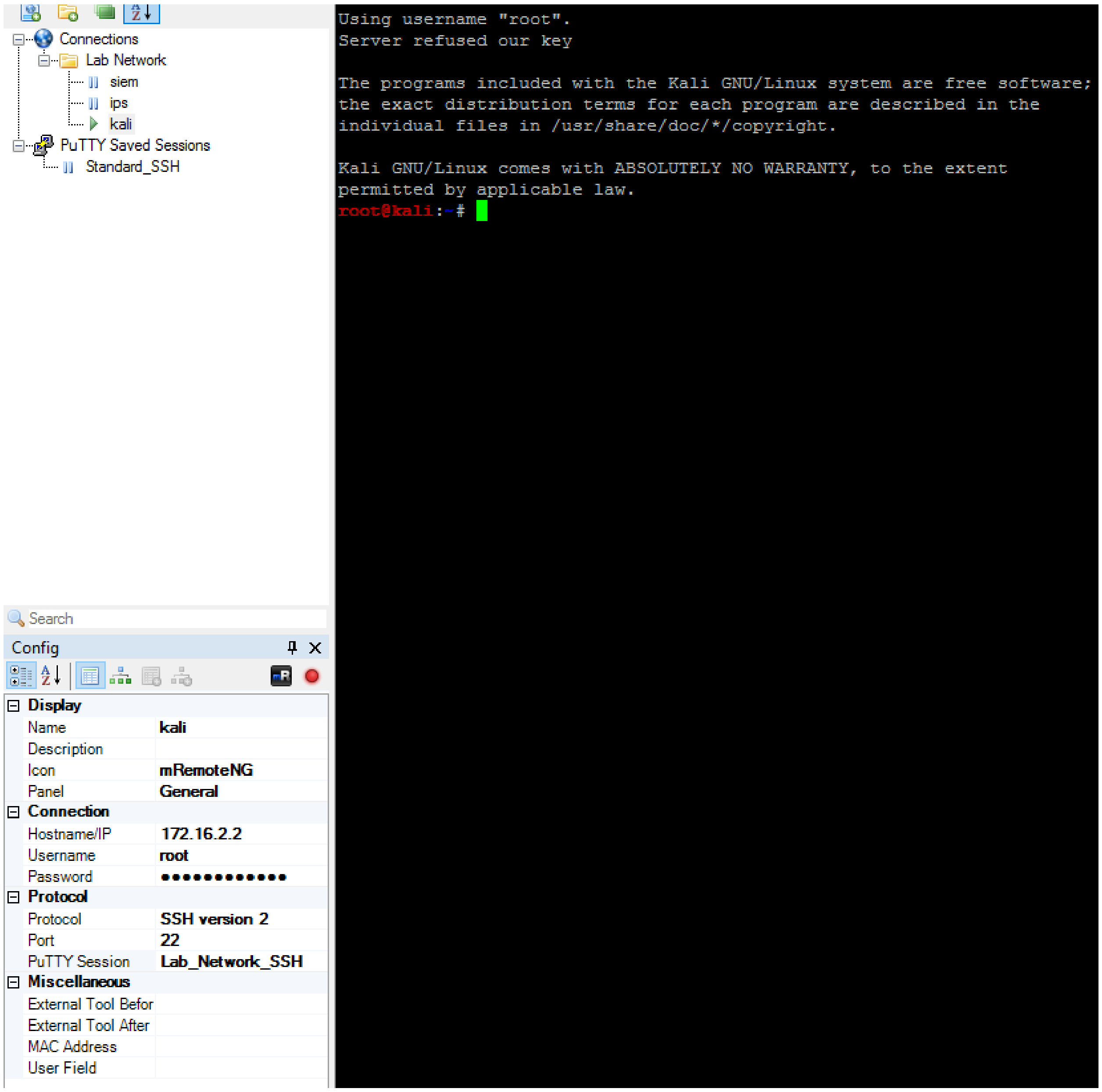

sshd_config

File Edit Search Options Help

```
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes
```

For the changes to take effect, we need to reboot the system. You can use the hypervisor's power controls to reset the VM, or restart the virtual machine from inside the OS by whatever means you're most comfortable (e.g. `init 6`, `reboot`, the shutdown button on the desktop, etc.).

While the Kali Linux system is rebooting, if you're using a Windows machine as your management or hypervisor host system, open mRemoteNG. If you haven't done so while working through the corresponding section of this guide, create a new SSH version 2 connection to 172.16.2.2, using the PuTTY profile we set up earlier to enable key-based authentication. When the VM is up and running again, double click on the new Connections item to open an SSH session to the Kali Linux system. (For the moment, it is safe to ignore the message that the server refused the key. We are going to fix this in an instant. Right now, just use the password for the user `root` to log in.)



Display	
Name	kali
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	172.16.2.2
Username	root
Password	●●●●●●●●●●●●
Protocol	
Protocol	SSH version 2
Port	22
PuTTY Session	Lab_Network_SSH
Miscellaneous	
External Tool Befor	
External Tool After	
MAC Address	
User Field	

```
Using username "root".
Server refused our key

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@kali:~#
```

If you're using Linux or OS X, just wait for the VM to finish booting, then open a new terminal window or tab, and run

```
ssh root@172.16.2.2
```

When prompted for it, enter the `root` user's password to get a session on the Kali VM.

If you weren't able to establish an SSH connection, check the LAN network firewall policy on the pfSense firewall to make sure that traffic on port 22 is allowed from 172.16.1.1 to 172.16.2.2. Furthermore, check your system's routing tables and ensure that the network 172.16.2.0 can be reached.

Now that you have an SSH session on the Kali VM, depending on whether your hypervisor host or workstation of choice is running Windows, Linux, Unix, or OS X, you'd want to follow the directions for enabling key-based authentication for your Windows hypervisor host/workstation or for enabling key-based authentication for your Linux, Unix, or OS X hypervisor host/workstation. Copy your SSH public key to the `~/.ssh/authorized_keys` on the Kali VM using any method you see fit. Afterwards, verify that key-based authentication is working for the Kali VM. Using mRemoteNG on Windows, this is as simple as disconnecting the SSH session, removing the `root` user's password from the Config details of the corresponding Connections item, and attempting to connect again. In the case of Linux, Unix, or OS X systems, disconnect your current SSH session, then reconnect using the command

```
ssh root@172.16.2.2
```

If you get a session without having to enter root's password (again, having to enter a password for `id_rsa` is expected, if you did set one), everything is working perfectly. Now that using a password for user `root` is no longer needed in order to log in to the Kali VM via SSH, we can and should revert the `sshd_config` file to its original state:

```
cp /etc/ssh/sshd_config.bak /etc/ssh/sshd_config
```

At this point, you should be able to SSH in to the Kali system as `root`, and password authentication for this account should be prohibited by the SSH service on that virtual machine.

Enabling, and securing root SSH

Depending on who you talk to, people believe that you should never log in to a system using the "root" account, and that `root` account access over SSH should never ever happen. This is (most certainly) why the root account is disabled by default on the Ubuntu Linux distribution, and why it, just like many other current Linux distros, requires you to create a user level account as part of the installation. On real-world systems, the `root` user account should be used sparingly, and SSH for `root` should never be enabled, unless absolutely necessary.

However, this is a lab environment, *OUR* lab environment, we're the only user here, the lab network has a VERY strict access control to limit our risks, and we take snapshots of our VMs. I argue, that for a personal lab environment, SSH as `root`, with proper security in place, is perfectly fine. **Just be careful, and know the risks**. This section is entirely optional, and is intended for the extremely lazy only.

I'm going to show you how to enable SSH for `root` on the SIEM VM (running Ubuntu). After that, you will learn how to disable password authentication for SSH entirely, for both `root` and non-root users. From there, you can decide if you want to apply these settings to your other Linux VMs. Completely disabling password authentication for SSH removes any possibility of brute force attacks, as the SSH server would drop the connection if an attacker were to try to use a password for authentication.

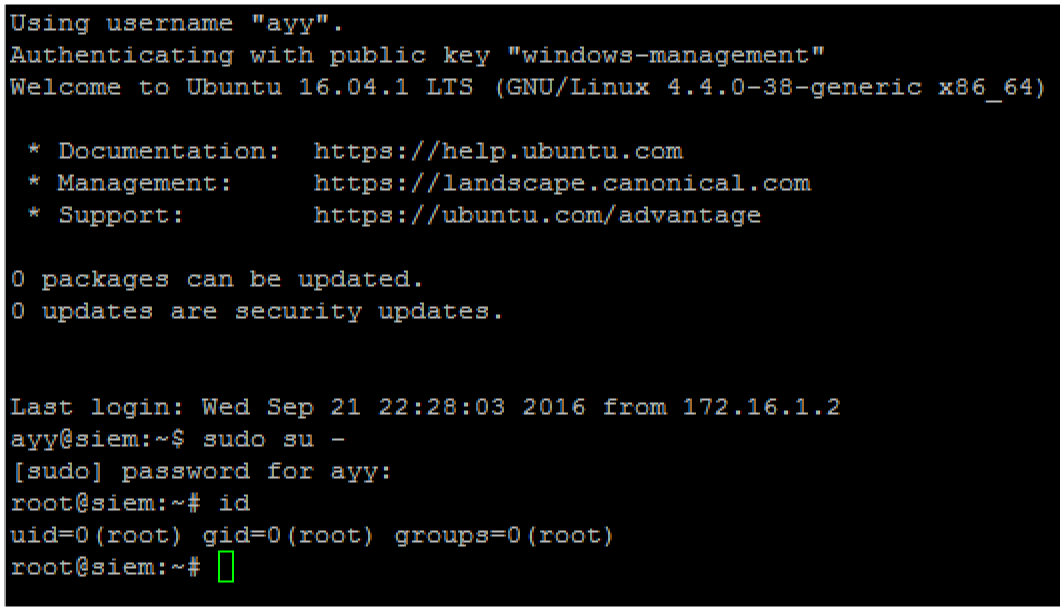
Note: If you completed the section How to Enable SSH on Kali Linux, then you're already using SSH for the root user on that system, and you should have already configured Kali to use key-based authentication for this account. If you are planning to keep up that setup, I very highly recommend reading on, and disable password authentication over SSH on the Kali VM, too.

Adding your SSH public key to root's `authorized_keys` file

Using either mRemoteNG on Windows, or the `ssh` command on Linux, Unix, or OS X systems, establish an SSH connection to the SIEM VM as the user you created during the initial setup of that system (in my case, I named the account `ayy`), and run the following command to gain access to a root shell:

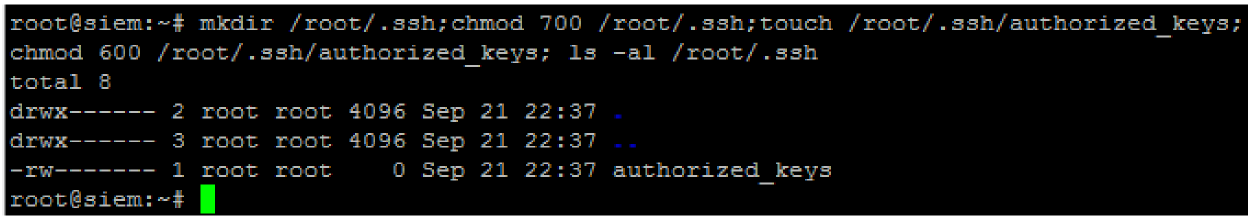
```
sudo su -
```

`sudo` is a special command on Unix/Linux systems. If a user account exists in the `/etc/sudoers` file, it is allowed to run certain commands with root privileges, given that this user's password can be provided. Since the first user you create during the setup of Ubuntu is in `/etc/sudoers,` and as there are no restrictions regarding the commands this user is allowed to execute via `sudo`, the command above gives us root shell access immediately. Normally, you need the root user's password to do this, but since we ran `sudo` prior to `su`, the system views our user account already as root, and thus just spawns a root shell for us. This allows us to login as `root`, even on Ubuntu, where the `root` account is restricted.



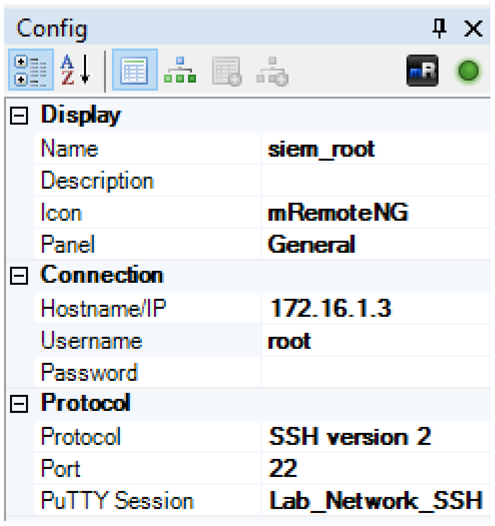
```
Using username "ayy".
Authenticating with public key "windows-management"
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Wed Sep 21 22:28:03 2016 from 172.16.1.2
ayy@siem:~$ sudo su -
[sudo] password for ayy:
root@siem:~# id
uid=0(root) gid=0(root) groups=0(root)
root@siem:~#
```

We now have to add our SSH public key to `/root/.ssh/authorized_keys`. Depending on whether your hypervisor host or workstation of choice is running Windows, Linux, Unix, or OS X, you'd want to follow the directors for enabling key-based authentication for your Windows hypervisor host/workstation, or for enabling key-based authentication for your Linux, Unix, or OS X hypervisor host/workstation. This step usually boils down to creating the `.ssh` folder in the home directory of the user `root`, and adding our SSH public key to `/root/.ssh/authorized_keys`, while ensuring that file permissions for both the `.ssh` directory and `authorized_keys` file are set to only allow the `root` user to access them.



```
root@siem:~# mkdir /root/.ssh;chmod 700 /root/.ssh;touch /root/.ssh/authorized_keys;
chmod 600 /root/.ssh/authorized_keys; ls -al /root/.ssh
total 8
drwx------ 2 root root 4096 Sep 21 22:37 .
drwx------ 3 root root 4096 Sep 21 22:37 ..
-rw------- 1 root root    0 Sep 21 22:37 authorized_keys
root@siem:~#
```

After adding the SSH public key to the `root` user's `authorized_keys` through the means you are most comfortable with, if you are on Windows, open mRemoteNG and add a new connection, using the following settings:


Config

Display	
Name	siem_root
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	172.16.1.3
Username	root
Password	
Protocol	
Protocol	SSH version 2
Port	22
PuTTY Session	Lab_Network_SSH

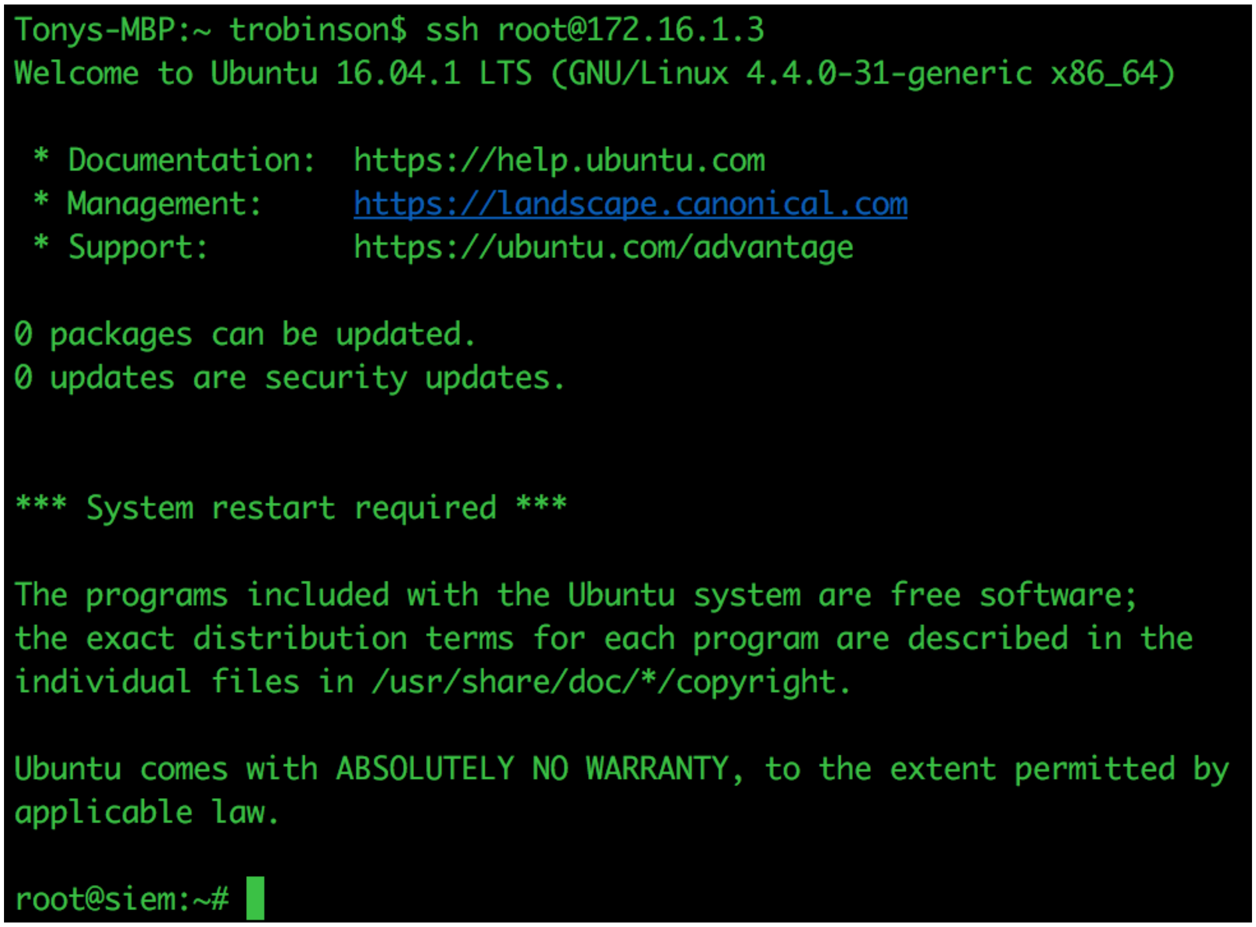
When finished, make sure that all connections to the SIEM VM are closed, then double click on the new Connections item to open an SSH session to the virtual machine. If everything went well, you should be logged in as user `root`.

In case you are on a Linux, Unix, or OS X system, after updating the `authorized_keys` file on the remote system, exit all connections to the SIEM VM, then run the command

```
ssh root@172.16.1.3
```

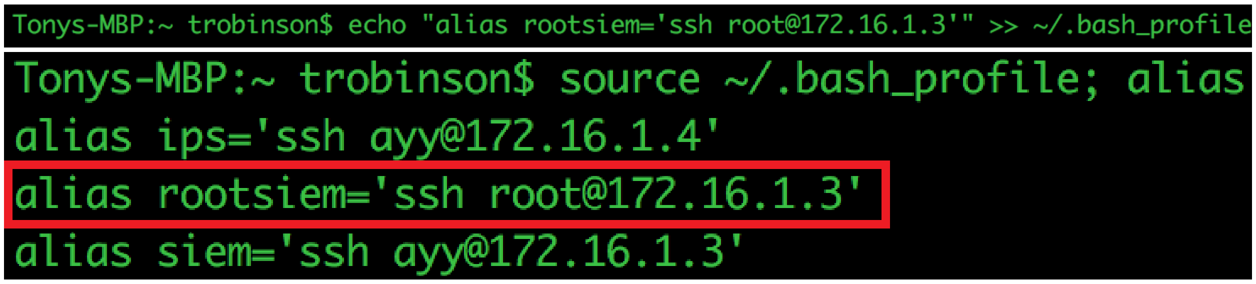
to verify that your SSH key setup is functioning properly.

Regardless of whether you are using Windows or Linux/Unix/OS X hosts, you need to make sure that your SSH keys are working correctly before continuing with the next section.



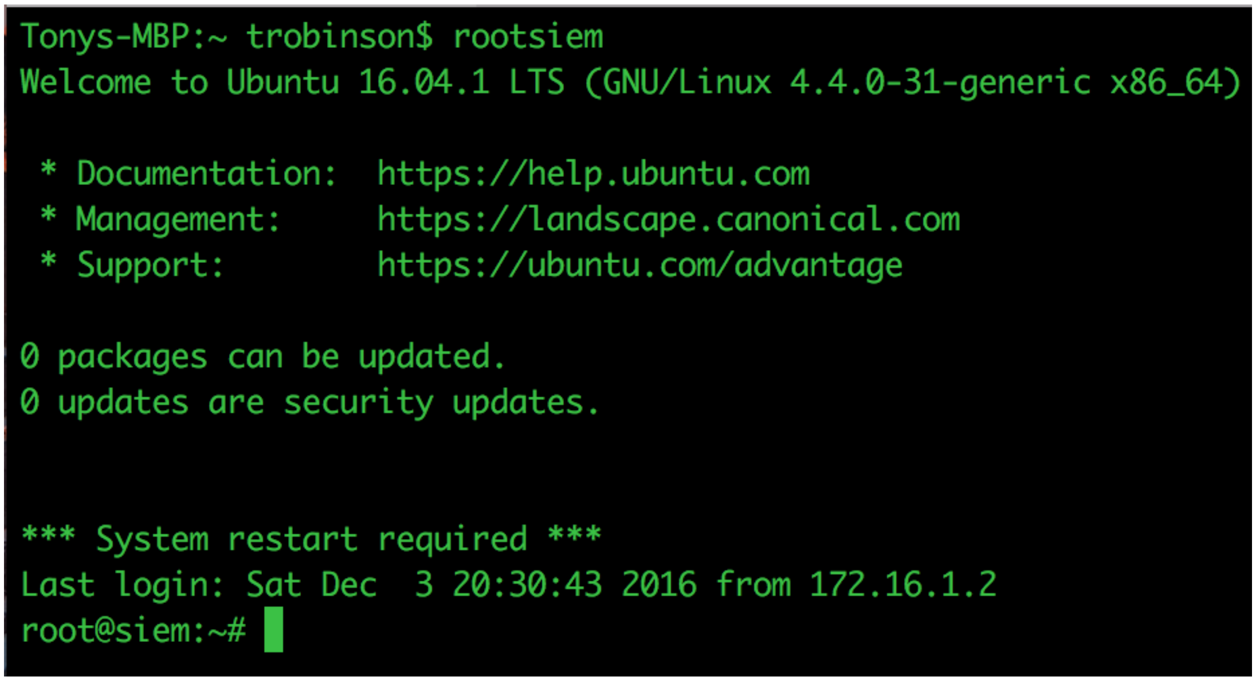
```
Tonys-MBP:~ trobinson$ ssh root@172.16.1.3
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


*** System restart required ***

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@siem:~# 
```

Note: If you are using Linux, Unix or OS X, you may want to review the section on using the alias command in order to add new or r modify existing aliases in your local user's `.bashrc` or `.bash_profile`, to simplify opening SSH connections to the SIEM VM as user `root`. You can either use your favorite text editor to change (or create) any of said files, or you can simply create and append a new, unique alias to one of them by running a command similar to the one displayed in the picture below. Make sure that it is transparent that this alias logs you in to the SIEM VM as user `root`.



```
Tonys-MBP:~ trobinson$ echo "alias rootsiem='ssh root@172.16.1.3'" >> ~/.bash_profile
```

```
Tonys-MBP:~ trobinson$ source ~/.bash_profile; alias
alias ips='ssh ayy@172.16.1.4'
alias rootsiem='ssh root@172.16.1.3'
alias siem='ssh ayy@172.16.1.3'
```



```
Tonys-MBP:~ trobinson$ rootsiem
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Sat Dec  3 20:30:43 2016 from 172.16.1.2
root@siem:~# 
```

Disabling password authentication entirely via sshd_config

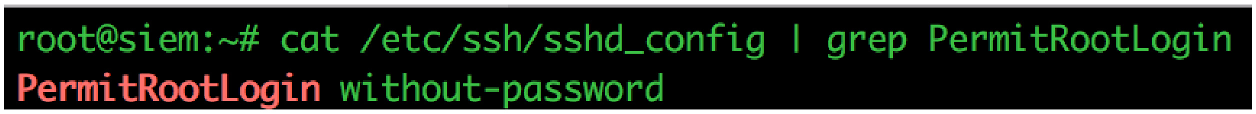
Note: Before proceeding with editing the sshd_config file on any of your VMs, you will want to make a backup of the `sshd_config` file to revert to in case one of the modifications goes wrong. Run the command

```
cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
```

to make a backup copy. If you run into any problems, you can restore the backup by running the command

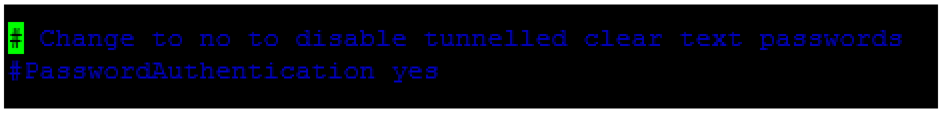
```
cp /etc/ssh/sshd_config.bak /etc/ssh/sshd_config
```

While the local login for the user `root` is disabled in Ubuntu Linux, its SSH server (sshd) is set to allow connections for this account, as long as key-based authentication is used for the attempted connection (and the corresponding `authorized_keys` file has been set up accordingly). This is because of the `PermitRootLogin without-password` directive in the `/etc/ssh/sshd_config`. If for some reason `PermitRootLogin` is set to anything other than `without-password`, or `prohibit-password`, modify the directive to either of those settings. (When used with said directive, the values `without-password` and `prohibit-password` can be used interchangeably. You may encounter one or the other, depending on the Linux distro used.)



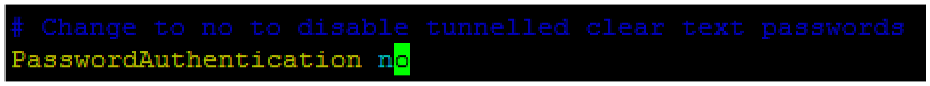
```
root@siem:~# cat /etc/ssh/sshd_config | grep PermitRootLogin
PermitRootLogin without-password
```

Additionally, we need to edit the `PasswordAuthentication` directive in `/etc/ssh/sshd_config`, setting its value from `yes` to `no`. Change respective line in the file from



```
# Change to no to disable tunnelled clear text passwords
#PasswordAuthentication yes
```

to



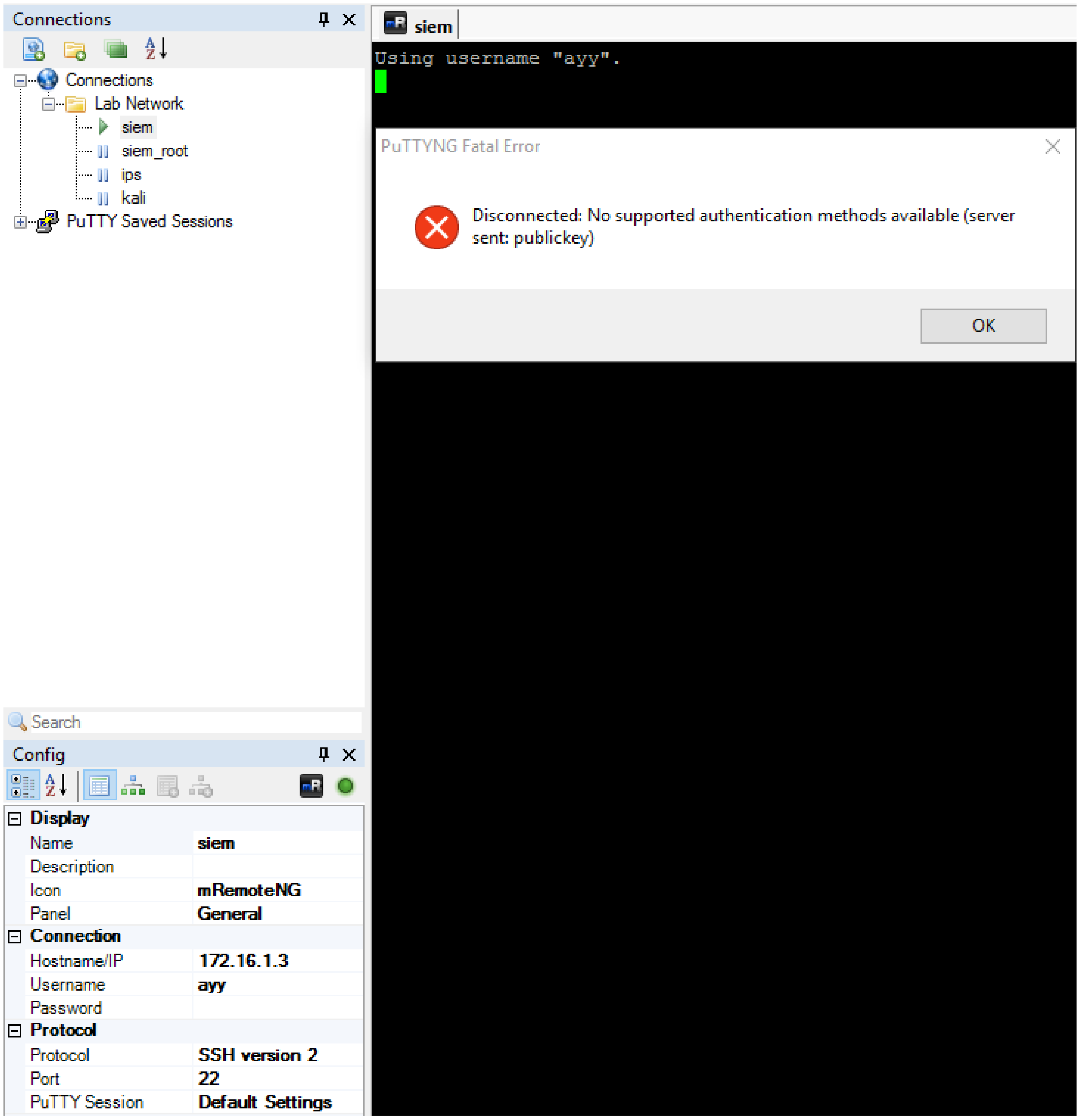
```
# Change to no to disable tunnelled clear text passwords
PasswordAuthentication no
```

Remember this in the event you ever have a need to revert these changes.

Save the `/etc/ssh/sshd_config` file, then restart the SSH service through the command:

```
service ssh restart
```

This change of setting disabled the use of passwords to authenticate SSH sessions entirely. As a consequence, the only means left for connecting to the remote system over SSH is key-based authentication. **This affects all other user accounts on the customized system as well!**



```
Using username "ayy".
```

PuTTYNG Fatal Error

Disconnected: No supported authentication methods available (server sent: publickey)

OK

Display	
Name	siem
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	172.16.1.3
Username	ayy
Password	
Protocol	
Protocol	SSH version 2
Port	22
PuTTY Session	Default Settings

In the illustration above, I attempted to log in to the SIEM VM with the account of user ‘ayy’, using the default PuTTY session instead of key-based authentication, and the connection was dropped immediately.

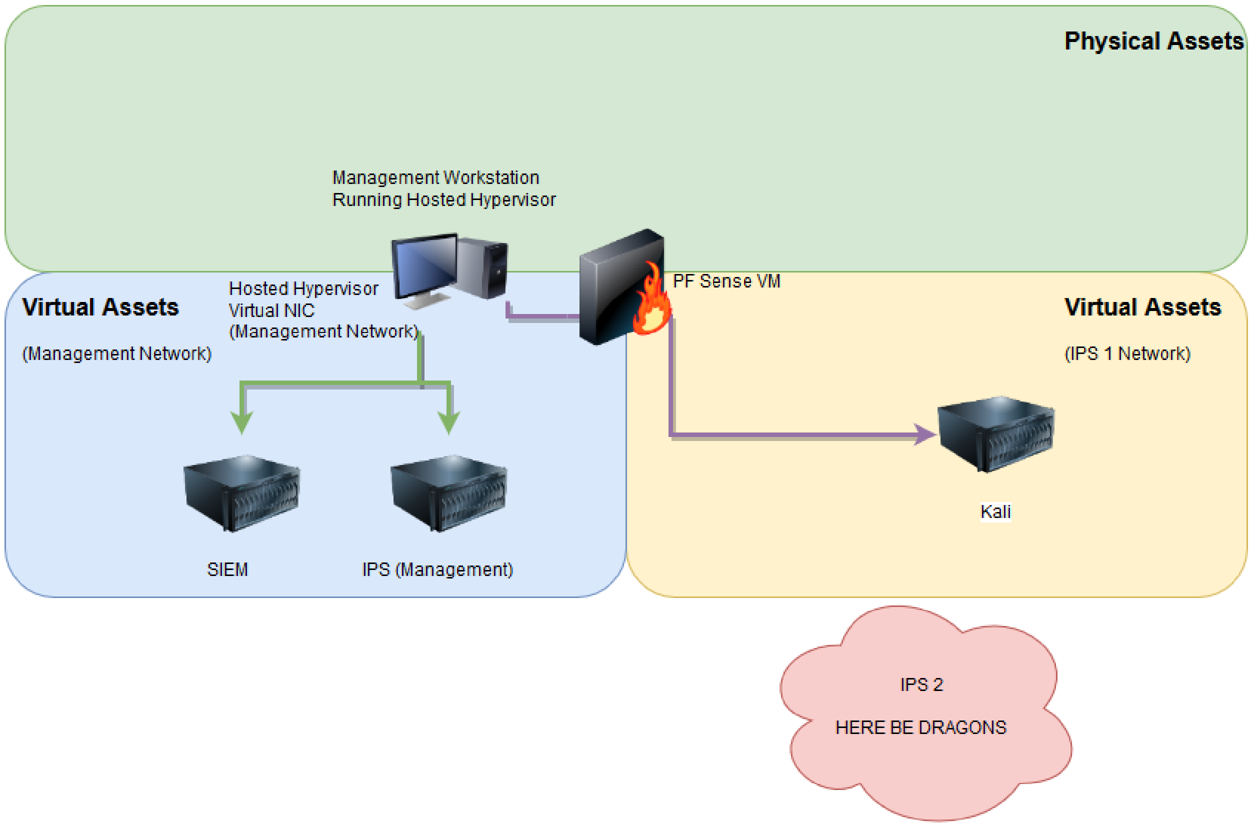
Disabling the use of passwords to authenticate SSH sessions provides an extra layer of security, but also means you must keep control of your SSH private and public keys, ensuring they remain secure. If you lose the keys, the only way left to access systems adjusted as described above is through the hypervisor’s console connection to the VM.

Note: If enabled for the `root` user, key-based authentication can be used with the Linux/Unix scp command or tools like WinSCP on Windows to authenticate this account.

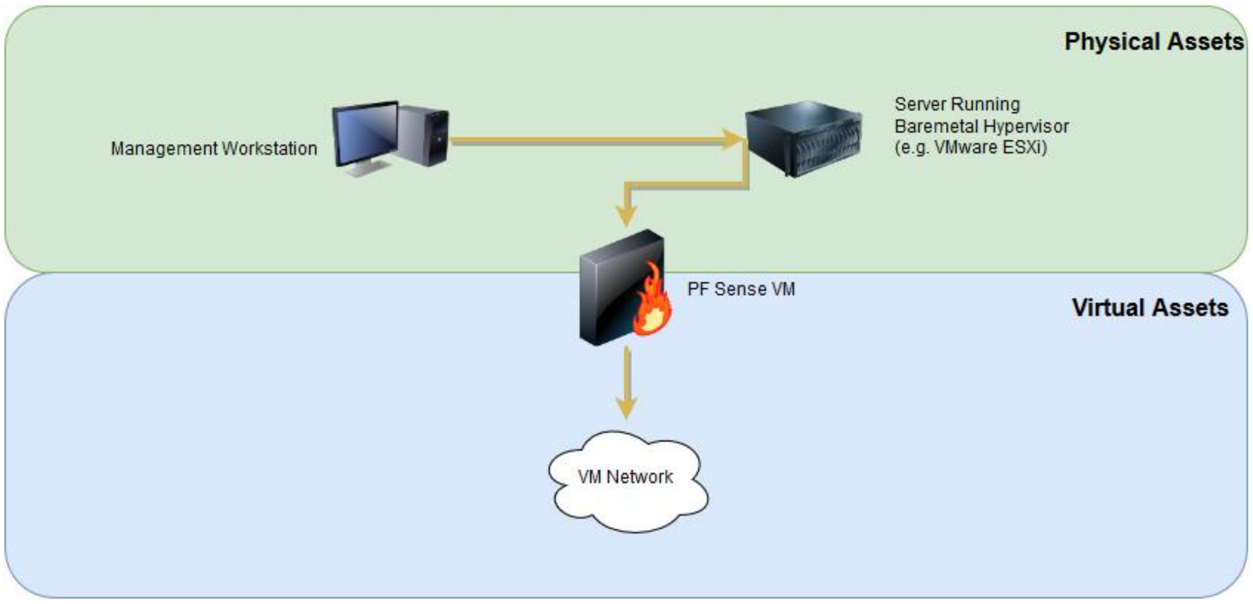
I'm going to repeat the same general warning that I gave you in the section on enabling SSH via the `root` account: it's a nice-to-have feature, fast, convenient, and all of that, but best practice in the real-world is to transfer files via a non-root user if possible, and to elevate privileges to the `root` user only when really necessary. That being said, this is our lab, we've taken fairly strong precautions against shooting ourselves in the foot, we're not running anything mission-critical, and we're not storing any sensitive data. So, if you want to SCP stuff as root, have fun, know the risks involved, and **be careful**.

Network Design Factors When Working with bare-metal Hypervisors

So far, this book has provided a great deal of instruction on how to create virtual machines and how to set up virtual network infrastructures, yet almost all of the virtualization platforms covered in this document are hosted hypervisors, the only exception being VMware ESXi (VMware vSphere Hypervisor). Consider the following illustration:



In the illustration above, our management workstation (a physical asset) is running a hosted hypervisor. This setup allows us to create a virtual network card on our host system, directly connected to the *Management* virtual network. This allows us to access virtual machines on the *Management* network easily, since the virtual network card and the virtual machines are considered to be on the same local network. The only virtual machine in the diagram above we need firewall rules and static routes to communicate with is the Kali Linux VM, which is on the *IPS* network. Compare that network diagram to this one:



This illustration reflects a lab environment composed of a management workstation and a system running a bare-metal hypervisor, hosting the lab network, and all of the virtual machines (the cloud labeled *VM Network*) we want to access. As you can see, the management workstation, a separate physical system, has to be configured to route its traffic through the pfSense VM, which then has to be configured to allow access to the lab networks. In order to achieve this, we need to setup static routes for both of the lab networks, as well as configure firewall rules that allow our management workstation to access each of the lab's VMs over the network, for the specific services we require access to.

The upcoming sections detail what static routes and/or firewall rules management workstations require in order to interact with virtual machines in your lab when they are hosted on a bare-metal hypervisor.. There is also a small section with regards to DHCP and how to configure static DHCP allocations on the physical network your management workstation, bare-metal hypervisor and pfSense WAN interface are deployed on to ensure that their IP addresses don't change. Finally, there is a detailed section on how to deal with specific scenarios where your management workstation may be in a network where you cannot statically assign its IP address, or guarantee a static DHCP allocation. This section will define how to create either a physical or a virtual machine bastion host (referred to as a “jump box”). If required, this jump box, when configured with a static IP address or DHCP allocation, can be used to access the hypervisor, pfSense webConfigurator, and all of the VMs on the hypervisor through the use of SSH tunnels (which we will also be discussing in this chapter).

Prereqs

This guide assumes that you are running ESXi 6.x or higher, with the latest updates available installed, with the ESXi lab environment configured as instructed in the VMWare vSphere Hypervisor setup guide (meaning all the virtual networks, as well as all the virtual machines presented in said section have already been created, and the pfSense VM can be accessed and managed via its webConfigurator UI). It is presumed that the *Management* and *IPS* virtual networks are set to 172.16.1.0/24 and 172.16.2.0/24, respectively. Most of the commands and illustrations in this guide assume that your management workstation is running Microsoft Windows 7, 8, or 10. However, for the OS X, Linux and BSD users out there, OS-specific commands and instructions will be provided on an as-needed basis.

This guide also assumes that the management interface of the ESXi server, the WAN interface of the pfSense firewall VM, and the network card of the workstation (or Bastion Host/jump box) being used to manage your bare-metal hypervisor and the virtual machines hosted on it, all either have static IP addresses or static DHCP reservations. This is in order to ensure that your lab network and bare-metal hypervisor can be easily accessed and managed at all times.

Just to reiterate, at an absolute minimum, you need to set up a static IP address or static DHCP allocation for your ESXi server’s management interface, the WAN interface of your pfSense VM, and for the network card of an additional system (whether that system is a management workstation OR a jump box) on a physical network that can access the ESXi server and the pfSense WAN interface in order to manage your lab network.

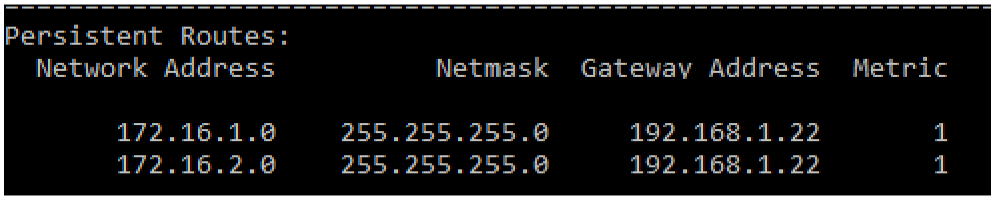
Creating Static Routes

As mentioned above, managing a lab environment that is based on a bare-metal hypervisor, using a dedicated management workstation, requires the creation of custom network routes to be able to interact with the lab virtual machines over the network. On the network I used to create the labs for this guide, I assigned IP addresses from the 192.168.1.0/24 range to my physical machines. The WAN interface of the pfSense VM was given a static DHCP allocation, the IP address being 192.168.1.22. With this information, we can create static routes to the 172.16.1.0/24 and the 172.16.2.0/24 networks as necessary. If you’re using Windows 7 or greater, open an elevated command prompt, then execute the following commands, one at a time:

```
route -p add 172.16.1.0 mask 255.255.255.0 192.168.1.22
route -p add 172.16.2.0 mask 255.255.255.0 192.168.1.22
```

Replace the IP address 192.168.1.22 in the statements above with one you assigned to the WAN interface of the pfSense firewall VM. To check the updated routing information of your system, just run:

```
route print
```



```
===========================================================================
Persistent Routes:
  Network Address          Netmask  Gateway Address  Metric

       172.16.1.0    255.255.255.0     192.168.1.22       1
       172.16.2.0    255.255.255.0     192.168.1.22       1
```

If you receive an output with entries similar to the ones in the illustration above, you have successfully added the static routes needed. Additionally, these routes should be persistant, meaning they will automatically be recreated on reboot.

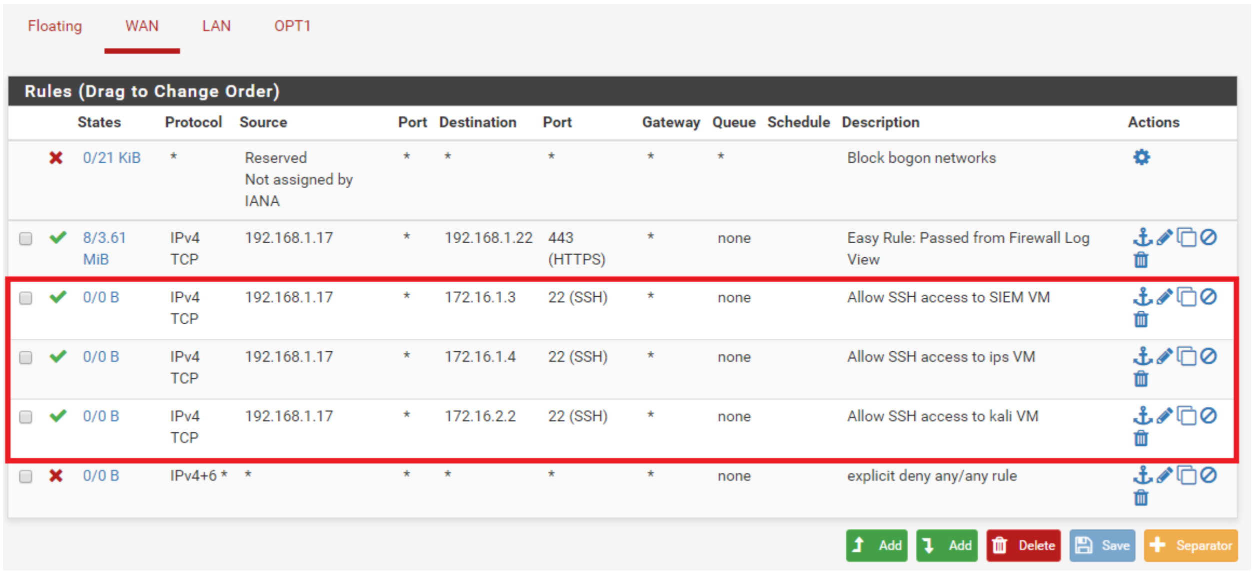
In case that you're running OS X or a Linux distro on your management workstation, check out the corresponding Remote Access guide. Specifically, if you're using Linux, pay attention to the sections detailing the `ip` command, while OS X and BSD users can read up on the `route` command. After having configured the necessary routes accordingly, you need to apply a method to ensure the routes persist upon reboot. Linux and BSD users should check out the section on using `/etc/rc.local`. Those of you working with OS X are referred to the section OS X Route Persistence for Bare-metal Hypervisors.

Creating Firewall Rules

Note: If you plan on configuring a bastion host/jump box, you may want to skip over this section for now. Ensure that your management workstation is able to access the pfSense webConfigurator via the *WAN* IP address, and continue to the next section.

In addition to the data packets being routed to the correct network, we have to ensure that the respective network traffic is actually permitted to reach its destination. To achieve this, we have to add firewall rules to the *WAN* interface to allow the traffic inbound. You need to log in to the web interface of your pfSense firewall VM via the *WAN* interface. If you followed the ESXi setup guide, you should have already created an allow firewall rule, via the `easyrule` utility, to permit your management workstation access to the webConfigurator UI. Didn't do that? Check out the Network Configuration section for the pfSense VM to find out how to use pfSense's `pfctl` and `easyrule` command line utilities, in order to allow your management workstation to access the firewall from the WAN interface, or to add another management workstation, etc.

Once you have logged in to the webConfigurator of the pfSense VM, navigate to *Firewall > Rules*, and ensure that the *WAN* interface is selected. All of the firewall rules we'll be adding for remote access refer to the *WAN* interface. In our case, we have to add three firewall rules, each of them handling traffic for port 22/tcp (SSH). The destination IP addresses (use the *single host or alias* option) are 172.16.1.3, 172.16.1.4, and 172.16.2.2,for the SIEM IPS, and Kali virtual machines, respectively. The source IP address for each rule should be the one assigned to your management workstation, which in my case is 192.168.1.17 (again, use the *single host or alias* option, and, of course, be sure to modify the source IP address to match the one you did set on/for your management system). When you are done, your firewall rule set on the WAN interface should look similar to the one displayed in the picture below:



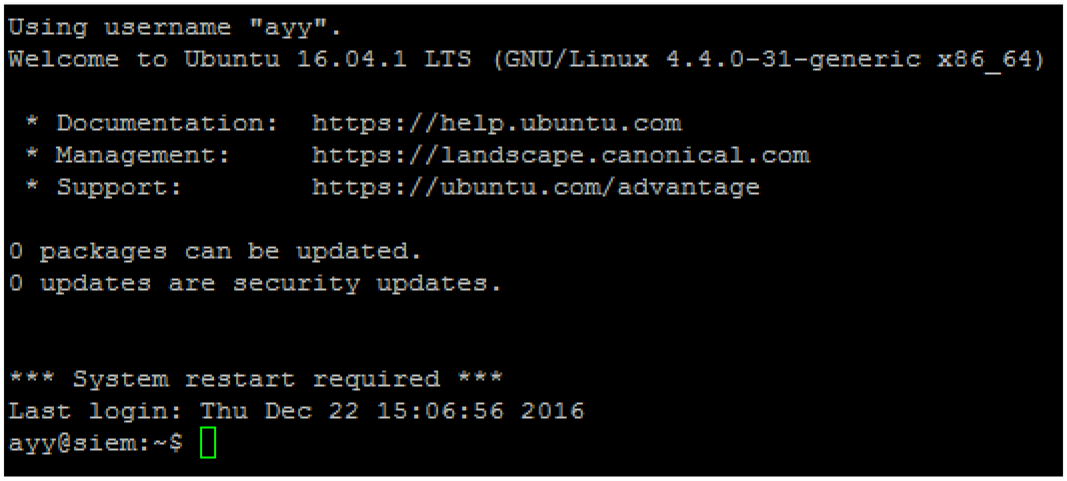
Rules (Drag to Change Order)

		States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	✗	0/21 KiB	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
☐	✓	8/3.61 MiB	IPv4 TCP	192.168.1.17	*	192.168.1.22	443 (HTTPS)	*	none		Easy Rule: Passed from Firewall Log View	
☐	✓	0/0 B	IPv4 TCP	192.168.1.17	*	172.16.1.3	22 (SSH)	*	none		Allow SSH access to SIEM VM	
☐	✓	0/0 B	IPv4 TCP	192.168.1.17	*	172.16.1.4	22 (SSH)	*	none		Allow SSH access to ips VM	
☐	✓	0/0 B	IPv4 TCP	192.168.1.17	*	172.16.2.2	22 (SSH)	*	none		Allow SSH access to kali VM	
☐	✗	0/0 B	IPv4+6 *	*	*	*	*	*	none		explicit deny any/any rule	

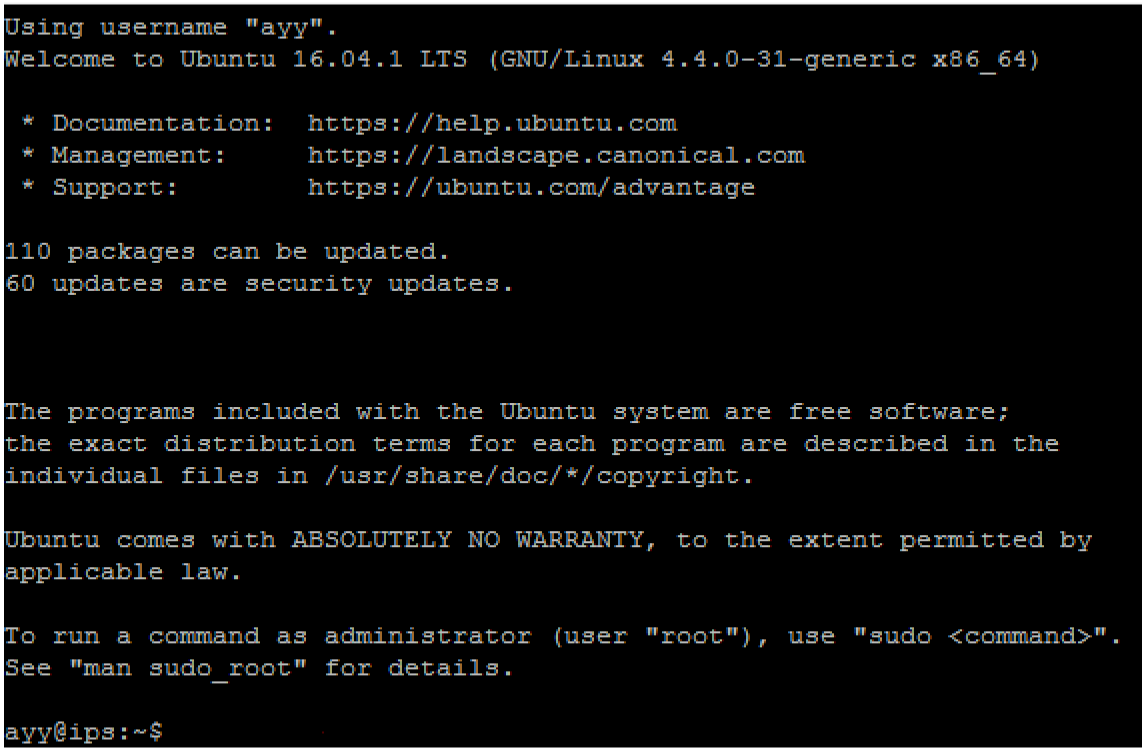
The new rules should be placed BELOW the pass rule that allows your management workstation to access to the pfSense's web interface, but ABOVE the explicit deny any/any rule at the bottom of the rule list (that is, in case you elected to create this rule). If you need to change the order of the rules to adjust the sequence of their evaluation, drag and drop the row(s) in question to their new position(s) in the list, then click the *Save* button.

The easiest way to verify if those new firewall rules are working is to attempt to establish SSH connections to the SIEM, IPS and Kali virtual machines. Refer to the Remote Lab Management chapter to learn how to setup SSH connectivity to your virtual machines. Depending on the OS of your management workstation, follow the Windows Remote Access guide or the Linux, BSD, and OS X Remote Access guide, up to and including the respective section on configuring key-based authentication for your remote systems. Additionally, if you haven't already, you may also want to have a look at the How to Enable SSH on Kali Linux guide, in order to be able to establish SSH sessions to your Kali Linux VM.

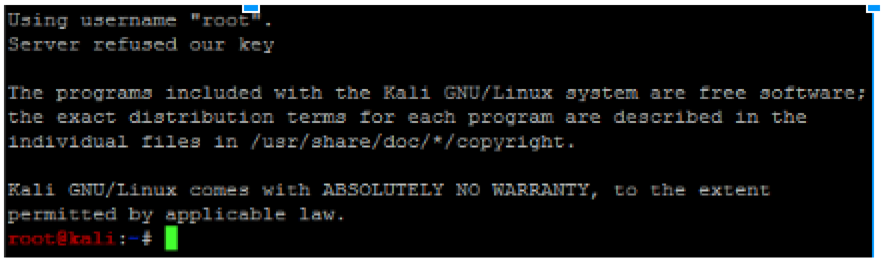
If everything is working as intended, you should be able to establish SSH connections to the SIEM, IPS, and Kali Linux virtual machines, with the output of the respective sessions looking similar to those shown in the pictures below:



```
Using username "ayy".
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Thu Dec 22 15:06:56 2016
ayy@siem:~$
```



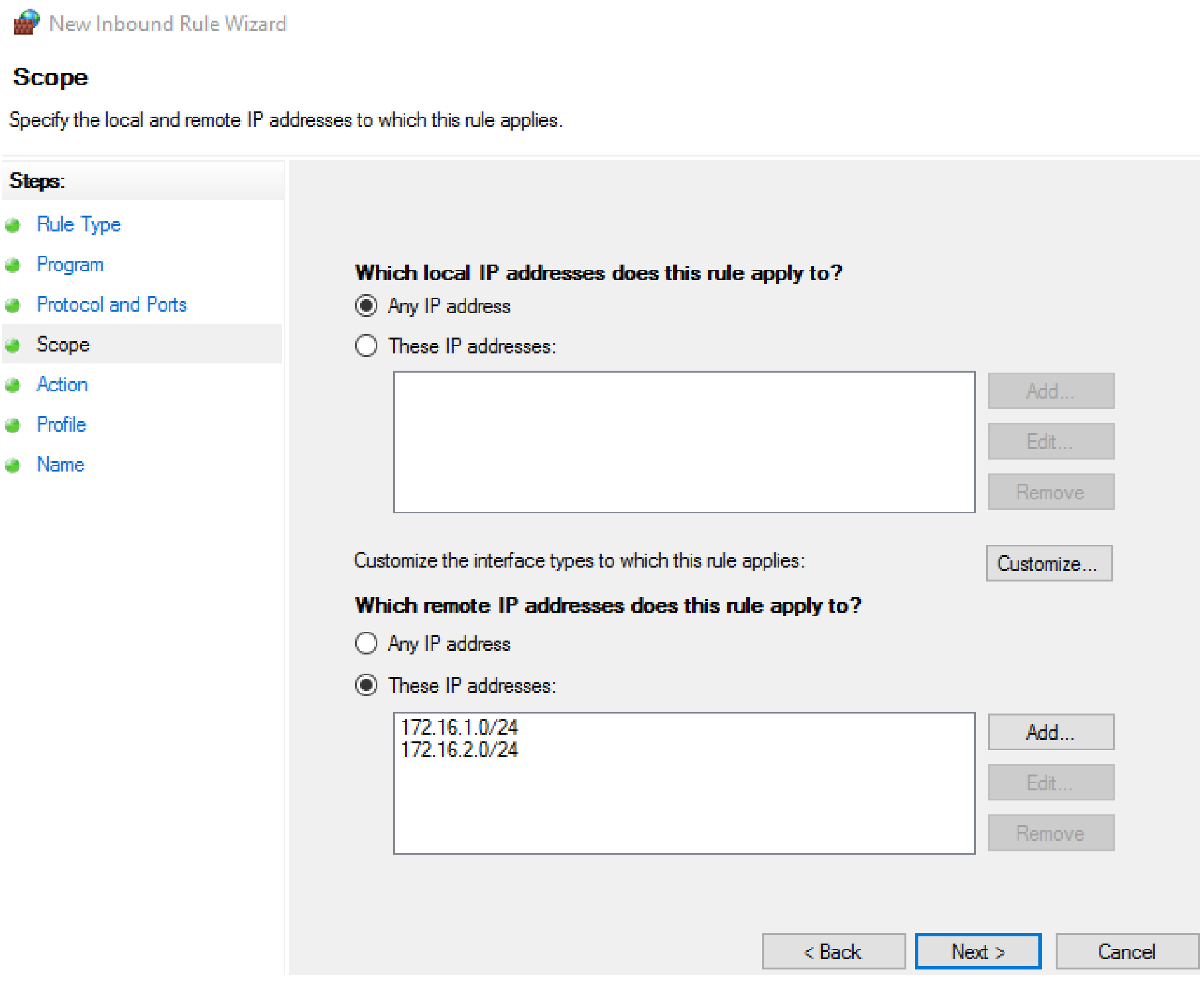
```
Using username "ayy".
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

110 packages can be updated.
60 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ayy@ips:~$
```



```
Using username "root".
Server refused our key

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@kali:~#
```

As you can see in the illustrations above, I was able to connect and authenticate successfully to each of the remote systems. At this point, if you haven't already configured key-based authentication, I suggest that you visit the respective part of the the remote access guide that corresponds with your management workstation to do so. I also recommend reading the guide on how to disable password authentication on your lab virtual machines as well.

So far, we have discussed firewall rules to control/allow the network traffic from our management workstation to the machines of our lab environment, using the pfSense system. It is sensible to take precautions for the management system, too. If you're using Windows as OS on your management workstation, you may want to read the guide on Defense in Depth for Windows Hosted Hypervisors, specifically the section Using Windows Firewall to Limit Exposure of Windows Hypervisor. Out of an abundance of caution, you probably want to make sure that you block ANY and all connections initiated from hosts on the 172.16.1.0/24, and 172.16.2.0/24 networks.


New Inbound Rule Wizard

Scope

Specify the local and remote IP addresses to which this rule applies.

Steps:
- Rule Type
- Program
- Protocol and Ports
- Scope
- Action
- Profile
- Name

Which local IP addresses does this rule apply to?
[✓ Any IP address] These IP addresses:

Customize the interface types to which this rule applies:

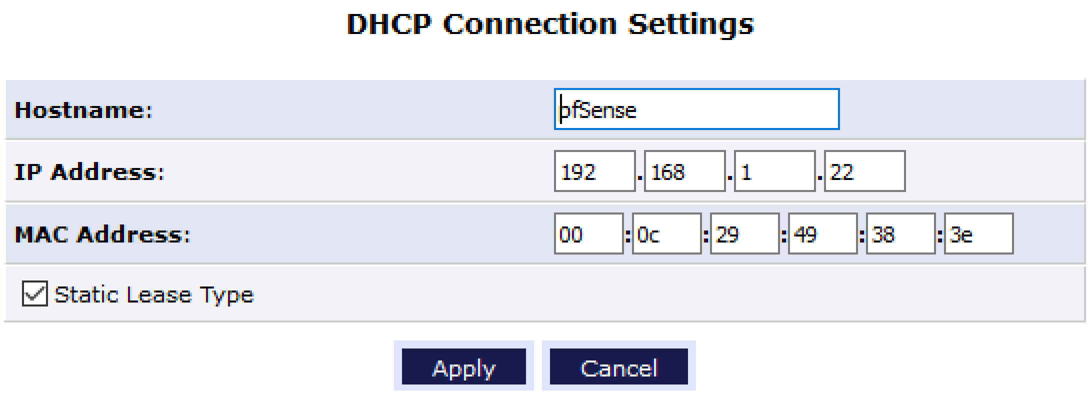
Which remote IP addresses does this rule apply to?
Any IP address [✓ These IP addresses:]

172.16.1.0/24
172.16.2.0/24

Dealing with DHCP

Using DHCP to assign IP addresses to certain components of a lab environment can lead to remote access problems when dealing with VMs hosted on a bare-metal hypervisor. Setting up static routes and firewall rules requires the IP addresses that you're routing to, as well as the source/destination addresses for the firewall rules to remain constant. As a part of the network setup portion for all of the hypervisors in this guide, we configured static DHCP mappings for all the virtual machines in our lab networks. This was to make sure that the IP addresses of each of our VMs will always remain the same, even after reboot, without having to set IP addresses, default gateways, or DNS information manually on each virtual machine individually.

As mentioned above, if at all possible, your management workstation, ESXi server, and WAN interface of the pfSense firewall VM should either have static DHCP mappings based on the MAC addresses of these systems, or static IP addresses configured on the corresponding network interfaces themselves. Without knowing what your home or office network looks like, I can say with some measure of confidence that most network devices (even SOHO routers provided by most ISPs) allow you to configure static DHCP mappings or "IP Address Reservations".


DHCP Connection Settings

Hostname:	pfSense
IP Address:	192 . 168 . 1 . 22
MAC Address:	00 : 0c : 29 : 49 : 38 : 3e
✓ Static Lease Type	

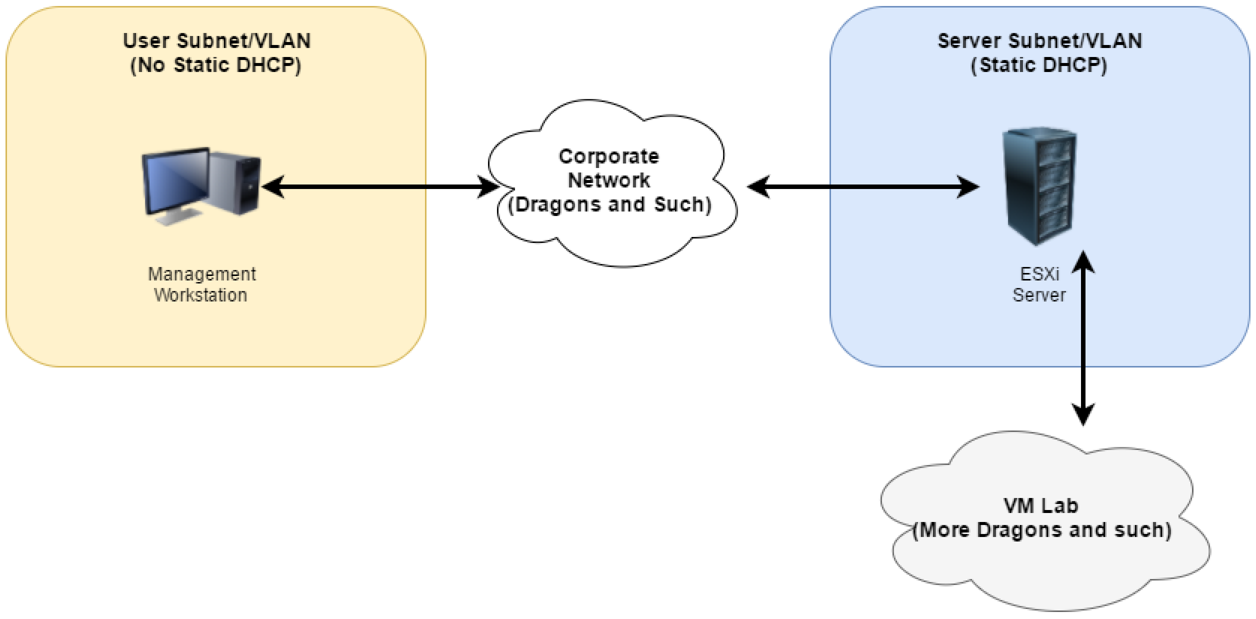
The screen capture above was taken from my home network router, an Actiontec device my ISP provided me. Notice the option to set the DHCP lease type to static. If this thing is capable of static DHCP allocations, I'm sure that whatever gear you are using supports this functionality as well! Consult with the device documentation or your ISP's tech support department if you need help.

As an alternative, you can check the DHCP settings of your gear to identify the range of IP addresses used for the network(s) your workstation, ESXi Server and/or pfSense VM are connected to, and simply configure a static settings on the respective system's NIC, using an IP address from the same local network, but outside of the DHCP address pool. For example, my home network has a SOHO router my ISP provides, with the internal network set to 192.168.1.0/24. The DHCP pool of the device is 192.168.1.2 - 192.168.1.100. This means that for my workstation, the ESXi server, and the WAN interface on my pfSense VM, I could use any IP address from the range 192.168.1.101 - 192.168.1.254 as static setting for each of the systems mentioned above. Bear in mind that if you set static IP addresses, you need to manually set the default gateway for these systems, as well as the DNS servers to use, too.

Jump Boxing

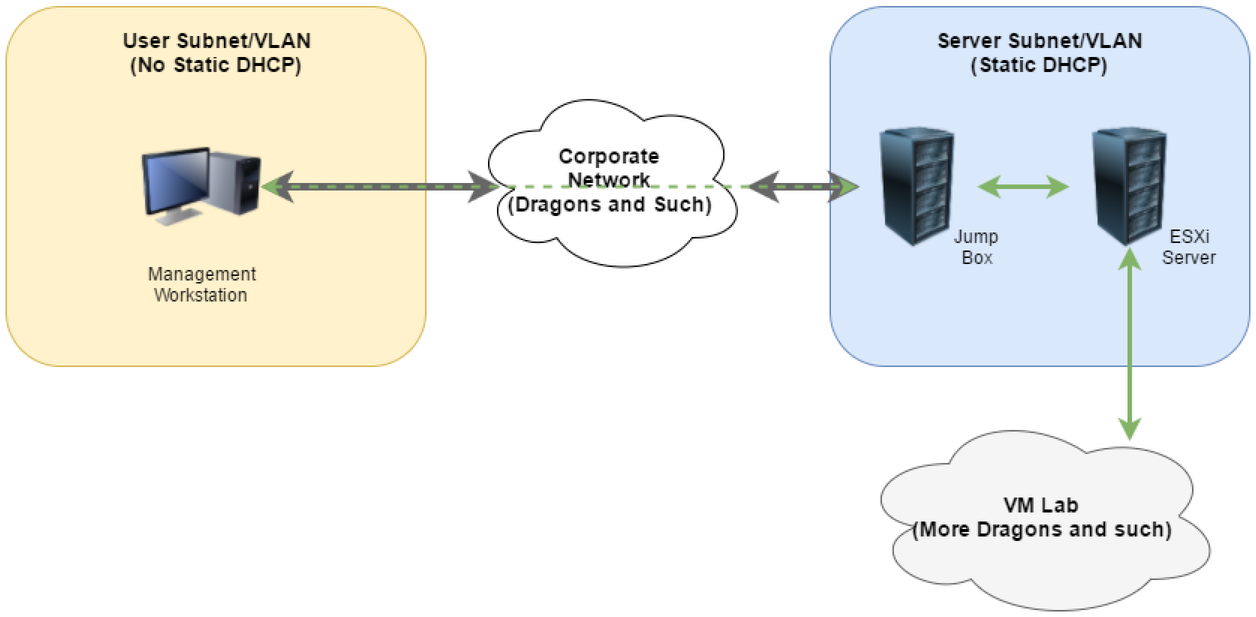
Note: The remainder of this chapter details how to create and configure a jump box for accessing your lab environment, using a Raspberry Pi, an additional virtual machine, or any other physical hardware you happen to have available. If you were able to configure your management workstation with a static DHCP address allocation to ensure access to your lab, you may opt to skip over the sections related to jump boxing entirely. However, if your lab is intended to support incoming connections from multiple users and/or machines, knowing how to set up a jump box would be a more scalable solution for allowing multiple users to access to the virtual machine lab more reliably.

What are you supposed to do when your management workstation resides in a network where static network interface configuration or static DHCP address allocations are not available?



In most enterprise networks, end-user systems usually reside in a subnet and/or vlan that is configured to use DHCP. Unless you are VERY good friends with your network admins, you probably won't be able to get a static DHCP mapping for your workstation. There may be technical reasons, or it may just be a policy. However, on the server subnet, the network and system admins usually have static DHCP address allocations or fixed IP address settings configured for the attached network interfaces. With that as a starting point, we can install a small embedded system (e.g. a Raspberry Pi), or create another small VM, which then can be connected to the server subnet, just like the ESXi server's management interface, and the pfSense VM's WAN interface. By adding firewall rules to let this system access the virtual lab networks, and by setting up the necessary static routes on this machine, we are able to redirect the traffic from our management workstation through this system into our lab environment. A system used to forward and broker connections from one network to another network in this manner is known as a bastion host, or a jump box.

Note: Jump boxes are also referred to as proxies or redirectors. These names are interchangeable. Jump boxes are used as a connection broker to forward your network connections to another destination you cannot or do not want to interact with directly. Jump boxes are used by IT and information security staff to access assets in other network segments securely, in a manner that allows connections to/from sensitive networks to be audited (if needed), with the jump box acting as both a choke point, and a way to audit connections to sensitive assets.



In the diagram above, the arrow headed grey lines denote a tunnel connection from our management workstation to the jump box, which is located on the same subnet/vlan as our ESXi server. The green lines to the ESXi server (more precisely to the pfSense VM on the ESXi server) and the VM lab network represent connections from our management workstation that were forwarded from the jump box system to destinations in the VM lab.

Using a Raspberry Pi as a Jump Box

I had a Raspberry Pi Model B (revision 1) from some time ago at hand, so I wrote this guide using this hardware, specifically. The process described in the following should be nearly identical for most of the Raspberry Pi models, especially for those that come equipped with an Ethernet port. This guide also assumes that you'll be using wired Ethernet to connect to your Raspberry Pi jump box, so if you want to use Wi-Fi, you'll be on your own (though there shouldn't be too much of a difference, really).

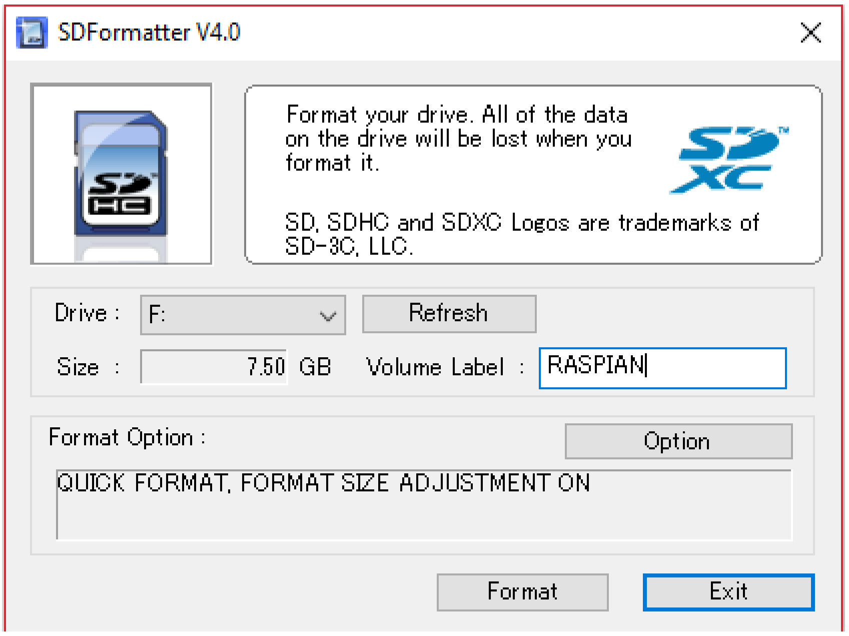
Why go through the trouble of setting up a physical jump box, you ask? Having a physical jump box has some advantages to it that you should consider. A lot of newer servers and server motherboards come with small, embedded systems integrated into their chassis, often referred to as LOM, IPMI or some variation. As long as the server itself is plugged in and getting power, those integrated systems can be accessed over the network, providing a variety of functions that normally require physical access to a machine. Among other things, they make it possible to power the server on or off physically, and to redirect the server's console output over the network. Having a separate physical jump box not only allows us to access our lab's VM systems, but to connect to and use these management systems, too. For example, if the power on the server goes out, and you can't physically access this machine, having a physical jump box at least gives to possibility to try to troubleshoot the issue remotely (e.g. virtually pushing the power button etc.).

While you don't have to use a Raspberry Pi specifically, one of its advantages that deserves attention is that it doesn't require much in the way of power, which is supplied via a micro USB connector. Using a USB to micro USB cable, you can choose to power the Raspberry Pi by connecting it to the server itself, or by hooking it up to an AC adapter. In either case, the Raspberry Pi requires very little power to operate, making it an ideal physical jump box.

Installing the Raspbian Image to your Raspberry Pi

The fine folks at raspberrypi.org have made available a great guide for doing an initial setup of the Raspberry Pi. We're going to follow their instructions for installing Raspbian, an RPi (shorthand for Raspberry Pi) optimized version of the Debian Linux distro, onto an SD card, which we will use to boot the tiny computer. You can use any SD card you'd like, so long as it is 4GB or larger. In my case, I used a spare 8GB SD card I found lying around.

The first thing recommended in the guide is reformatting the SD card. The instructions for doing this are located at https://www.raspberrypi.org/documentation/installation/noobs.md. I was using a machine running Windows,, so I downloaded the suggested SD card formatter tool from sdcard.org/downloads, then reformatted my card. (Please note that the page link above provides instructions for formatting your SD card on both OS X and Linux too, suggesting the use of the SD card formatter tool for OS X, and of the `gparted` utility for Linux, respectively). Be sure to configure the *Format size adjustment* option to *ON* in order to resize the SD card partition to make use of the entire capacity of the card if necessary.


SDFormatter V4.0

Format your drive. All of the data on the drive will be lost when you format it.

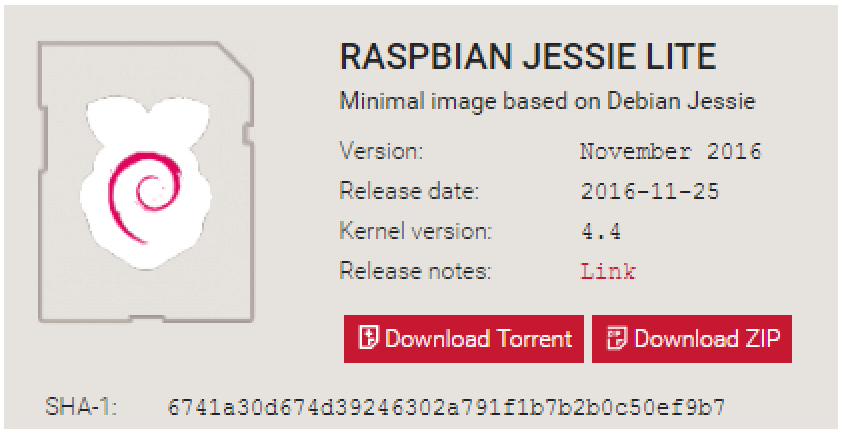
SD, SDHC and SDXC Logos are trademarks of SD-3C, LLC.

Drive : F:
Size : 7.50 GB
Volume Label : RASPIAN

Format Option :
QUICK FORMAT, FORMAT SIZE ADJUSTMENT ON

Note: If you are using an SDXC card (these are defined as SD cards over 32GB in size), make sure to follow the instructions on how to format this kind of cards on Windows, Linux, and OS X, available at https://www.raspberrypi.org/documentation/installation/sdxc_formatting.md.

Next, visit https://www.raspberrypi.org/downloads/raspbian/ to download the latest version of Raspbian (**Note:** Raspbian "Jessie" was the version used in this guide). I recommend downloading the "lite" image, because apart from the initial setup that we are going to do together, you'll probably never connect a keyboard and monitor to your RPi again (unless something breaks), let alone using the jump box as a regular Linux desktop machine. Anything that can be done to save drive space, RAM and CPU cycles is worthwhile, which makes the lite image the perfect choice.


RASPBIAN JESSIE LITE

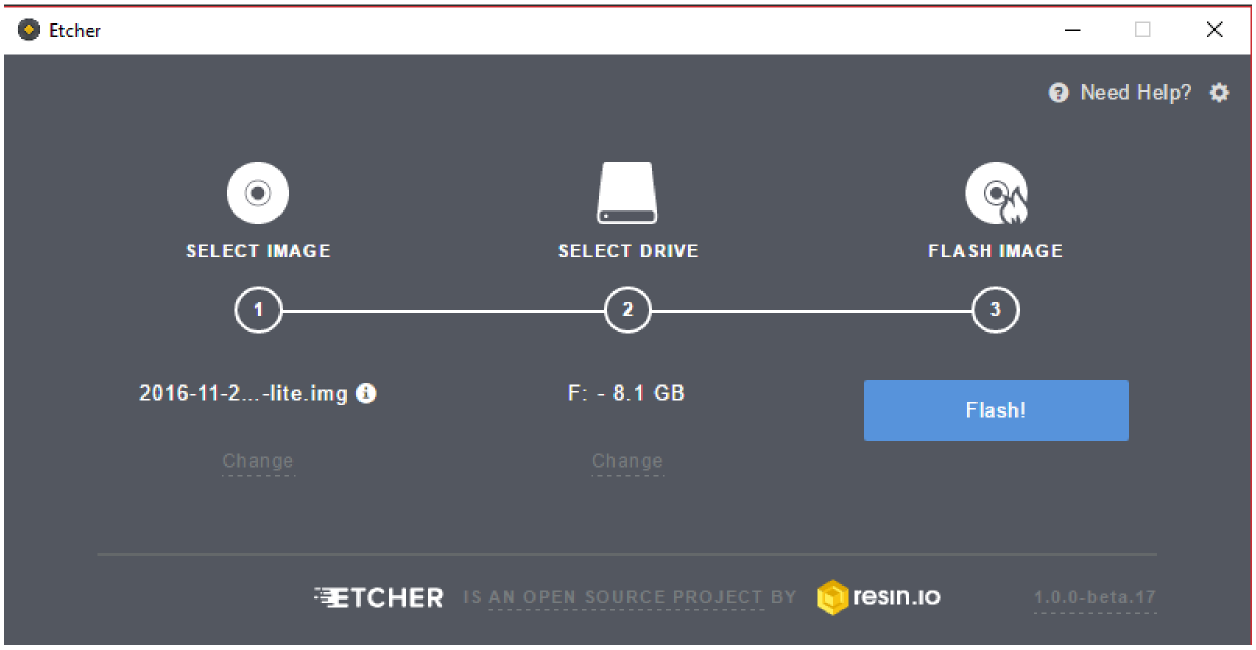
Minimal image based on Debian Jessie

Version:	November 2016
Release date:	2016-11-25
Kernel version:	4.4
Release notes:	Link

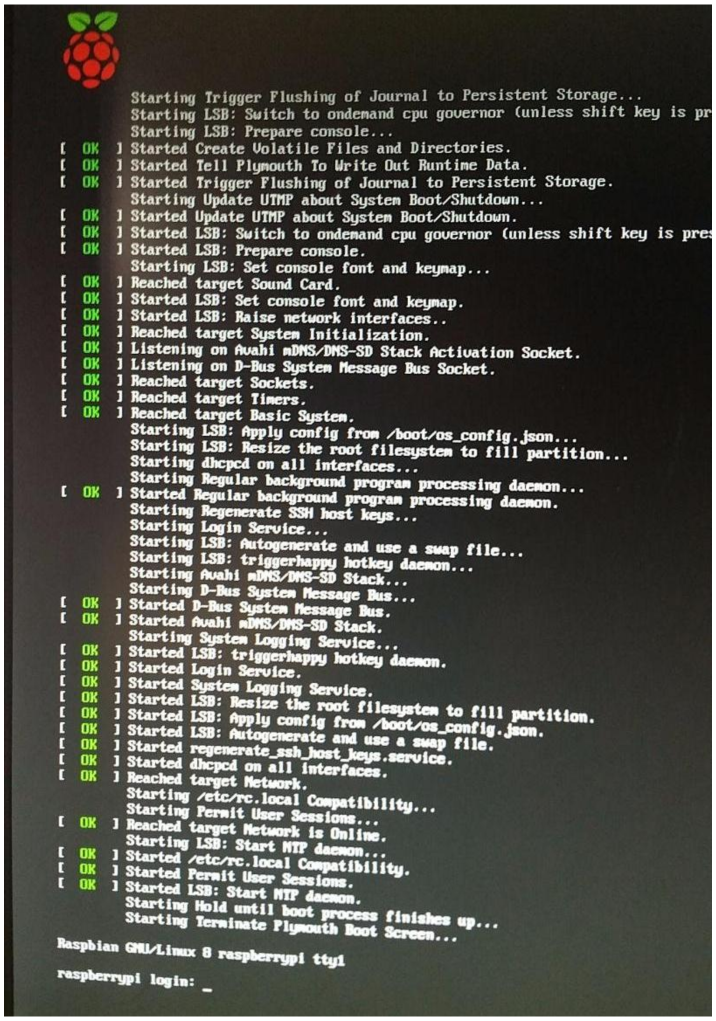
SHA-1: 6741a30d674d39246302a791f1b7b2b0c50ef9b7

After downloading the file, you need to write the image to the SD card. There are a number of image writing utilities for a variety of operating systems. The official instructions on raspberrypi.org suggest to using a graphical tool called Etcher, available from the website etcher.io that is available for Windows, Linux, and OS X, which from my personal experience fits the bill nicely. Simply select the image file, select the destination disk (the SD card), and click the *Flash!* button to write the image to the drive and have it verified after it has been written.

Note: in the past you would normally have to unzip the downloaded image file prior to writing it to an SD card. etcher is capable of unzipping the compressed image file automatically, making this process no longer necessary.



Now comes the fun part. Install the SD card into your Raspberry Pi, hook up a USB keyboard, Ethernet cable (make sure that it is attached to the same network that the ESXi server and pfSense VM’s *WAN* interface are available on), and a HDMI video device to the mini computer, and finally connect USB power to your RPi to start it up.



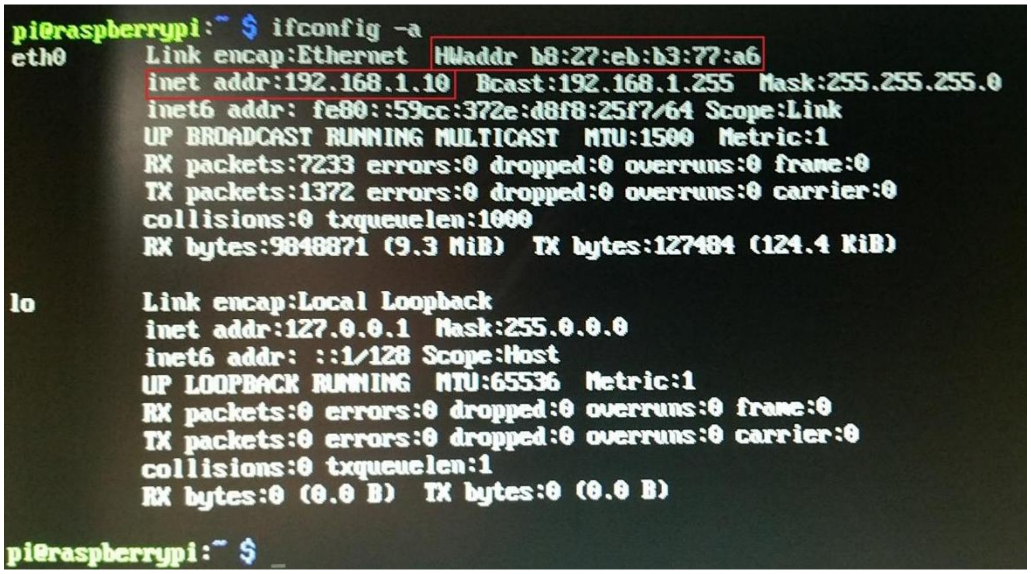
```
         Starting Trigger Flushing of Journal to Persistent Storage...
         Starting LSB: Switch to ondemand cpu governor (unless shift key is pr
         Starting LSB: Prepare console...
[  OK  ] Started Create Volatile Files and Directories.
[  OK  ] Started Tell Plymouth To Write Out Runtime Data.
[  OK  ] Started Trigger Flushing of Journal to Persistent Storage.
         Starting Update UTMP about System Boot/Shutdown...
[  OK  ] Started Update UTMP about System Boot/Shutdown.
[  OK  ] Started LSB: Switch to ondemand cpu governor (unless shift key is pres
[  OK  ] Started LSB: Prepare console.
         Starting LSB: Set console font and keymap...
[  OK  ] Reached target Sound Card.
[  OK  ] Started LSB: Set console font and keymap.
[  OK  ] Started LSB: Raise network interfaces..
[  OK  ] Reached target System Initialization.
[  OK  ] Listening on Avahi mDNS/DNS-SD Stack Activation Socket.
[  OK  ] Listening on D-Bus System Message Bus Socket.
[  OK  ] Reached target Sockets.
[  OK  ] Reached target Timers.
[  OK  ] Reached target Basic System.
         Starting LSB: Apply config from /boot/os_config.json...
         Starting LSB: Resize the root filesystem to fill partition...
         Starting dhcpcd on all interfaces...
         Starting Regular background program processing daemon...
[  OK  ] Started Regular background program processing daemon.
         Starting Regenerate SSH host keys...
         Starting Login Service...
         Starting LSB: Autogenerate and use a swap file...
         Starting LSB: triggerhappy hotkey daemon...
         Starting Avahi mDNS/DNS-SD Stack...
         Starting D-Bus System Message Bus...
[  OK  ] Started D-Bus System Message Bus.
[  OK  ] Started Avahi mDNS/DNS-SD Stack.
         Starting System Logging Service...
[  OK  ] Started LSB: triggerhappy hotkey daemon.
[  OK  ] Started Login Service.
[  OK  ] Started System Logging Service.
[  OK  ] Started LSB: Resize the root filesystem to fill partition.
[  OK  ] Started LSB: Apply config from /boot/os_config.json.
[  OK  ] Started LSB: Autogenerate and use a swap file.
[  OK  ] Started regenerate_ssh_host_keys.service.
[  OK  ] Started dhcpcd on all interfaces.
[  OK  ] Reached target Network.
         Starting /etc/rc.local Compatibility...
         Starting Permit User Sessions...
[  OK  ] Reached target Network is Online.
         Starting LSB: Start NTP daemon...
[  OK  ] Started /etc/rc.local Compatibility.
[  OK  ] Started Permit User Sessions.
[  OK  ] Started LSB: Start NTP daemon.
         Starting Hold until boot process finishes up...
         Starting Terminate Plymouth Boot Screen...

Raspbian GNU/Linux 8 raspberrypi tty1

raspberrypi login: _
```

Booting the system takes a few moments. The output on your screen should look similar to the one shown in the picture above. If you are finally greeted with a login prompt, given you have everything connected to it, your Raspberry Pi is ready to go.

Configuring Raspbian

The default credentials to log in to Raspbian are username `pi` and password `raspberry`. Enter these credentials to log in via the console, using the keyboard and monitor you attached to the Raspberry Pi. In order to use it as our jump box, we need to make sure that the RPi system has an IP address, verify the its MAC address for creating a static DHCP mapping, and enable Raspbian’s OpenSSH service.

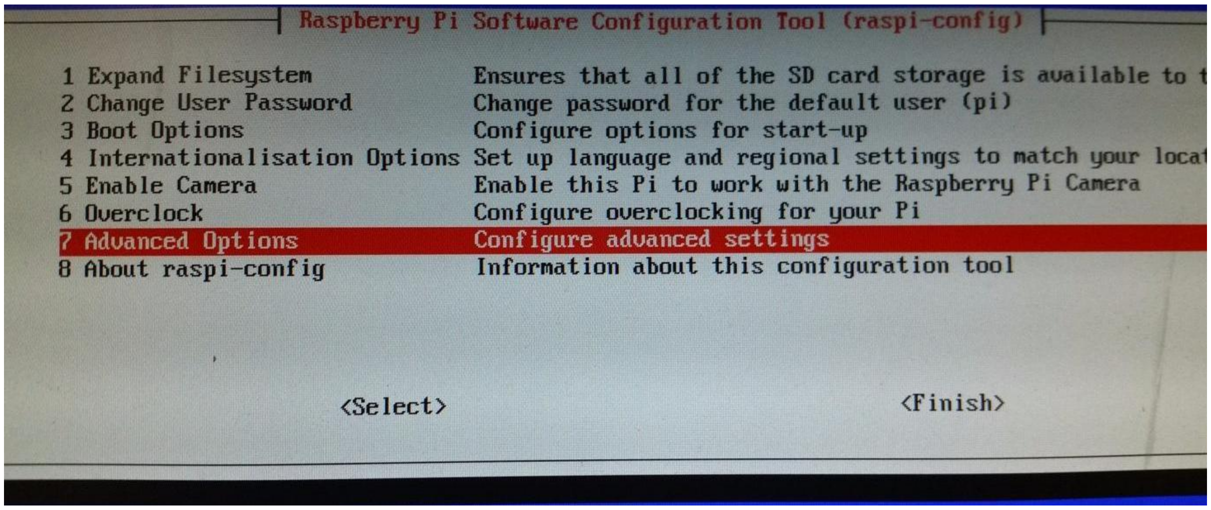
Once you have logged in, run the command `ifconfig -a` to display information about the network interfaces on the RPi. GIven that there is only one physical network adapter present, look for the block of information that starts with `eth0`, note the corresponding MAC address (labeled `HWaddr` in the `ifconfig` output), and the currently assigned IP address (named `inet addr`).



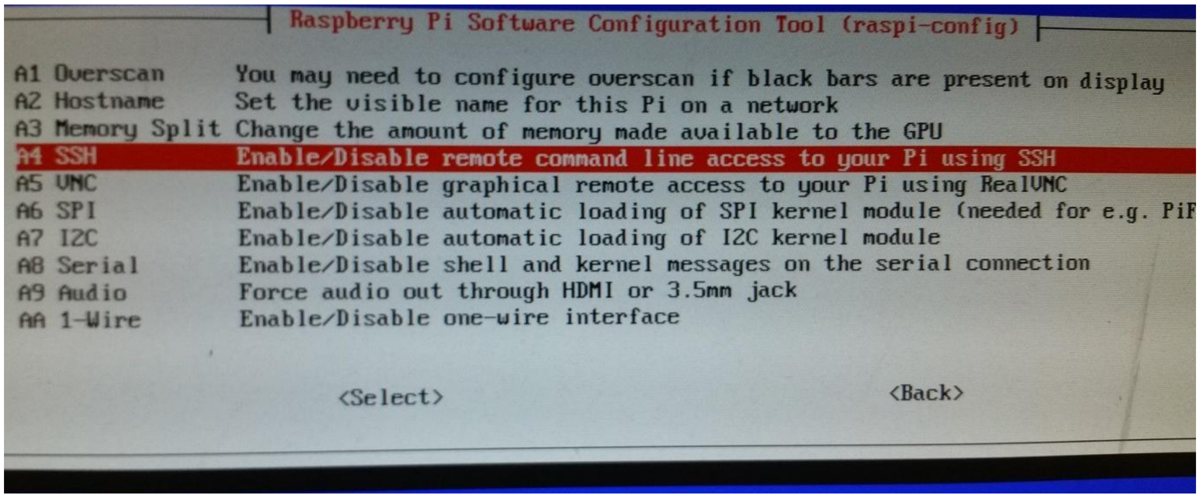
```
pi@raspberrypi:~ $ ifconfig -a
eth0      Link encap:Ethernet  HWaddr b8:27:eb:b3:77:a6
          inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::59cc:372e:d8f8:25f7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7233 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1372 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9848871 (9.3 MiB)  TX bytes:127484 (124.4 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

pi@raspberrypi:~ $ _
```

In the picture above, taken from the screen of my system, you can see that the RPi has an IP address of 192.168.1.10, and a MAC address of b8:27:eb:b3:77:a6. Document the MAC address displayed in order to configure a static DHCP mapping for this system. I would advise setting up a static DHCP mapping now, or as an alternative, configure a static IP address for the interface `eth0` on this system. Revisit the section Dealing with DHCP on instructions on how to setup static DHCP address allocations.

Next, run the command `sudo raspi-config`. A menu captioned *Raspberry Pi Software Configuration Tool (raspi-config)* will pop up, offering a variety of different choices. Select option *7 Advanced Options*, followed by option *A4 SSH* on the next page to enable the SSH service.



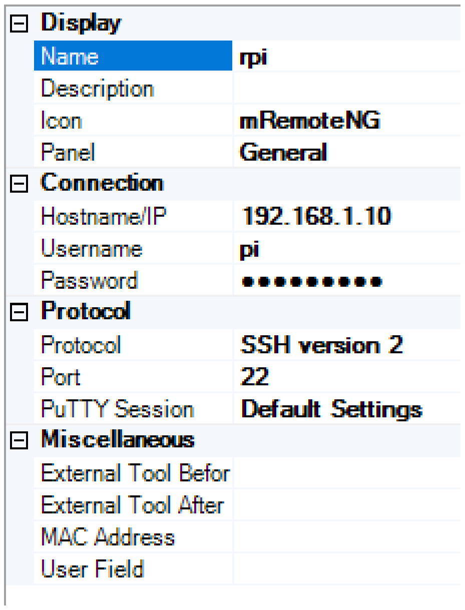
```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Expand Filesystem            Ensures that all of the SD card storage is available to t
2 Change User Password         Change password for the default user (pi)
3 Boot Options                 Configure options for start-up
4 Internationalisation Options Set up language and regional settings to match your locat
5 Enable Camera                Enable this Pi to work with the Raspberry Pi Camera
6 Overclock                    Configure overclocking for your Pi
7 Advanced Options             Configure advanced settings
8 About raspi-config           Information about this configuration tool

              <Select>                          <Finish>
```



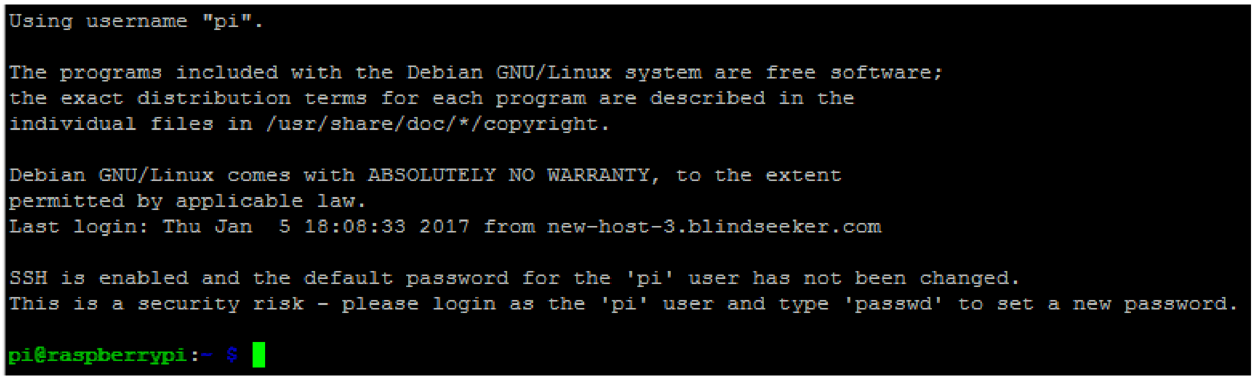
```
Raspberry Pi Software Configuration Tool (raspi-config)

A1 Overscan      You may need to configure overscan if black bars are present on display
A2 Hostname      Set the visible name for this Pi on a network
A3 Memory Split  Change the amount of memory made available to the GPU
A4 SSH           Enable/Disable remote command line access to your Pi using SSH
A5 VNC           Enable/Disable graphical remote access to your Pi using RealVNC
A6 SPI           Enable/Disable automatic loading of SPI kernel module (needed for e.g. PiF
A7 I2C           Enable/Disable automatic loading of I2C kernel module
A8 Serial        Enable/Disable shell and kernel messages on the serial connection
A9 Audio         Force audio out through HDMI or 3.5mm jack
AA 1-Wire        Enable/Disable one-wire interface

              <Select>                          <Back>
```

After enabling the SSH server, you are returned to the main menu. Hit ESC or select *Finish* to exit the Raspberry Pi Software Configuration Tool and gain access to the command line again. Type `exit` to logout of the Raspberry Pi's console.Next, we have to test if the SSH server is available now and confirm that it is possible to log in remotely to the RPi.

Using the IP address you recorded from the `ifconfig` output, try to establish an SSH connection to the Raspberry Pi, using your SSH client of choice (e.g. mRemoteNG if you are using Windows, or the `ssh` command in case you are on Linux, BSD, or OS X). Verify that you can log in using the system's default username and password. The picture below shows the mRemoteNG Connections item I used for my test. Linux/OSX/BSD users should run the command `ssh [Raspberry Pi’s IP address]` to verify SSH connectivity.



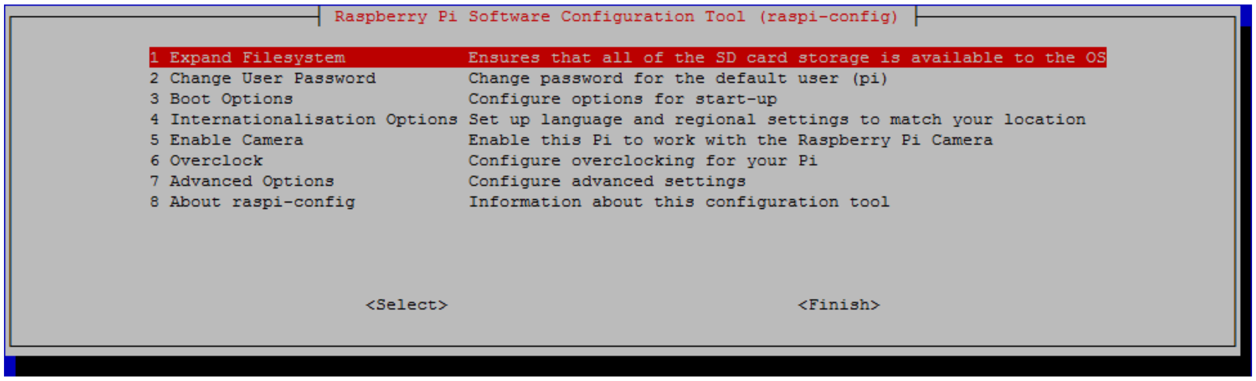
⊟ Display	
Name	rpi
Description	
Icon	mRemoteNG
Panel	General
⊟ Connection	
Hostname/IP	192.168.1.10
Username	pi
Password	●●●●●●●●●
⊟ Protocol	
Protocol	SSH version 2
Port	22
PuTTY Session	Default Settings
⊟ Miscellaneous	
External Tool Befor	
External Tool After	
MAC Address	
User Field	



```
Using username "pi".

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jan  5 18:08:33 2017 from new-host-3.blindseeker.com

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~ $
```

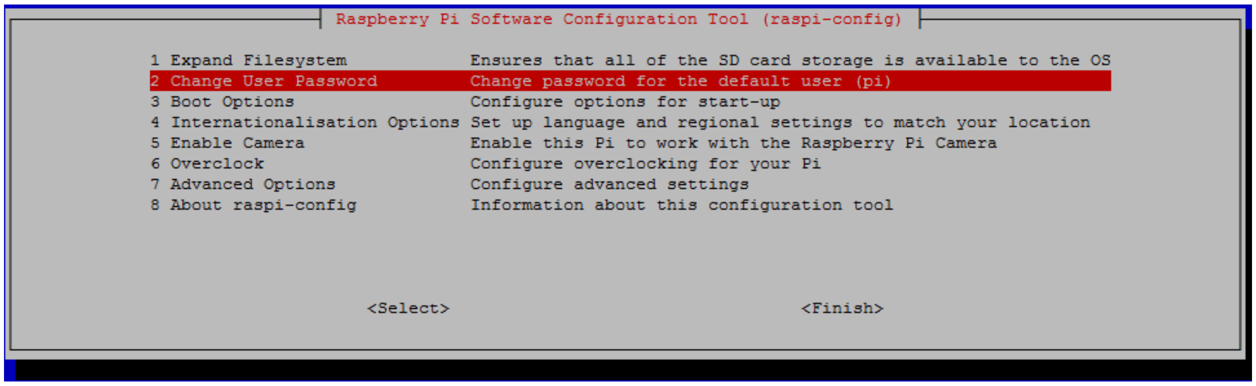
Note: If you are able to login successfully via SSH, the system will inform you that you have NOT changed the default password for the pi user yet. We will be fixing this momentarily.

Now that you have remote access to your Raspberry Pi, there are some things that need to be taken care of before moving on. To do so, open the Raspberry Pi Software Configuration Tool again by running the command `sudo raspi-config`. In the main menu, as depicted below, select option *1 Expand Filesystem*. This will expand the Raspbian image to utilize the full capacity of the SD card in order help prevent the system from running out of disk space. On selecting this option, the system informs you that the partition will be re-sized on the next reboot.



```
┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

1 Expand Filesystem              Ensures that all of the SD card storage is available to the OS
2 Change User Password           Change password for the default user (pi)
3 Boot Options                   Configure options for start-up
4 Internationalisation Options   Set up language and regional settings to match your location
5 Enable Camera                  Enable this Pi to work with the Raspberry Pi Camera
6 Overclock                      Configure overclocking for your Pi
7 Advanced Options               Configure advanced settings
8 About raspi-config             Information about this configuration tool

              <Select>                                  <Finish>
```

Continue by selecting option 2 (Change User Password) from the main menu, and change the pi user’s password to a more secure one. Even though we are going to password authentication via SSH, it is good security hygiene to never leave default credentials enabled on any system you use.

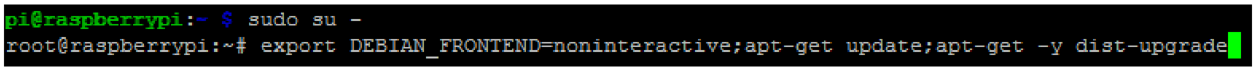


```
┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

1 Expand Filesystem              Ensures that all of the SD card storage is available to the OS
2 Change User Password           Change password for the default user (pi)
3 Boot Options                   Configure options for start-up
4 Internationalisation Options   Set up language and regional settings to match your location
5 Enable Camera                  Enable this Pi to work with the Raspberry Pi Camera
6 Overclock                      Configure overclocking for your Pi
7 Advanced Options               Configure advanced settings
8 About raspi-config             Information about this configuration tool

              <Select>                                  <Finish>
```

When you are done, back in the Raspberry Pi Software Configuration Tool’s main menu, hit ESC or select *Finish* to exit. The system will ask if you want to reboot now. Select *Yes*, then give the RPi some time to go through the reboot. Afterwards, reconnect to your Raspberry Pi via SSH. Remember to use the new password you assigned to the `pi` user.

Note: It is **very** important that you set a complex password for the `pi` user. The `pi` user has the ability to use `sudo` and run commands as root without a password. Take extra care to secure this account’s password, and ensure that you run through the section on disabling password authentication for SSH.

Once you have logged back into the RPi over SSH, you’ll want to update the system and its packages by running the command line `export DEBIAN_FRONTEND=noninteractive; apt-get update; apt-get -y dist-upgrade`. I also highly recommend utilizing the `updater.sh` script from the Automated Patching for Linux Lab VMs section for ensuring that your RPi jump box continues to stay up to date with the latest patches and security updates. At this point, you can safely remove the monitor and keyboard from the RPi, thus making your jump box “headless”.

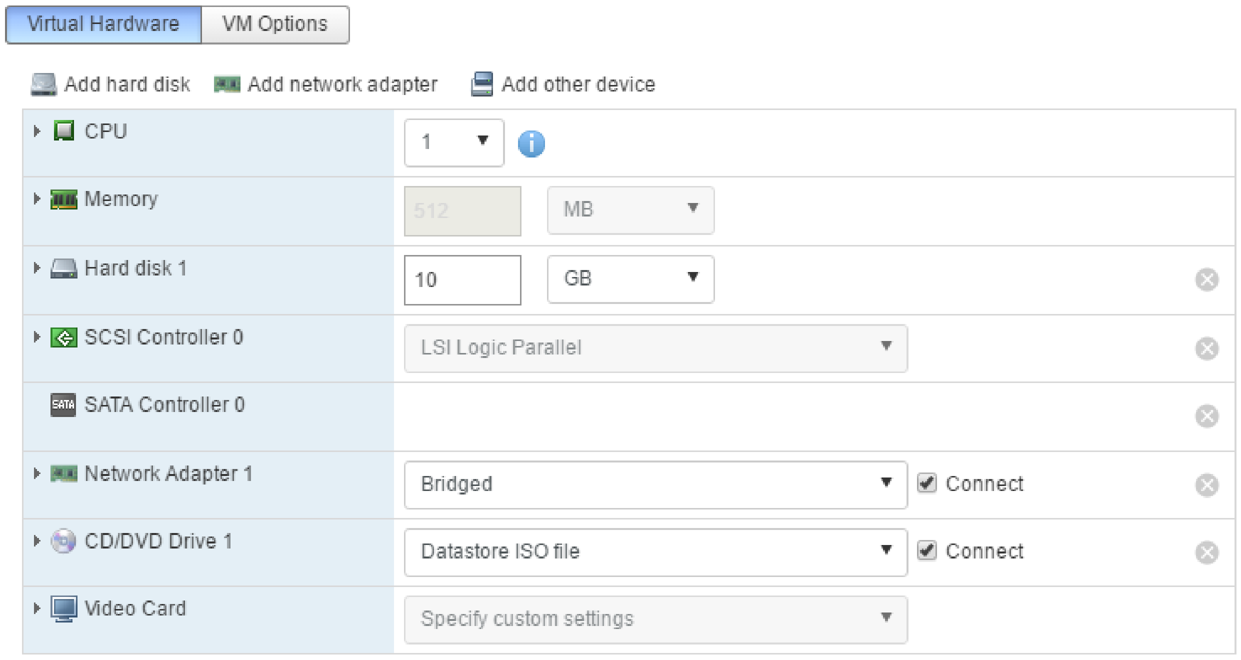


```
pi@raspberrypi:~ $ sudo su -
root@raspberrypi:~# export DEBIAN_FRONTEND=noninteractive;apt-get update;apt-get -y dist-upgrade
```

After you have finished updating your RPi system, review the Setting Up Your Jump Box section for the next steps.

Creating a Jump Box VM

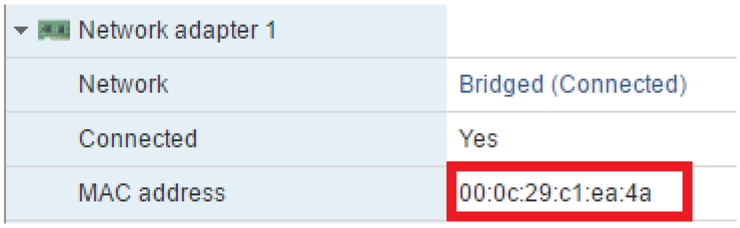
As I explained in the section on using a Raspberry Pi as a jump box, using dedicated physical hardware holds some advantages. However, in the event that you simply do not have a Raspberry Pi (or other, physical hardware) available, running a VM to serve as your jump box is also an option. Start by creating a small Ubuntu Linux 64-bit VM on your ESXi server. Assign it 512mb of RAM, 10GB of hard drive space (thick provisioned, attached to SATA Controller 0 (0:0)), and a single virtual network card connected to the Bridged network. The picture below shows the suggested virtual hardware setup:



CPU	1	
Memory	512	MB
Hard disk 1	10	GB
SCSI Controller 0	LSI Logic Parallel	
SATA Controller 0		
Network Adapter 1	Bridged	✓ Connect
CD/DVD Drive 1	Datastore ISO file	✓ Connect
Video Card	Specify custom settings	

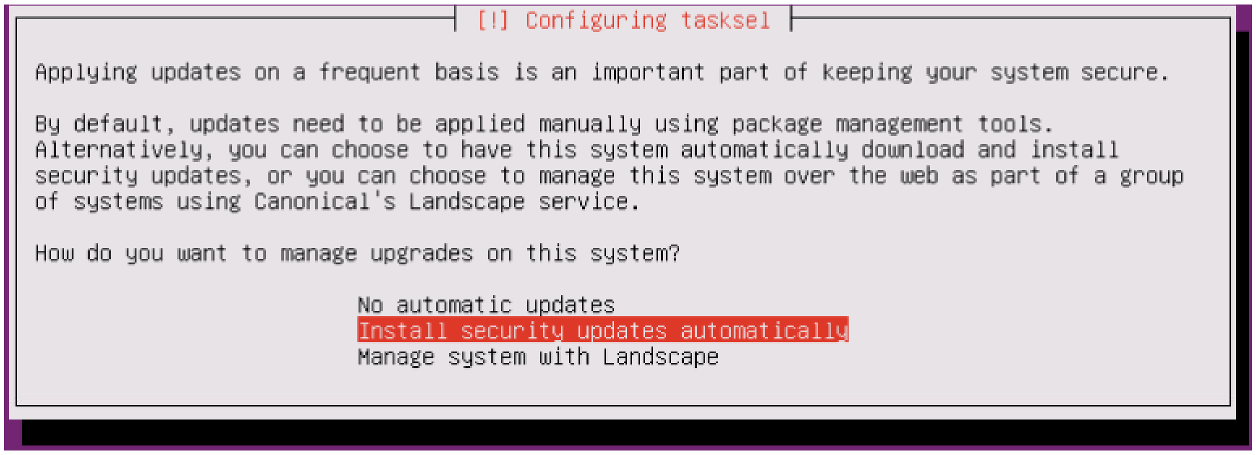
Just like for the majority of our virtual lab's Linux systems, install Ubuntu Server 16.04. Use the ISO you previously uploaded to the datastore.

While the installer is running, check the hardware configuration of your VM from the ESXi web interface, and record the MAC address of your VM, if you plan on configuring a static DHCP address allocation for your jump box.

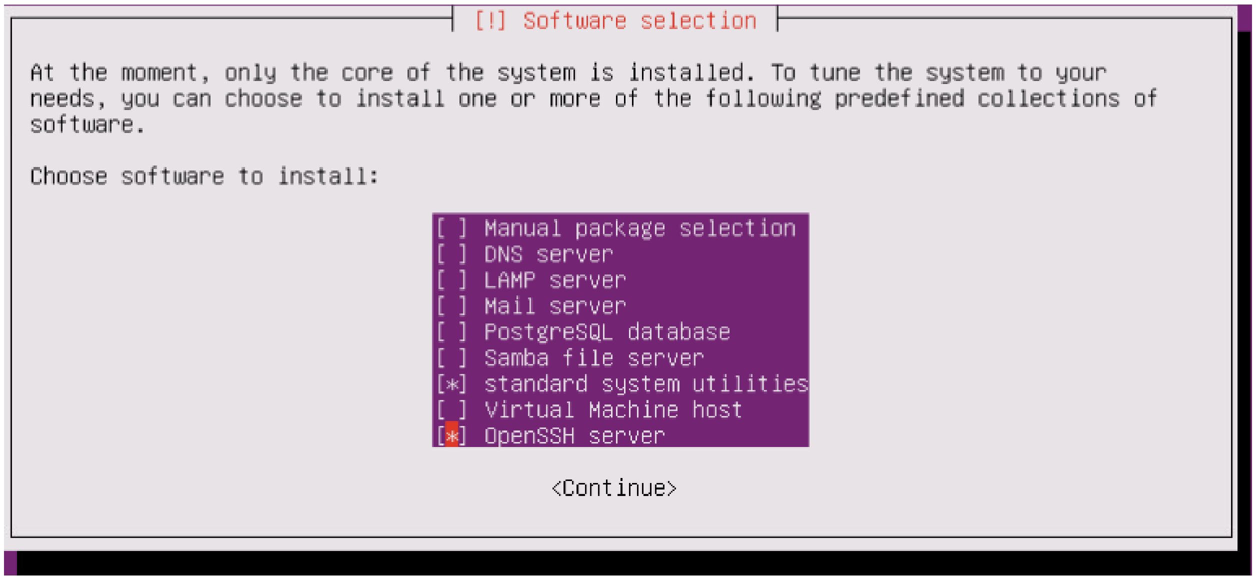


Network adapter 1	
Network	Bridged (Connected)
Connected	Yes
MAC address	00:0c:29:c1:ea:4a

Follow the same OS installation instructions as you did for the other Ubuntu systems on your ESXi host. Make sure to configure the host to receive security updates automatically, and that the *OpenSSH server*, and the *standard system utilities* packages, are installed.

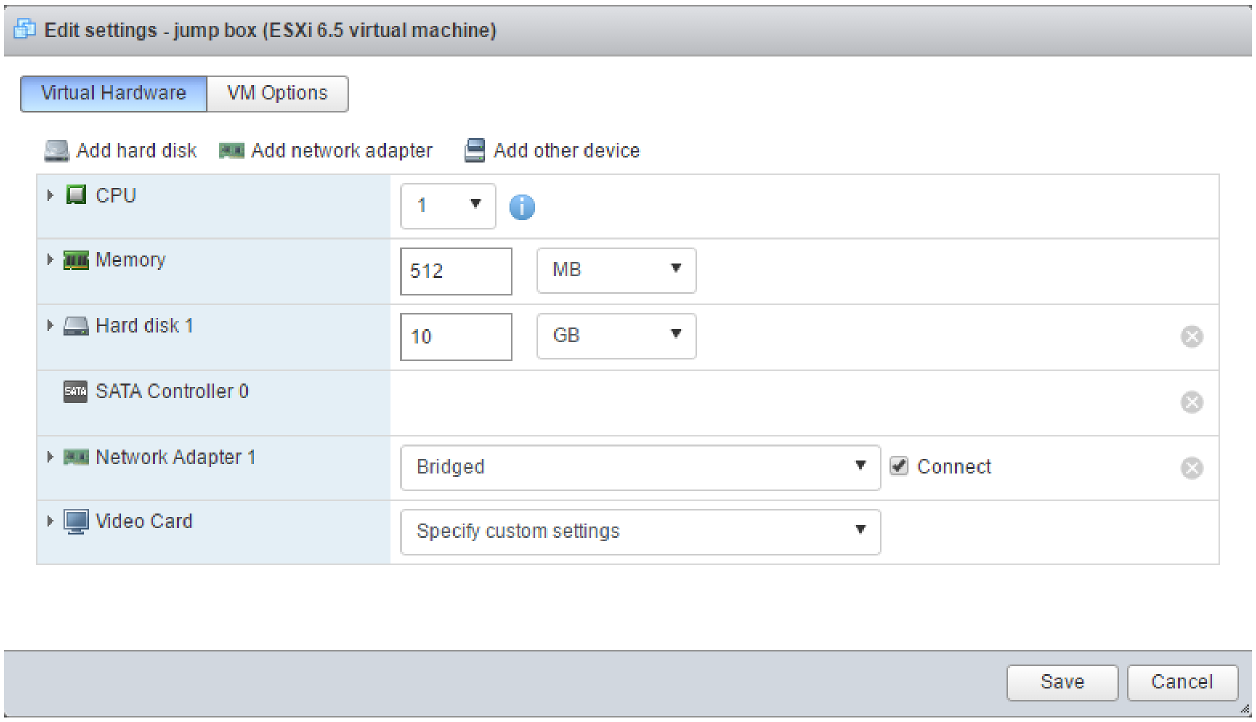


```
[!] Configuring tasksel

Applying updates on a frequent basis is an important part of keeping your system secure.

By default, updates need to be applied manually using package management tools.
Alternatively, you can choose to have this system automatically download and install
security updates, or you can choose to manage this system over the web as part of a group
of systems using Canonical's Landscape service.

How do you want to manage upgrades on this system?

                         No automatic updates
                         Install security updates automatically
                         Manage system with Landscape
```



```
[!] Software selection

At the moment, only the core of the system is installed. To tune the system to your
needs, you can choose to install one or more of the following predefined collections of
software.

Choose software to install:

                               [ ] Manual package selection
                               [ ] DNS server
                               [ ] LAMP server
                               [ ] Mail server
                               [ ] PostgreSQL database
                               [ ] Samba file server
                               [*] standard system utilities
                               [ ] Virtual Machine host
                               [*] OpenSSH server

                                     <Continue>
```

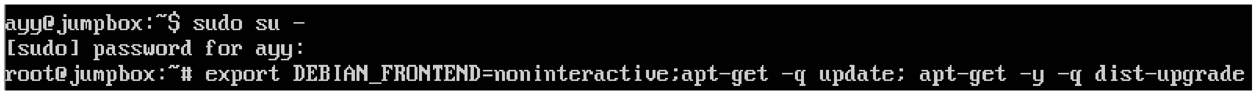
After the installation is complete, power down the VM, and edit the hardware settings. Make sure to remove the USB controller, CD/DVD drive, and the SCSI controller.



Edit settings - jump box (ESXi 6.5 virtual machine)

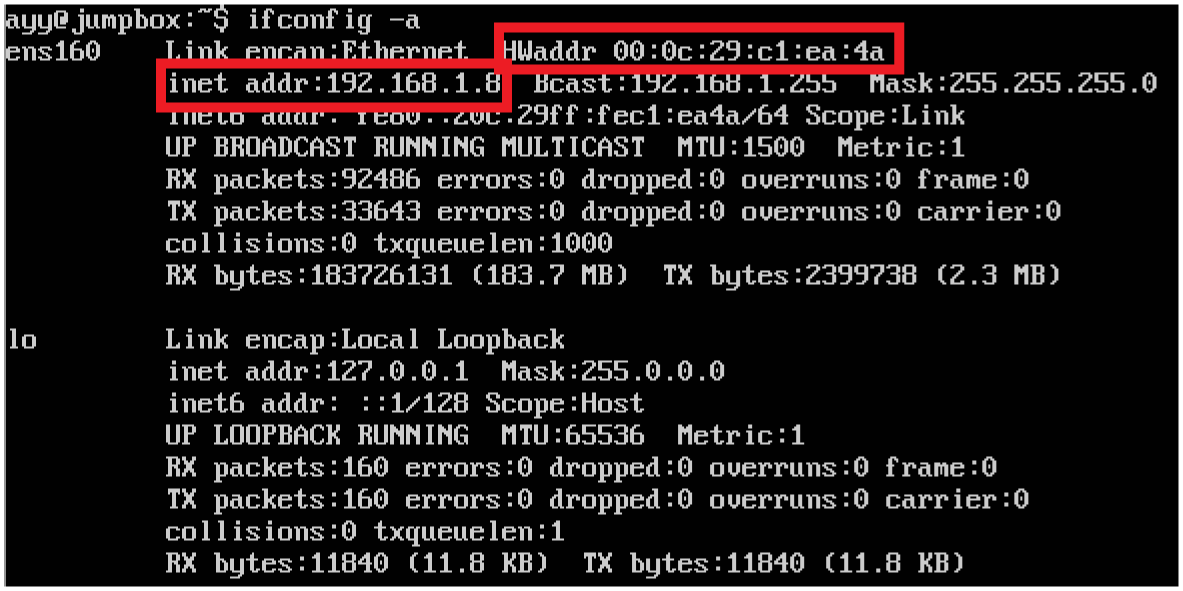
CPU: 1
Memory: 512 MB
Hard disk 1: 10 GB
SATA Controller 0
Network Adapter 1: Bridged, Connect ✓
Video Card: Specify custom settings

Power the VM back on, login to your new VM, and download the latest updates for it by executing the command `sudo su -` to become root, followed by running `export DEBIAN_FRONTEND=noninteractive;apt-get -q update;apt-get -y -q dist-upgrade`.



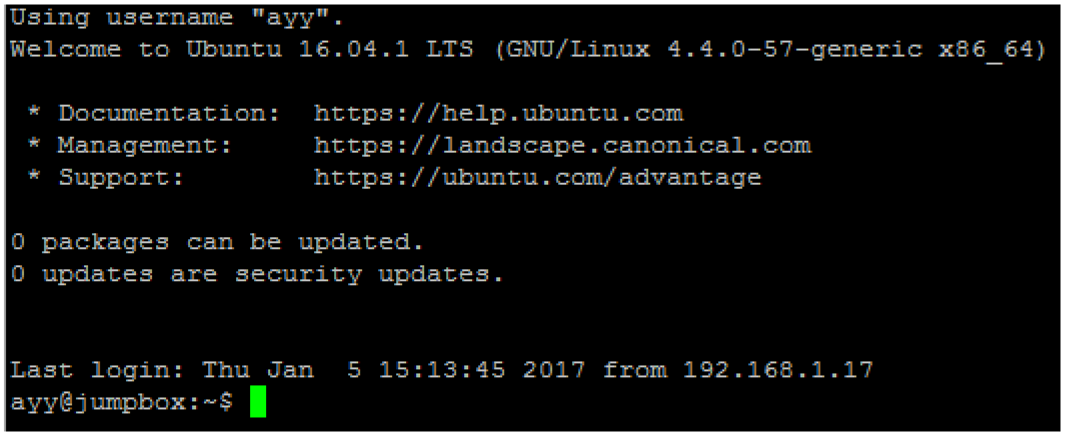
```
ayy@jumpbox:~$ sudo su -
[sudo] password for ayy:
root@jumpbox:~# export DEBIAN_FRONTEND=noninteractive;apt-get -q update; apt-get -y -q dist-upgrade
```

As soon as your system is up to date, run `ifconfig -a` to display the IP address (labeled `inet addr`) that was assigned to your system, as well as to confirm your system's MAC address (tagged `HWaddr`). In the image below, showing the console output of my system, you can see that the VM has an IP address of 192.168.1.8, and a MAC address of 00:0c:29:c1:ea:4a. If you did not record the MAC address of your system earlier, and used it to create a static DHCP mapping for this VM's network interface on your physical network, I would advise doing so now. Revisit the section Dealing with DHCP on instructions on how to setup static DHCP address allocations, or as an alternative, configure a static IP address for this interface.



```
ayy@jumpbox:~$ ifconfig -a
ens160    Link encap:Ethernet  HWaddr 00:0c:29:c1:ea:4a
          inet addr:192.168.1.8  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fec1:ea4a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:92486 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33643 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:183726131 (183.7 MB)  TX bytes:2399738 (2.3 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)
```

Afterwards, confirm that you are able to establish an SSH connection to the VM using your SSH client of choice (e.g. Windows mRemoteNG if you are using Windows, or the `ssh` command in case you are on Linux, BSD, or OS X). Verify that you can log in with the username and password you created during the OS installation.



```
Using username "ayy".
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-57-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Thu Jan  5 15:13:45 2017 from 192.168.1.17
ayy@jumpbox:~$
```

While you’re here, you may also want to implement the `updater.sh` script mentioned in the Automated Patching for Linux Lab VMs section to ensure your jump box is always up to date with the latest patches. After you are finished, complete the Preparing Your Jump Box for Service section.

Other Physical Jump Boxes

If you prefer a physical machine over a VM as a jump box but don't have a Raspberry Pi available, you can easily substitute the latter with any spare hardware you have at hand. Follow along with instructions from the Creating a Jump Box VM section (ignoring the ESXi configuration portions), and simply install Ubuntu Server, with the prescribed installation packages, on your physical jump box.

The major differences to note are that you need to physically plug in your jump box to the same physical network the ESXi server and pfSense VM's *WAN* interface reside on, and that you will need to wait until the operating system is installed to check the MAC address of your system for making a static DHCP address allocation. Other than that, the process is more or less identical.

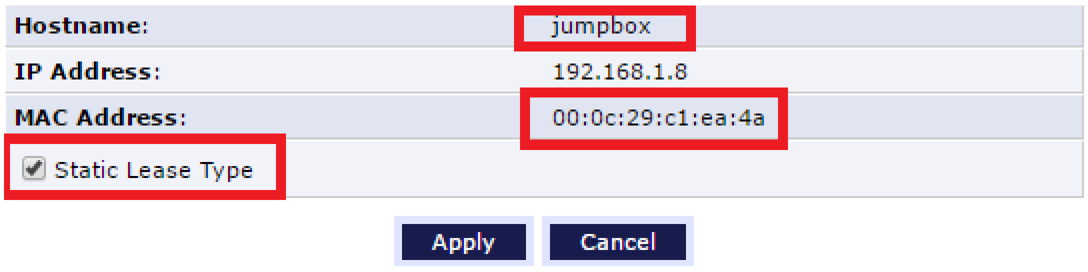
Preparing Your Jump Box for Service

This section instructs you on how to finish setting up your jump box, and how to configure it as well as the management workstation to enable access to your VM lab. This portion of the guide borrows very heavily from other material in this book, namely a vast majority of the Remote Lab Management guide. For example, setting up SSH on Windows or Linux/OS X/BSD management workstations (including recommended applications, generating SSH public/private key pairs, copying the public keys to other systems for key-based authentication, and finally, disabling password authentication as a safety measure to prevent unauthorized access), and setting up persistent static routes on the jump box. The following passages also cover setting up static DHCP address allocations, creating firewall rules to allow your jump box to access the lab VMs, and finally, how to the jump box to forward network traffic to the lab VMs via SSH Tunneling (also known as TCP Forwarding).

Configuring Static DHCP Address Allocations

I highly recommend setting up your jump box to utilize a static DHCP address allocation, if at all possible, as opposed to manual configuration. Which options you have available for implementing static DHCP address allocations on your network depends on where you set up your lab (e.g. home vs. work), and the network equipment you're using, allowed to use and able/allowed to access/configure.

In my case, the router on my home network allows me to configure static DHCP address allocations via an option called "IP Address Distribution". Among other things, this option lets me display current DHCP leases, and change the type of lease from dynamic to static allocations easily, as you can see in the image below.



Hostname:	jumpbox
IP Address:	192.168.1.8
MAC Address:	00:0c:29:c1:ea:4a
✓ Static Lease Type	

I was able to confirm that my host, named "jumpbox", with the MAC address I recorded earlier was assigned the IP address 192.168.1.8. All I had to do to turn this entry into a static DHCP address allocation was check the *Static Lease Type* checkbox and click *Apply*. Depending on the system serving DHCP on your home or office network, configuring static DHCP allocations may be just as simple, or more complicated.

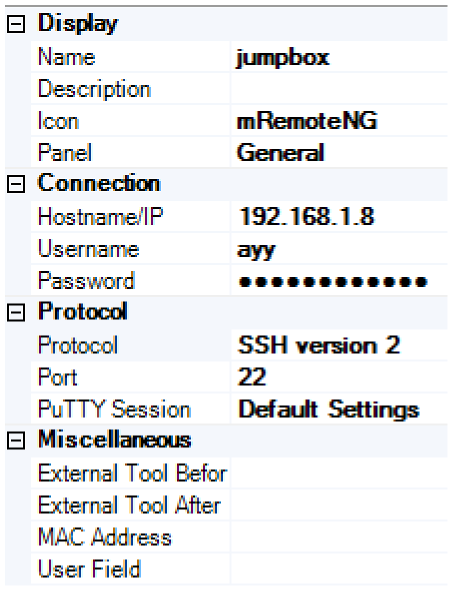
Enabling Key-Based Authentication for your Jump Box

The next step is to configure this system for remote access over SSH, specifically, copying your public SSH key to the host, and disabling password authentication on the jump box via `/etc/ssh/sshd_config`. I will go over doing this from both Windows and Linux/OSX/BSD hosts.

Windows

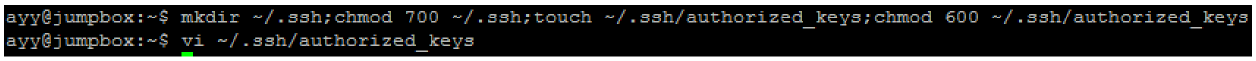
If you intend to connect to your jump box using a management workstation that is running Windows, follow the Windows Remote Access guide, skipping over the section on persistent static routes. Specifically, take care of the following:

1. You should have already used mRemoteNG to log in via the user you set up during the OS installation for your jump box. If you have not, download and install mRemoteNG, create a Connections item (see picture below for reference), and establish an SSH connection to the jump box.



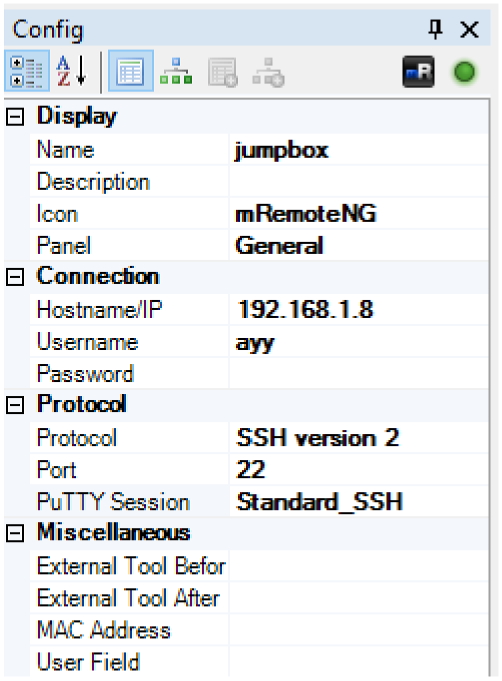
⊟ **Display**	
Name	jumpbox
Description	
Icon	mRemoteNG
Panel	General
⊟ **Connection**	
Hostname/IP	192.168.1.8
Username	ayy
Password	●●●●●●●●●●●●●
⊟ **Protocol**	
Protocol	SSH version 2
Port	22
PuTTY Session	Default Settings
⊟ **Miscellaneous**	
External Tool Befor	
External Tool After	
MAC Address	
User Field	

2. You should already have an SSH public/private key pair generated via PuTTYgen. If that is not the case, follow the corresponding section of the Windows Remote Access guide to do so now.. Add the SSH public key to the `~/.ssh/authorized_keys` file on the jump box.

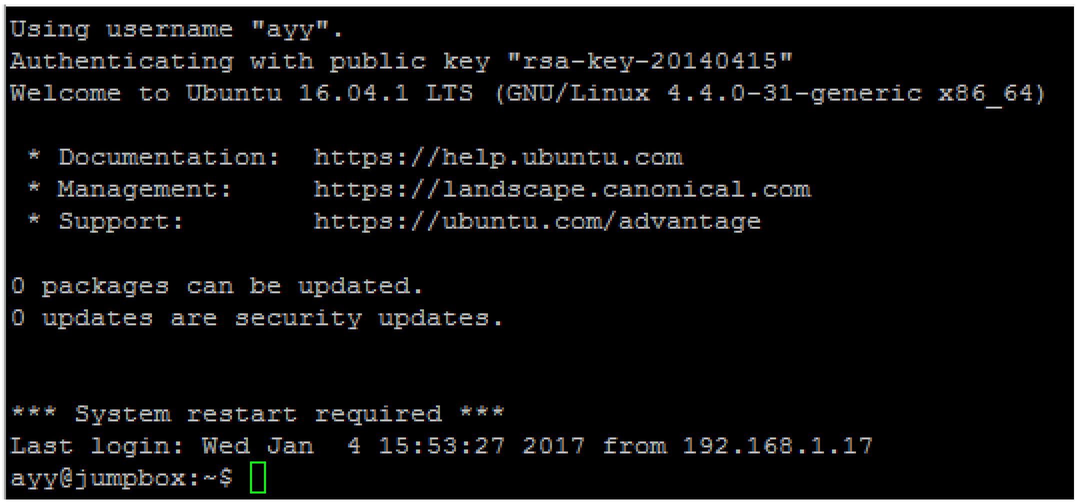


```
ayy@jumpbox:~$ mkdir ~/.ssh;chmod 700 ~/.ssh;touch ~/.ssh/authorized_keys;chmod 600 ~/.ssh/authorized_keys
ayy@jumpbox:~$ vi ~/.ssh/authorized_keys
```

3. Modify your PuTTY Session and mRemoteNG connection profile to log in to the jump box with your SSH key, and verify that key-based authentication is working (make sure to remove the password from your mRemoteNG connection profile). If you need assistance with this, the Windows Remote Access guide has detailed instructions for creating a new PuTTY session that utilizes your SSH key.



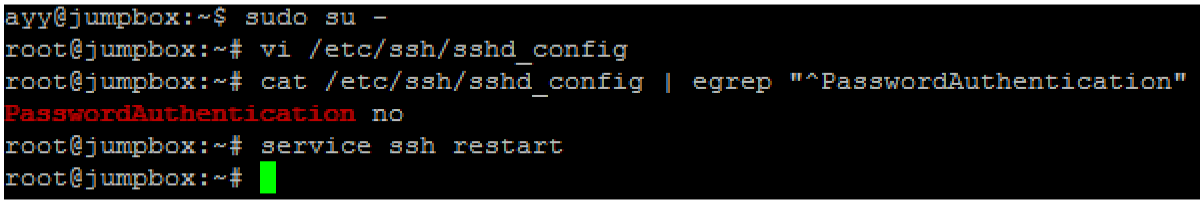
Display	
Name	jumpbox
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	192.168.1.8
Username	ayy
Password	
Protocol	
Protocol	SSH version 2
Port	22
PuTTY Session	Standard_SSH
Miscellaneous	
External Tool Befor	
External Tool After	
MAC Address	
User Field	



```
Using username "ayy".
Authenticating with public key "rsa-key-20140415"
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-31-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Wed Jan  4 15:53:27 2017 from 192.168.1.17
ayy@jumpbox:~$
```

4. I **highly** recommend that you disable password authentication for SSH connections on your jump box entirely due to its exposure to the physical network. If you need guidance

on how to do this, refer to the Disabling Password Authentication Entirely via sshd_config section of the Remote Lab Management guide. Become the root user (`sudo su -`) and modify `/etc/ssh/sshd_config` to disable password authentication, then restart the SSH service.



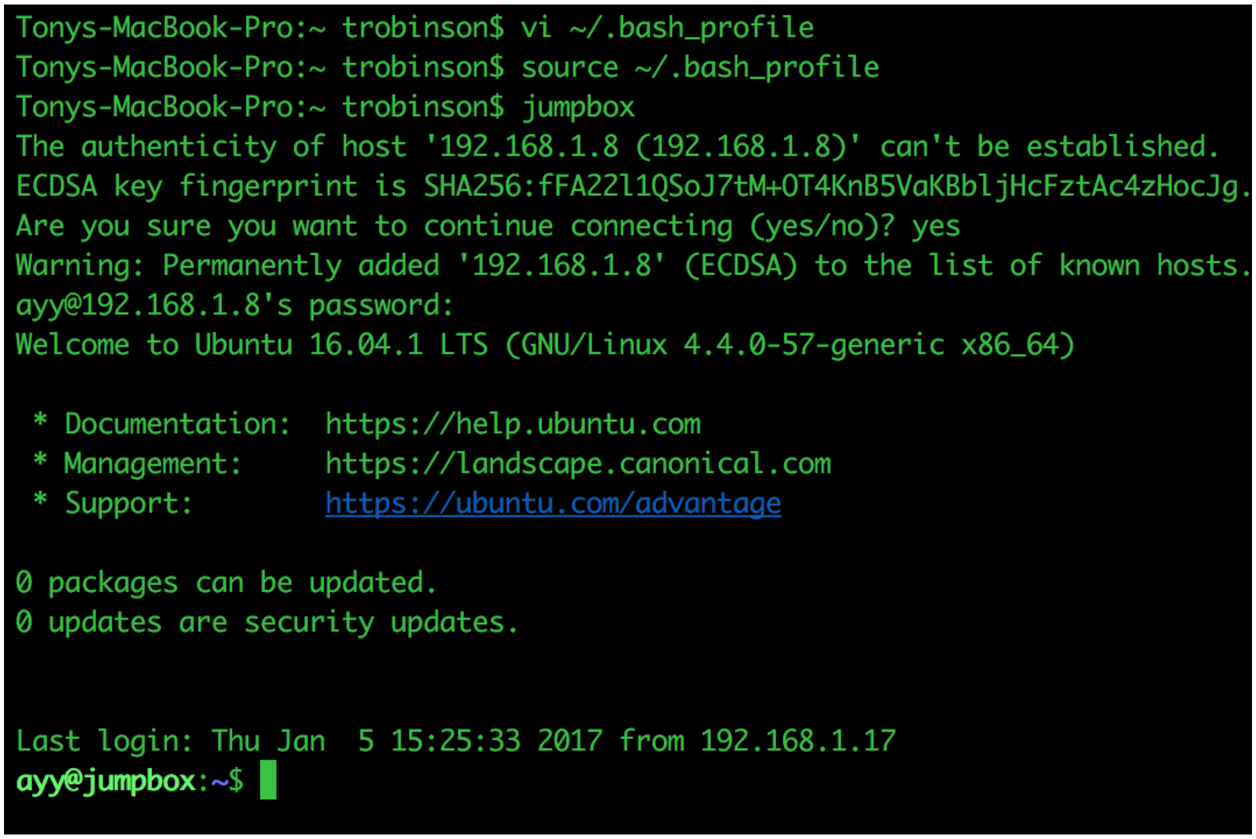
```
ayy@jumpbox:~$ sudo su -
root@jumpbox:~# vi /etc/ssh/sshd_config
root@jumpbox:~# cat /etc/ssh/sshd_config | egrep "^PasswordAuthentication"
PasswordAuthentication no
root@jumpbox:~# service ssh restart
root@jumpbox:~#
```

At this point, you should be able to connect to the jump box from your Windows workstation via SSH with no password required.

Linux/OS X/BSD

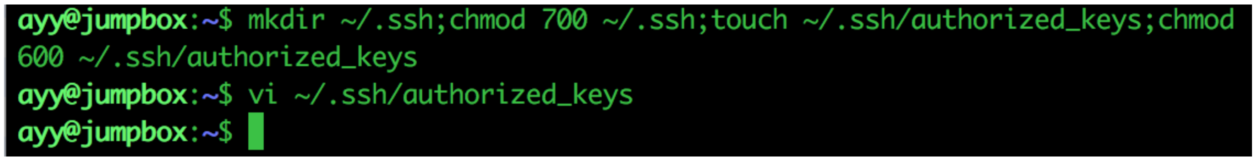
If you plan to connect to your jump box via Linux, OS X, or BSD as your management workstation you will use to connect to your jump box, follow the Remote Access for Linux and OS X guide, skipping over the sections on persistent static routes. Specifically, take care of the following:

1. You should have already used `ssh` to log in via the user you set up during the OS installation for your jump box. If you haven't already, I suggest creating an alias for establishing SSH connections to your jump box (e.g. `echo “alias jumpbox=’ssh ayy@192.168.1.8’” >> ~/.bash_profile; source ~/.bash_profile`). Try to connect to the system with the new alias to make sure it works.

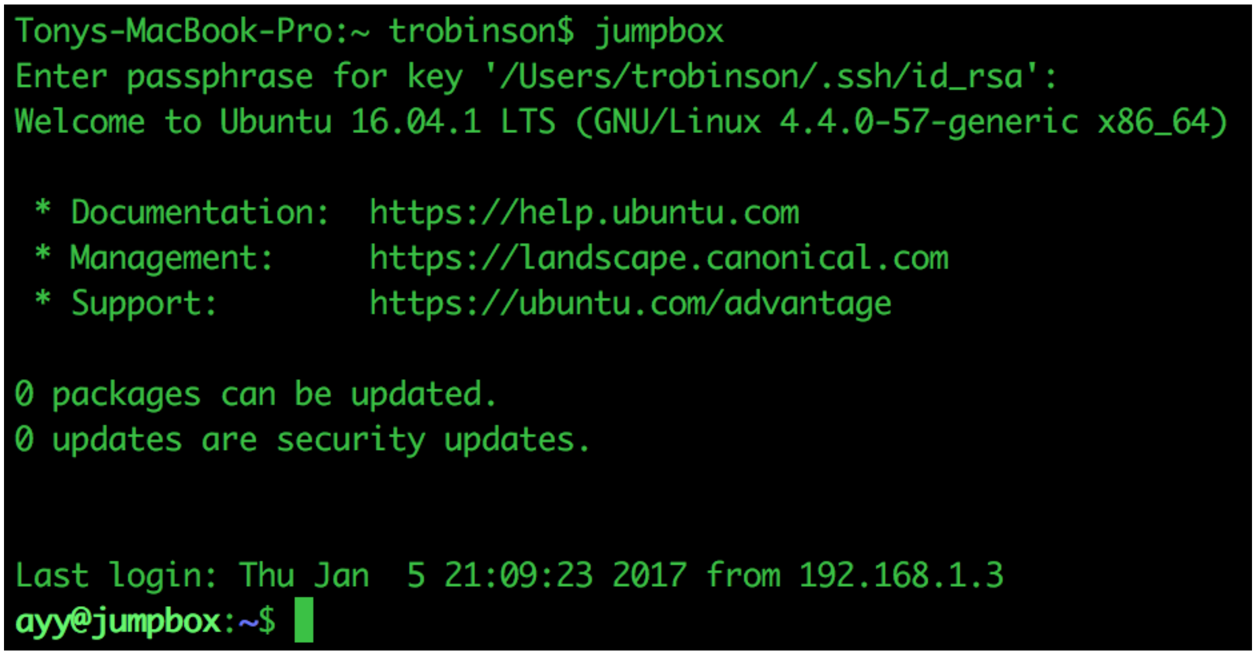


```
Tonys-MacBook-Pro:~ trobinson$ vi ~/.bash_profile
Tonys-MacBook-Pro:~ trobinson$ source ~/.bash_profile
Tonys-MacBook-Pro:~ trobinson$ jumpbox
The authenticity of host '192.168.1.8 (192.168.1.8)' can't be established.
ECDSA key fingerprint is SHA256:fFA22l1QSoJ7tM+OT4KnB5VaKBbljHcFztAc4zHocJg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.8' (ECDSA) to the list of known hosts.
ayy@192.168.1.8's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-57-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Thu Jan  5 15:25:33 2017 from 192.168.1.17
ayy@jumpbox:~$ 
```

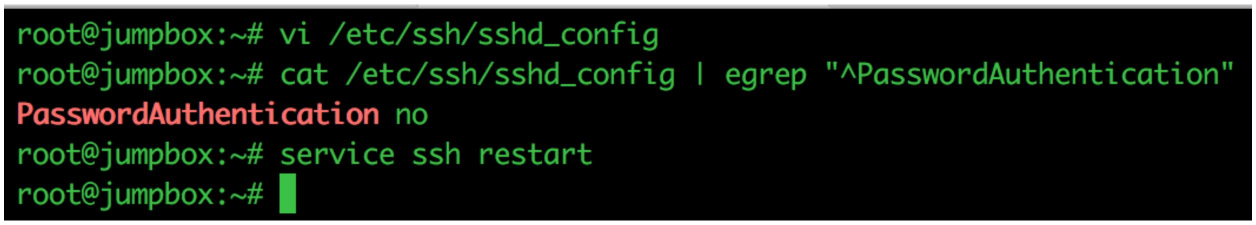
2. Add your SSH public key to the remote user's `~/.ssh/authorized_keys` file on the jump box.


```
ayy@jumpbox:~$ mkdir ~/.ssh;chmod 700 ~/.ssh;touch ~/.ssh/authorized_keys;chmod
600 ~/.ssh/authorized_keys
ayy@jumpbox:~$ vi ~/.ssh/authorized_keys
ayy@jumpbox:~$ 
```

3. Close the current connection, then try to reconnect to the jump box using the alias you made before to see if key-based authentication is working properly.


```
Tonys-MacBook-Pro:~ trobinson$ jumpbox
Enter passphrase for key '/Users/trobinson/.ssh/id_rsa':
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-57-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Thu Jan  5 21:09:23 2017 from 192.168.1.3
ayy@jumpbox:~$ 
```

4. I **highly** recommend that you disable password authentication for SSH connections on your jump box entirely due to its exposure to the physical network. If you need guidance on how to do this, refer to the Disabling Password Authentication Entirely via sshd_config section of the Remote Lab Management guide. Become the root user (`sudo su -`) and modify `/etc/ssh/sshd_config` to disable password authentication, then restart the SSH service.

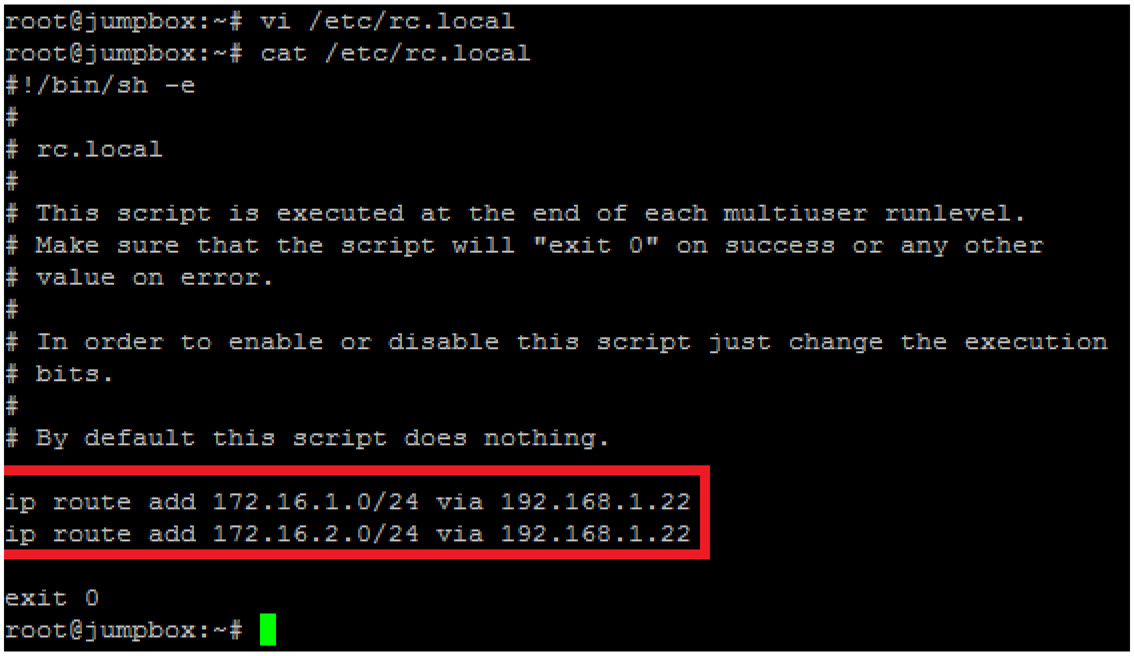

```
root@jumpbox:~# vi /etc/ssh/sshd_config
root@jumpbox:~# cat /etc/ssh/sshd_config | egrep "^PasswordAuthentication"
PasswordAuthentication no
root@jumpbox:~# service ssh restart
root@jumpbox:~# 
```

At this point, you should be able to connect to the jump box from your Linux/OS X/BSD workstation via SSH with no password required.

Adding Static Routes to your Jump Box

After verifying that we can connect to it from our management workstation, we need to set up the jump box to access the VM lab network. This section will be borrowing material from Remote Access for Linux and OS X, since we'll be configuring a Linux system (our jump box system) to access our VM lab networks. Specifically, we'll be using the passages on the `ip` command and `/etc/rc.local` to create persistent static routes on reboot.

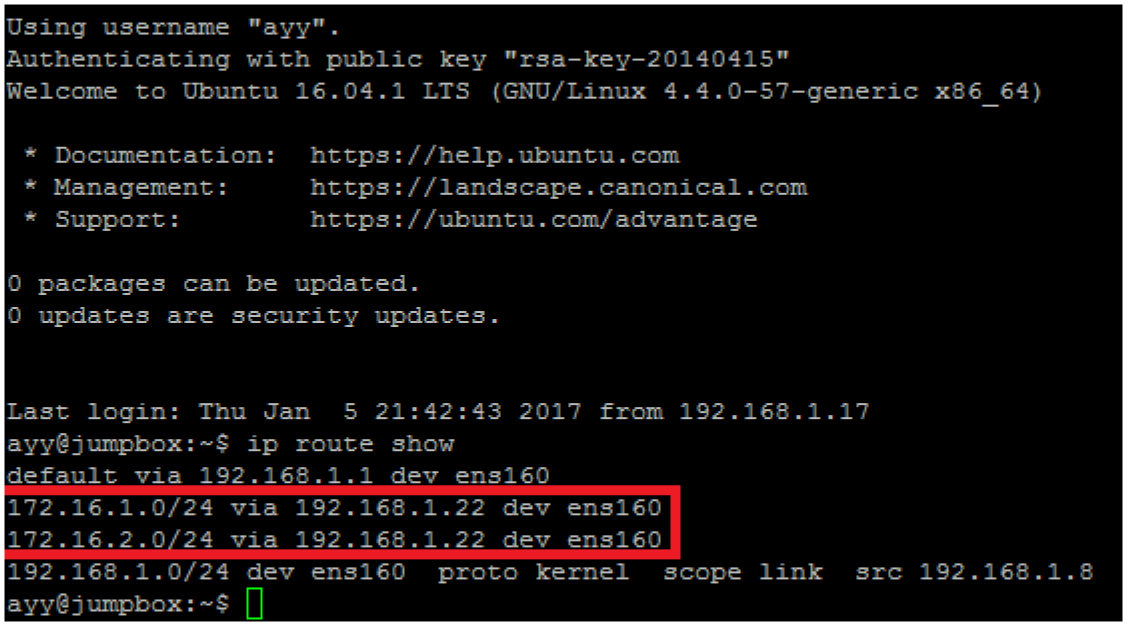
Connect to your jump box over SSH, then enter the `sudo su -` command to become the root user. Using `vi` or the command line text editor of your choice, you need to modify the file `/etc/rc.local`, in particular to add the commands necessary to establish static routes to the 172.16.1.0/24 and 172.16.2.0/24 networks.



```
root@jumpbox:~# vi /etc/rc.local
root@jumpbox:~# cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

ip route add 172.16.1.0/24 via 192.168.1.22
ip route add 172.16.2.0/24 via 192.168.1.22

exit 0
root@jumpbox:~#
```

In the screen capture above, 192.168.1.22 is the IP address of the *WAN* interface of the pfSense virtual machine on my lab network. Adjust that IP address accordingly for your network.

After you have finished editing the `/etc/rc.local` file as the `root` user, reboot your jump box, then reconnect to it over SSH. Next, run the command `ip route show` to display the contents of the routing table.



```
Using username "ayy".
Authenticating with public key "rsa-key-20140415"
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-57-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Thu Jan  5 21:42:43 2017 from 192.168.1.17
ayy@jumpbox:~$ ip route show
default via 192.168.1.1 dev ens160
172.16.1.0/24 via 192.168.1.22 dev ens160
172.16.2.0/24 via 192.168.1.22 dev ens160
192.168.1.0/24 dev ens160  proto kernel  scope link  src 192.168.1.8
ayy@jumpbox:~$
```

As you can see in the image above, `rc.local` has successfully added static routes to the VM lab networks on the jump box system. The output you received should look similar, with only the portion to the right of “via” being different.

Adding Firewall Rules and SSH tunnels to allow access to the VM lab networks

Within the VMware vSphere Hypervisor (ESXi) Setup guide, located in the portion on network configuration for pfSense, there is an initial walk-through on how to setup and maintain access to the pfSense web interface from the WAN interface of the firewall. The respective procedures invoke the `pfctl` and `easyrule` commands, issued from the console of the pfSense VM. We use the `easyrule` command to create a permanent firewall rule to allow our management workstation persistent access to the pfsense webConfigurator UI via the *WAN* interface.

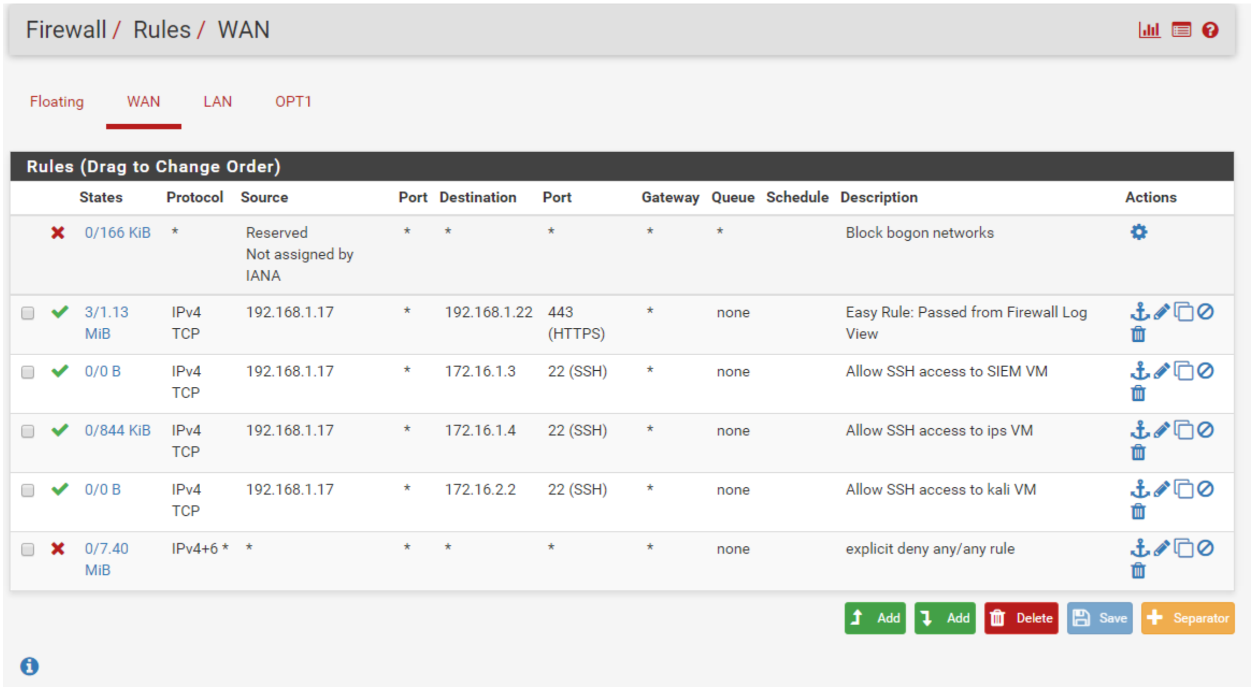
Adding this rule alone is not enough, however. By default, pfsense blocks RFC1918 addresses on the *WAN* interface. This is because usually, firewalls are deployed at the perimeter of a home, or enterprise network and the *WAN* interface is assumed to be connected to a public, routable network (e.g. the internet). RFC1918 address are not publicly routable, and therefore in that case, the default action of dropping traffic from RFC1918 addresses on the *WAN* interface is reasonable. In our case however, the *WAN* interface of our pfSense firewall is connected to an RFC1918 network, as is our management workstation we will be using to manage and configure the firewall (and all of our lab virtual machines). Because this default RFC1918 rule takes precedence over any rules we add via `easyrule` or the web interface, we must also use `pfctl -d` to temporarily disable the firewall in order to login to the webConfigurator and permanently disable the default setting of blocking RFC1918 IP addresses on the *WAN* interface, to let your management workstation connect to the webConfigurator UI.

In addition to the firewall rule above, in the pfSense Firewall Rules and Network Services Guide, there is a section dedicated to Creating Firewall Rules to enable SSH access to your lab VMs from your management workstation (Specifically, Firewall Rule Configuration - Bare-metal Hypervisors). We will simply be applying these firewall rules to our jumpbox system, as opposed to our management workstation.

If you are still able to access the the pfSense webConfigurator UI from your management workstation, you may simply go ahead creating new firewall rules for your jump box system in order to allow it to access the VM lab. Otherwise, if you cannot or are no longer able to reach the web UI, you have to use `pfctl -d` from the console of the pfSense virtual machine first, to temporarily disable the firewall, to enable yourself to set up the rules necessary via the webUI.

I Can Still Access the pfSense webConfigurator with my Management Workstation

Log in to the pfSense webConfigurator, then navigate to *Firewall* > *Rules*, and if it isn't already highlighted, select the *WAN* interface.

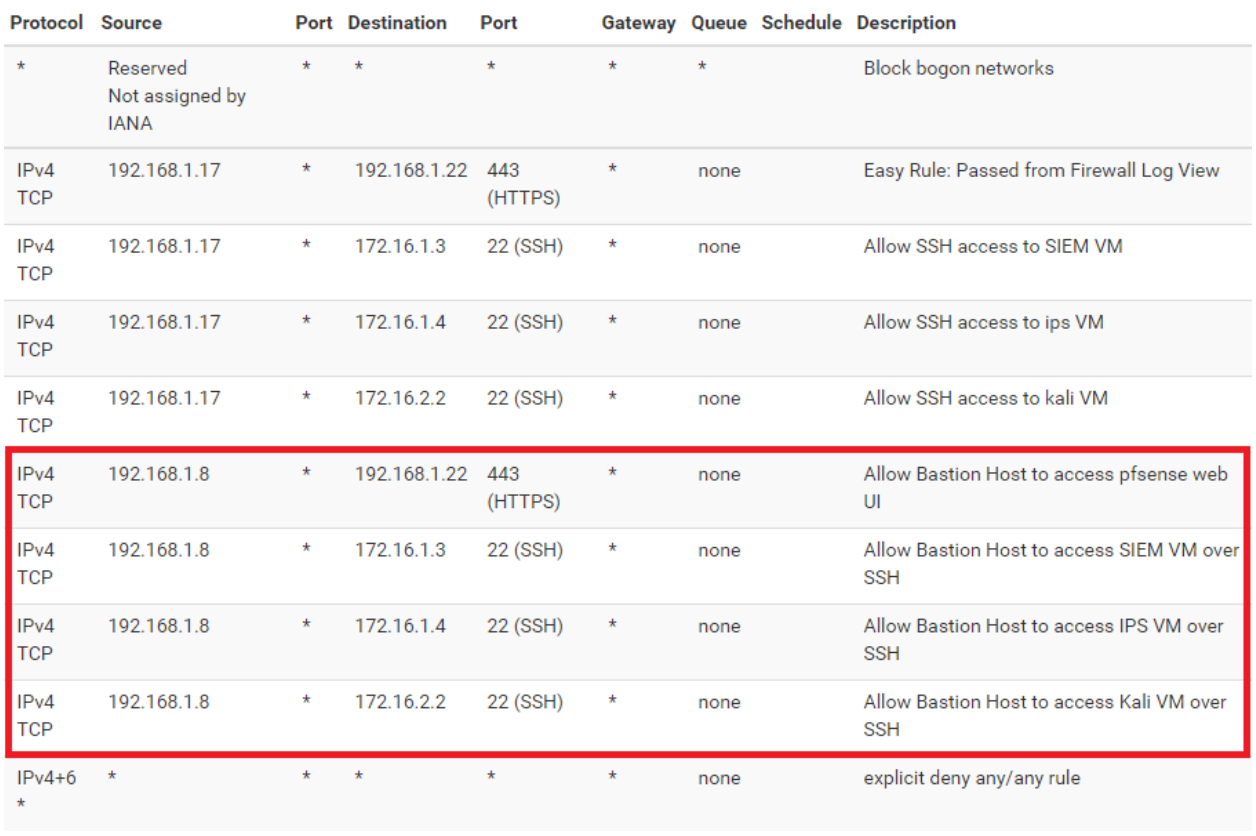


Rules (Drag to Change Order)

		States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
	✗	0/166 KiB	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
☐	✓	3/1.13 MiB	IPv4 TCP	192.168.1.17	*	192.168.1.22	443 (HTTPS)	*	none		Easy Rule: Passed from Firewall Log View	
☐	✓	0/0 B	IPv4 TCP	192.168.1.17	*	172.16.1.3	22 (SSH)	*	none		Allow SSH access to SIEM VM	
☐	✓	0/844 KiB	IPv4 TCP	192.168.1.17	*	172.16.1.4	22 (SSH)	*	none		Allow SSH access to ips VM	
☐	✓	0/0 B	IPv4 TCP	192.168.1.17	*	172.16.2.2	22 (SSH)	*	none		Allow SSH access to kali VM	
☐	✗	0/7.40 MiB	IPv4+6 *	*	*	*	*	*	none		explicit deny any/any rule	

As you can see from the screen capture above, I have a number of rules that allow my Windows workstation (192.168.1.17) access to the pfSense webConfigurator over HTTPS, and several of the lab VMs over port 22 (SSH). We're going to add a new set of rules to allow the jump box system (which in my case is configured with the IP address 192.168.1.8) as well to access those systems.

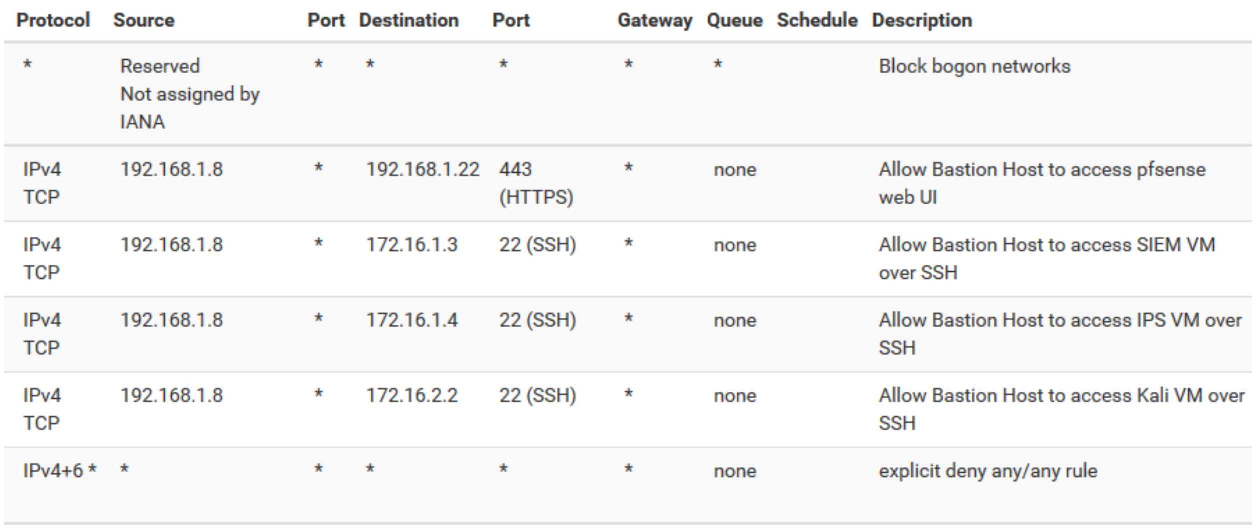
Modify the firewall policy for the *WAN* interface by adding 4 new pass rules: one to allow access the pfSense WAN IP address (e.g. 192.168.1.22 in my case) over port 443/tcp(https), and 3 rules to enable access to the Kali VM (172.16.2.2), the SIEM VM (172.16.1.3), and the IPS VM (172.16.1.4) over port 22/tcp(SSH) respectively. These new rules must be placed ABOVE the explicit deny any/any entry (that is, if you chose to create one. By default pfSense denies any traffic on an interface that is not explicitly allowed). When finished, your list of firewall rules for the WAN interface should look similar to the one shown in the picture below, with the values of the source IP addresses probably differing from the one depicted:



Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks
IPv4 TCP	192.168.1.17	*	192.168.1.22	443 (HTTPS)	*	none		Easy Rule: Passed from Firewall Log View
IPv4 TCP	192.168.1.17	*	172.16.1.3	22 (SSH)	*	none		Allow SSH access to SIEM VM
IPv4 TCP	192.168.1.17	*	172.16.1.4	22 (SSH)	*	none		Allow SSH access to ips VM
IPv4 TCP	192.168.1.17	*	172.16.2.2	22 (SSH)	*	none		Allow SSH access to kali VM
IPv4 TCP	192.168.1.8	*	192.168.1.22	443 (HTTPS)	*	none		Allow Bastion Host to access pfsense web UI
IPv4 TCP	192.168.1.8	*	172.16.1.3	22 (SSH)	*	none		Allow Bastion Host to access SIEM VM over SSH
IPv4 TCP	192.168.1.8	*	172.16.1.4	22 (SSH)	*	none		Allow Bastion Host to access IPS VM over SSH
IPv4 TCP	192.168.1.8	*	172.16.2.2	22 (SSH)	*	none		Allow Bastion Host to access Kali VM over SSH
IPv4+6 *	*	*	*	*	*	none		explicit deny any/any rule

At this point you can log out of the webConfigurator from your management workstation. Don't delete the existing firewall rules for your management workstation yet, and don't take a new snapshot of the pfSense VM at this point, because we need to verify that the extended firewall policy in general, and the new rules in particular, are working as intended. To find out if they do, go to the section TCP Forwarding and You, then Testing your Dynamic Tunnels with FoxyProxy. After having verified that your dynamic tunnel is working and having logged on from the IP address of your jump box, proceed to Testing Your Forward Tunnels.

After having testing connectivity to all of your lab assets using your jump box system, return to the web UI of your pfSense VM. On the page that lists the firewall rules for the WAN interface, use the icon that looks like a trash bin to delete the rows referring the IP address of your management workstation (e.g. in the above illustration, that would apply to all the rules containing the IP address 192.168.1.17). After cleaning up, your final ruleset should look similar to the one displayed in the image below:



Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks
IPv4 TCP	192.168.1.8	*	192.168.1.22	443 (HTTPS)	*	none		Allow Bastion Host to access pfsense web UI
IPv4 TCP	192.168.1.8	*	172.16.1.3	22 (SSH)	*	none		Allow Bastion Host to access SIEM VM over SSH
IPv4 TCP	192.168.1.8	*	172.16.1.4	22 (SSH)	*	none		Allow Bastion Host to access IPS VM over SSH
IPv4 TCP	192.168.1.8	*	172.16.2.2	22 (SSH)	*	none		Allow Bastion Host to access Kali VM over SSH
IPv4+6 *	*	*	*	*	*	none		explicit deny any/any rule

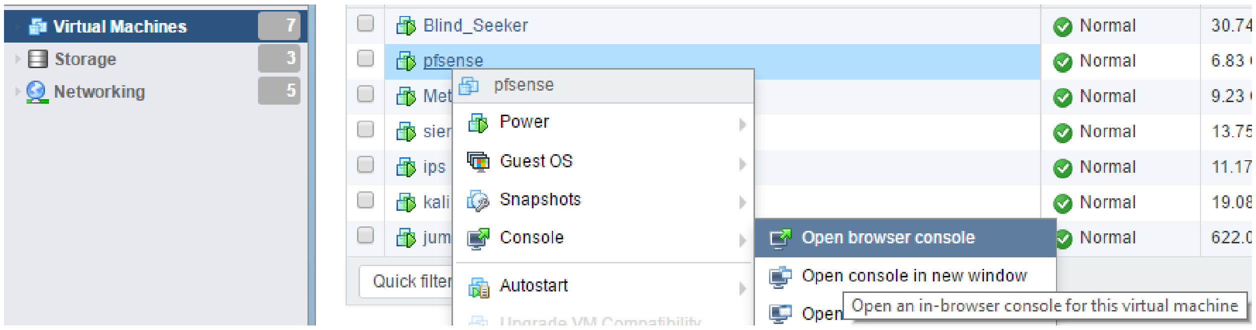
Once you have finished modifying the firewall rules for the WAN interface, save your changes and log out of the webConfigurator.

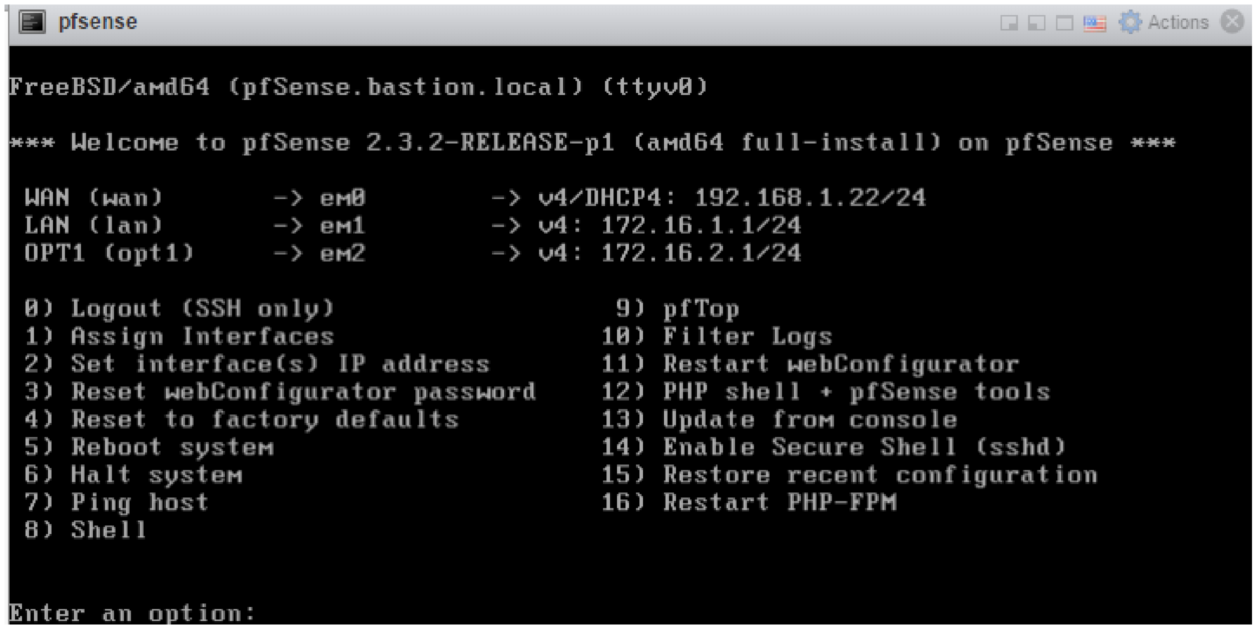
If everything has gone well so far, your jump box should be ready for service.

I Have Lost Access to the pfSense webConfigurator UI

So you no longer have access to the webConfigurator interface, either due to misconfiguration, or due to DHCP giving your workstation a different IP address. As long as you are still able to access the ESXi server via its web interface or by utilizing the VMware vSphere Client from your management workstation, that is not a problem.

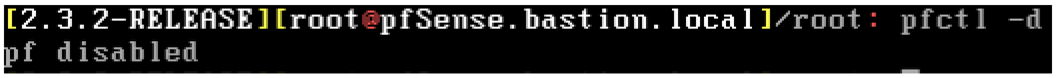
Use one of the two options mentioned above to connect to the ESXi host. (For the sake of simplicity, the following instructions are based on the web UI.) Click on Virtual Machines in the *Navigator* panel on the left side of the browser window, right click on the pfSense VM. From the context menu, choose *Console*, then click *Open browser console*. A console session to the pfSense VM appears in your web browser, just like the one we used to initially configure this system.





```
FreeBSD/amd64 (pfSense.bastion.local) (ttyv0)

*** Welcome to pfSense 2.3.2-RELEASE-p1 (amd64 full-install) on pfSense ***

 WAN (wan)       -> em0       -> v4/DHCP4: 192.168.1.22/24
 LAN (lan)       -> em1       -> v4: 172.16.1.1/24
 OPT1 (opt1)     -> em2       -> v4: 172.16.2.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell


Enter an option:
```

Click inside the console session, then select option 8 to access the shell interface of pfSense. Enter the command `pfctl -d` to completely disable the firewall. This is the fastest and easiest way for us to regain access to the system. Don't worry, we'll re-enable it in just a moment.

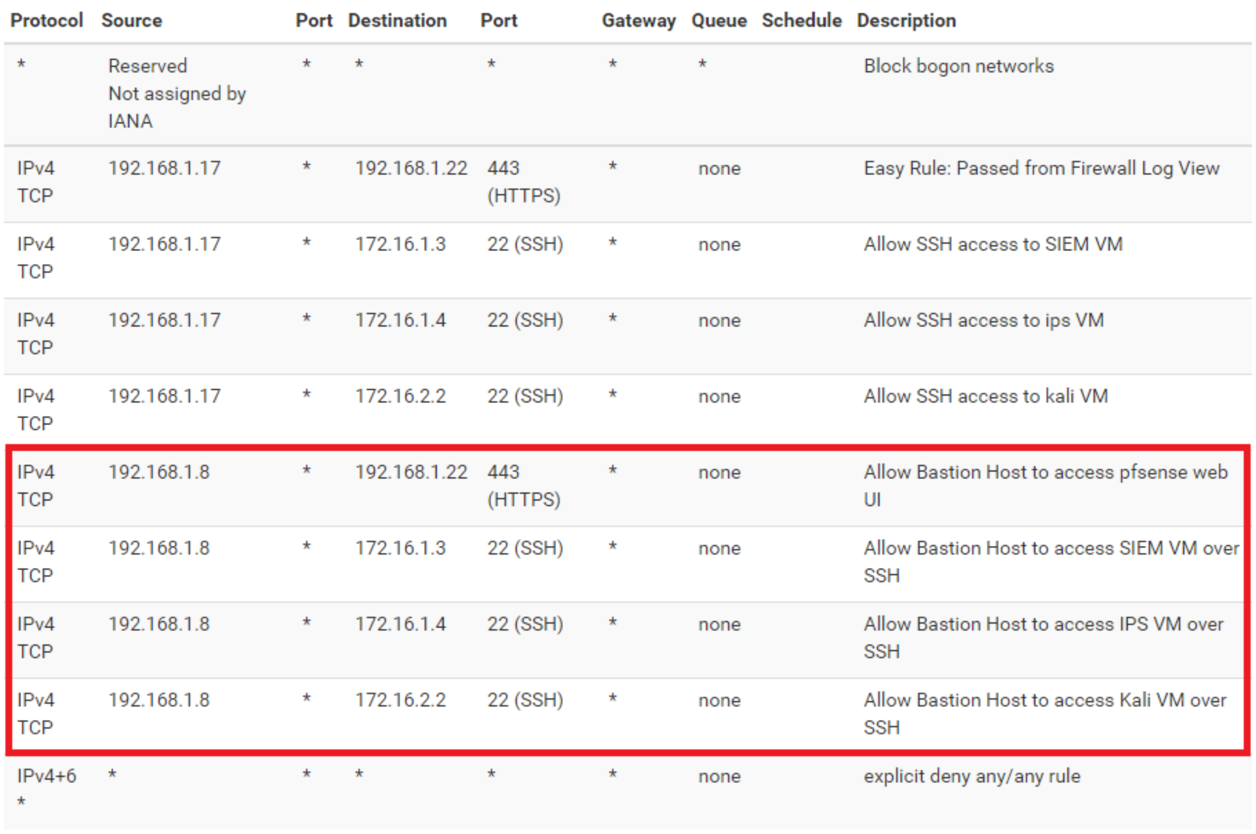


```
[2.3.2-RELEASE][root@pfSense.bastion.local]/root: pfctl -d
pf disabled
```

At this point, the firewall is disabled. We need to establish a dynamic tunnel and proxy off of our jump box to continue. Please proceed to the section TCP Forwarding and You, then Testing your Dynamic Tunnels with FoxyProxy. After working through those passages of the guide, return here to complete the configuration of the firewall.

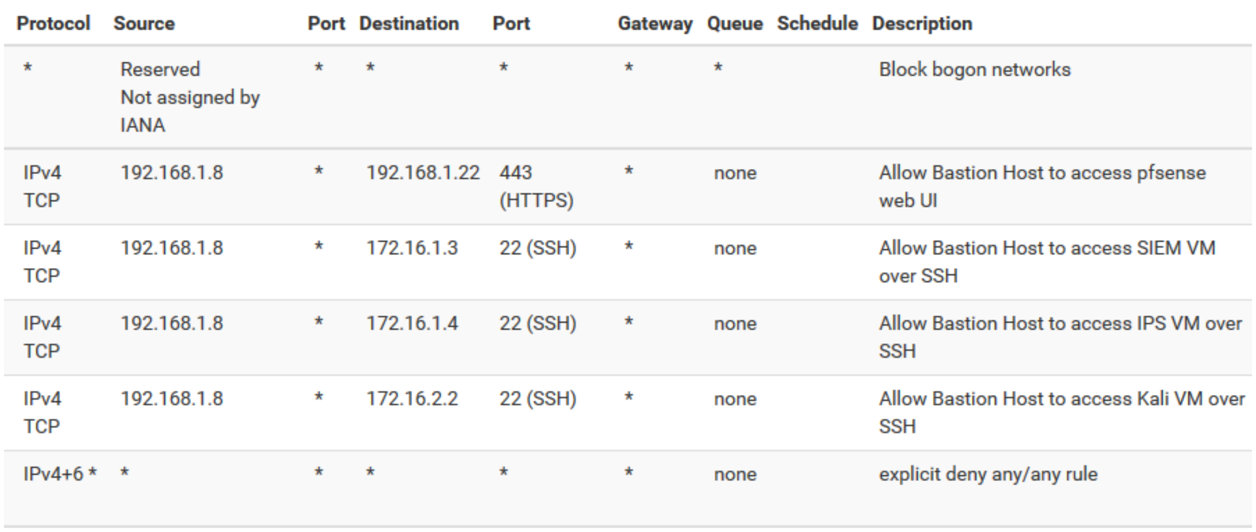
After successfully setting up and testing the TCP forwarding/SSH tunnelling, confirm that you can login to the pfSense webConfigurator from your management workstation via the WAN interface of the pfSense VM. Navigate to *Firewall* > *Rules*, and if it isn't already highlighted, select the *WAN* interface. We're going to add a new set of rules to allow the jump box system (which in my case is configured with the IP address 192.168.1.8) to access the pfSense webConfigurator over HTTPS (443/tcp), and several of the lab VMs over port 22/tcp (SSH).

Modify the firewall rules for the WAN interface by adding 4 new pass rules: one to access the pfSense WAN IP address (e.g. 192.168.1.22 in my case) over port 443/tcp(https), and 3 rules to enable access to the Kali VM (172.16.2.2), the SIEM VM (172.16.1.3), and the IPS VM (172.16.1.4) over port 22/tcp(SSH) respectively. These new rules must be placed ABOVE the explicit deny any/any entry (that is, if you chose to create one, by default pfSense denies any traffic on an interface that is not explicitly allowed). Depending on whether or not you had added firewall rules to allow the (presumably now invalid) IP address of your management workstation access to your lab network, and apart from the actual values of the source IP addresses, your list of rules for the WAN interface should look similar to the one displayed in the image below:



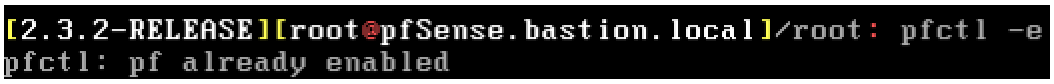
Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks
IPv4 TCP	192.168.1.17	*	192.168.1.22	443 (HTTPS)	*	none		Easy Rule: Passed from Firewall Log View
IPv4 TCP	192.168.1.17	*	172.16.1.3	22 (SSH)	*	none		Allow SSH access to SIEM VM
IPv4 TCP	192.168.1.17	*	172.16.1.4	22 (SSH)	*	none		Allow SSH access to ips VM
IPv4 TCP	192.168.1.17	*	172.16.2.2	22 (SSH)	*	none		Allow SSH access to kali VM
IPv4 TCP	192.168.1.8	*	192.168.1.22	443 (HTTPS)	*	none		Allow Bastion Host to access pfsense web UI
IPv4 TCP	192.168.1.8	*	172.16.1.3	22 (SSH)	*	none		Allow Bastion Host to access SIEM VM over SSH
IPv4 TCP	192.168.1.8	*	172.16.1.4	22 (SSH)	*	none		Allow Bastion Host to access IPS VM over SSH
IPv4 TCP	192.168.1.8	*	172.16.2.2	22 (SSH)	*	none		Allow Bastion Host to access Kali VM over SSH
IPv4+6 *	*	*	*	*	*	none		explicit deny any/any rule

Using the icon that looks like a trash bin, simply delete the rows referring the (old) IP address of your management workstation (e.g. in the above illustration, all the rules containing the IP address 192.168.1.17). After cleaning up, your final ruleset should look similar to the one shown in the picture below:



Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks
IPv4 TCP	192.168.1.8	*	192.168.1.22	443 (HTTPS)	*	none		Allow Bastion Host to access pfsense web UI
IPv4 TCP	192.168.1.8	*	172.16.1.3	22 (SSH)	*	none		Allow Bastion Host to access SIEM VM over SSH
IPv4 TCP	192.168.1.8	*	172.16.1.4	22 (SSH)	*	none		Allow Bastion Host to access IPS VM over SSH
IPv4 TCP	192.168.1.8	*	172.16.2.2	22 (SSH)	*	none		Allow Bastion Host to access Kali VM over SSH
IPv4+6 *	*	*	*	*	*	none		explicit deny any/any rule

Once you have finished modifying the firewall rules for the WAN interface, save your changes and log out of the webConfigurator. On saving the firewall configuration, pfSense should have re-enabled the firewall automatically. However, if you want to be 100% sure about that, log in to the ESXi server via its web interface and open a console session to the pfSense VM. Select option 8 to open a root shell on the system, then run the command `pfctl -e`.



```
[2.3.2-RELEASE][root@pfSense.bastion.local]/root: pfctl -e
pfctl: pf already enabled
```

If you received an output similar to the image above, everything worked as expected and the firewall has already been re-enabled. If the status message on your system looks any different, you almost certainly didn't save the recent firewall policy changes. In that case you probably need to run `pfctl -d` once more, and reconfigure the rules on the WAN interface again.

If everything has gone well so far, proceed to the Testing Your Forward Tunnels section. As soon as the forwarding tunnels are working properly, your jump box should be ready for service.

TCP Forwarding and You

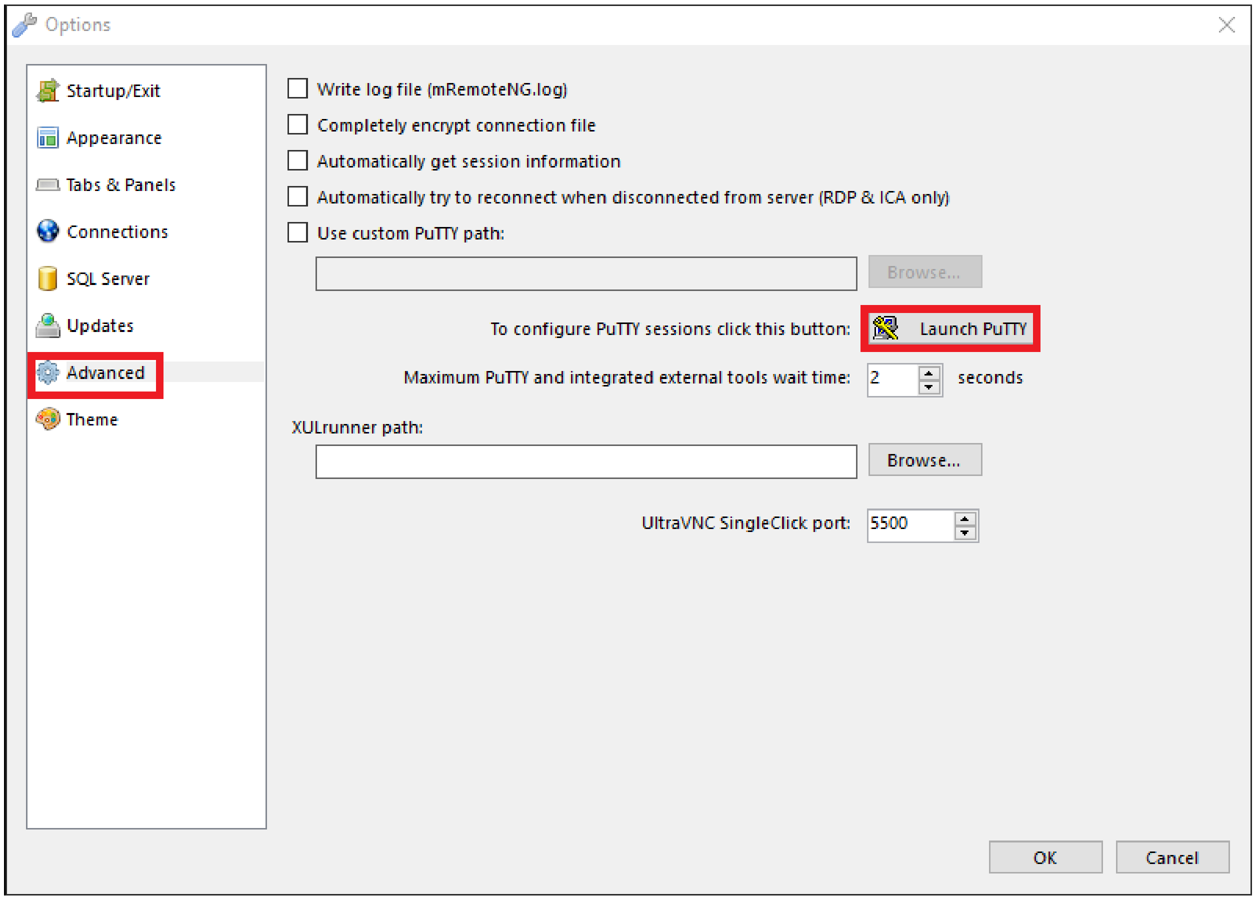
Most SSH servers have an option called "TCP Forwarding" available. A lot of IT professionals also refer to this capability as "SSH Tunneling". The idea is that SSH can be used to forward TCP network connections, not unlike a proxy, resulting in TCP connections to appearing as though they are being sent from the SSH server itself. This functionality is the core of what will enable us to use our jump box to access the systems inside our VM lab.

You need to make some modifications to the SSH sessions you created to access your bastion host earlier. If you are using a management workstation running Windows, review the Windows

SSH Tunnels section. If you are using Linux/OS X/BSD, check the Linux/BSD/OS X SSH Tunnels section instead. These guides will teach you how to create dynamic SSH tunnels for forwarding web traffic through your jump box, and how to forward SSH connections (as well as those based on other protocols) to the lab VMs as well.

Windows SSH Tunnels

If you created mRemoteNG SSH sessions earlier to use key-based authentication, most of the following setup instructions will seem VERY familiar. Start up the mRemoteNG program, then on the menu bar at the top of the screen, select *Tools* > *Options* to open the corresponding window. Once in the Options menu, select *Advanced* on the left pane, then click on the *Launch PuTTY* button on the right side of the window.



☐ Write log file (mRemoteNG.log)
☐ Completely encrypt connection file
☐ Automatically get session information
☐ Automatically try to reconnect when disconnected from server (RDP & ICA only)
☐ Use custom PuTTY path:

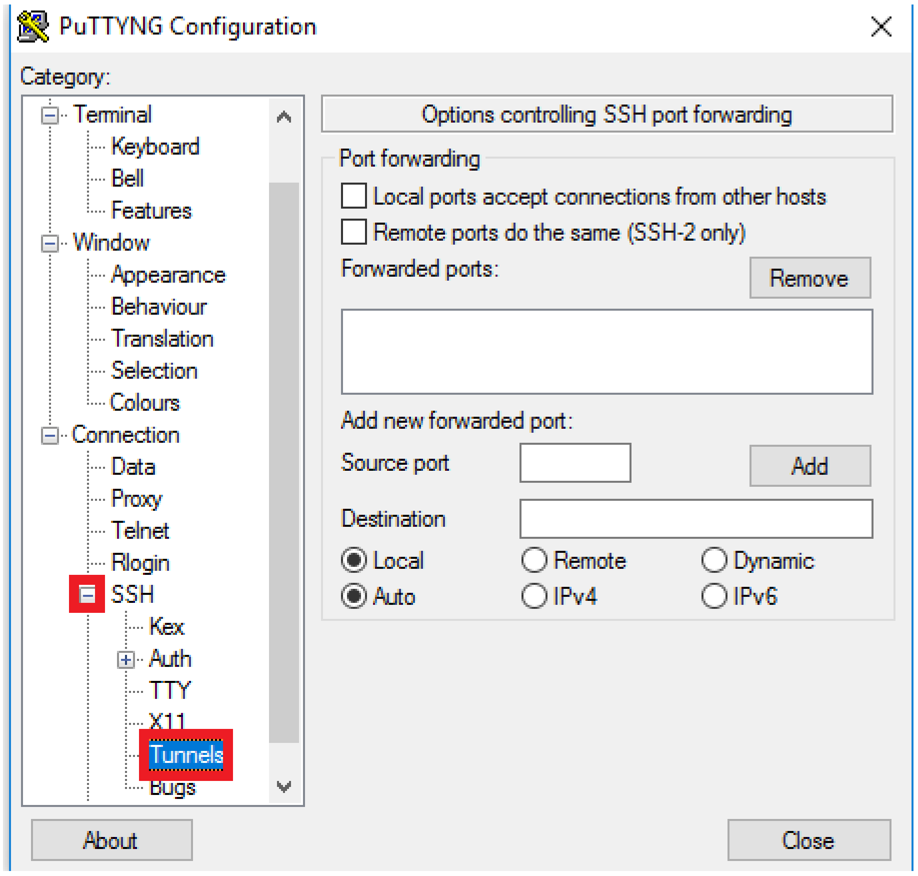
To configure PuTTY sessions click this button: Launch PuTTY

Maximum PuTTY and integrated external tools wait time: 2 seconds

XULrunner path:

UltraVNC SingleClick port: 5500

This should open up a window titled *PuTTYNG Configuration*. (The PuTTY executable is employed by mRemoteNG to establish SSH connections.) Under the *Category* pane, click on the plus sign next to *SSH*, then click to select the *Tunnels* entry.



Options controlling SSH port forwarding

Port forwarding

☐ Local ports accept connections from other hosts

☐ Remote ports do the same (SSH-2 only)

Forwarded ports: Remove

Add new forwarded port:

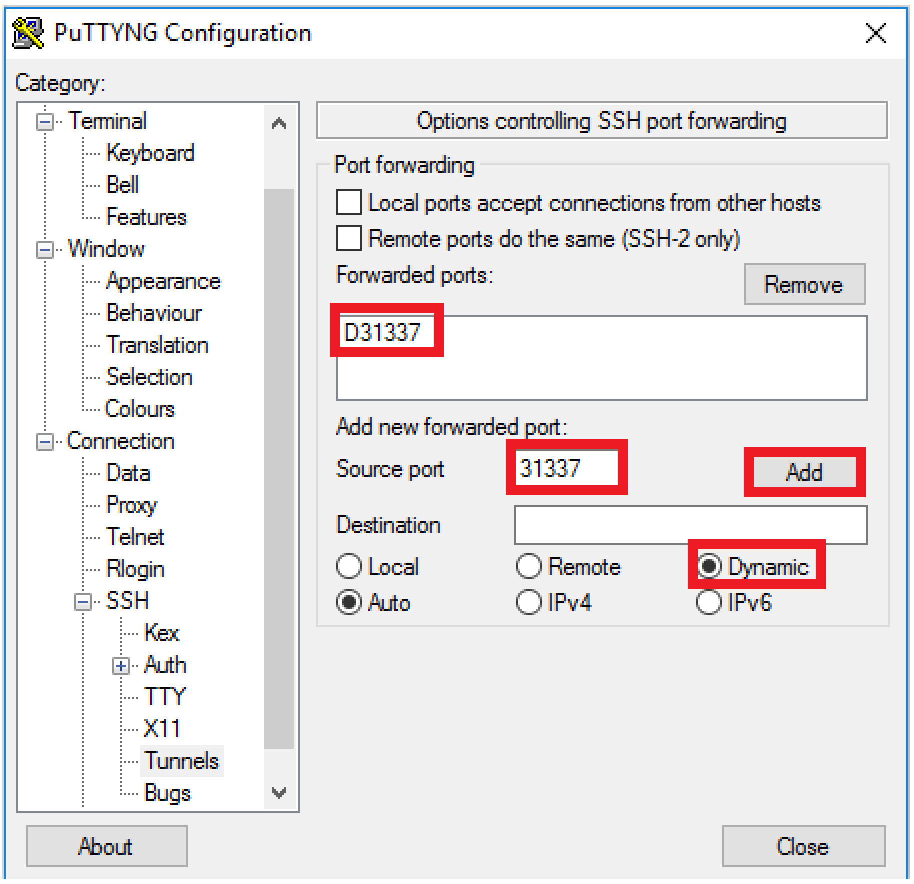
Source port: Add

Destination:

[✓ Local] Remote Dynamic

[✓ Auto] IPv4 IPv6

Enter a random number (above 1024) into the box next to the label *Source port*. This value sets the TCP port that the Windows workstation will bind to when a connection is created based on these configuration settings. I recommend picking a port number above 30000, like 31337, or another random, high-numbered TCP port. Leave the input box named *Destination* blank, click the radio button next to *Dynamic*, make sure that the one next to *Auto* is set, then finish by clicking the *Add* button. If you did it right, the value you entered for the source port, expanded with a leading "D", should appear in the message box labeled *Forwarded ports:*. In my case, I chose 31337 for the source port, resulting in the value "D31337" being displayed in said box, as you can see in the picture below.



Port forwarding

☐ Local ports accept connections from other hosts

☐ Remote ports do the same (SSH-2 only)

Forwarded ports: D31337

Add new forwarded port:

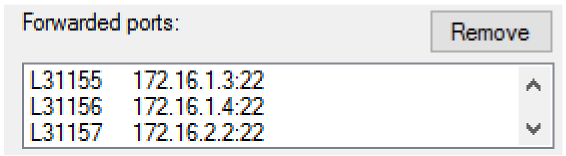
Source port: 31337

Destination:

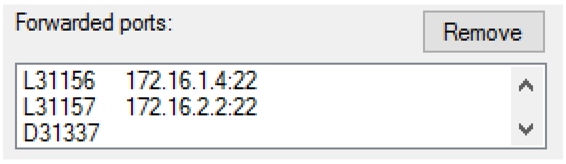
Local [✓ Dynamic] Remote

[✓ Auto] IPv4 IPv6

While still in the PuTTYNG Configuration tool, we're going to add three more tunnels that will allow us to access our Lab VMs more easily. Start by entering another random number in the *Source port* input box; for instance, I chose 31155. Next, enter "172.16.1.3:22" into the *Destination* input box. Make sure that the *Local*" radio box is selected, then click the *Add* button. Repeat this process twice, using a different value for the source port each time (I used ports 31156 and 31157), setting the destination to "172.16.1.4:22" and "172.16.2.2:22", respectively. Your *Forwarded ports:* field should look similar to the ones in the pictures below:



```
L31155    172.16.1.3:22
L31156    172.16.1.4:22
L31157    172.16.2.2:22
```

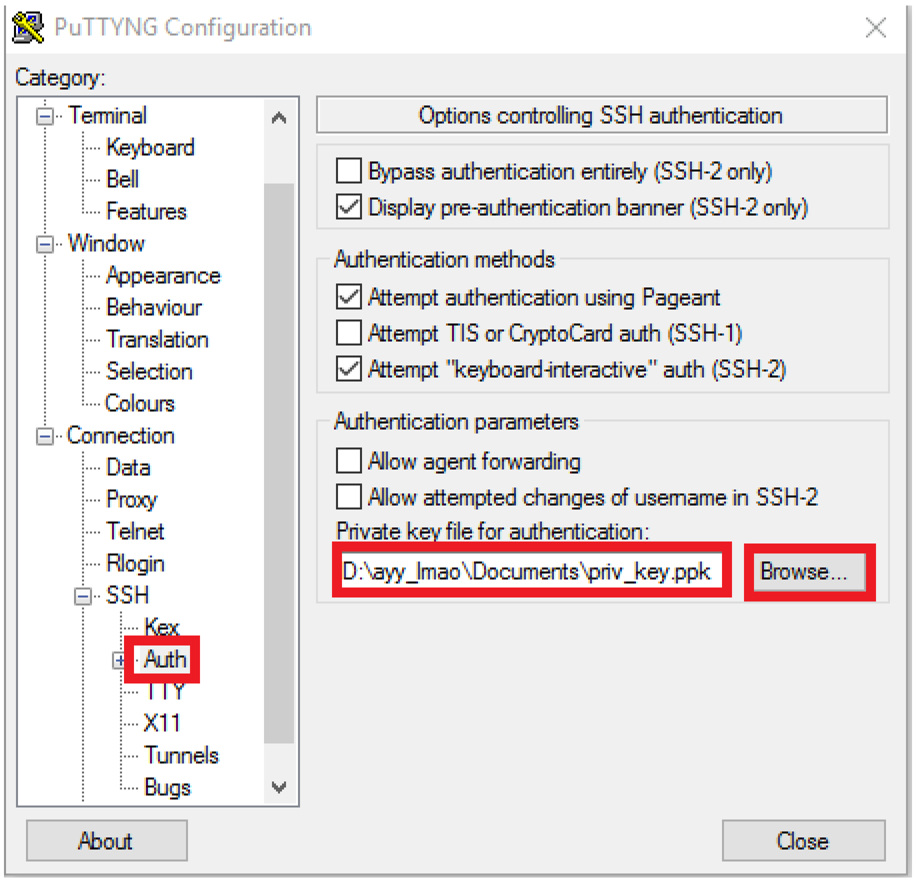


```
L31156    172.16.1.4:22
L31157    172.16.2.2:22
D31337
```

Note: The entry D31337 should be there, you just might have to scroll down in the "Forwarded ports" field to reveal it. **There should be <u>FOUR</u> entries in total.**

This configuration creates a ***<u>new</u>*** PuTTY session specifically for connecting to our jump box with one dynamic TCP tunnel, and three forward connect TCP tunnels. The dynamic tunnel allows us to use our jump box as an HTTP/HTTPS proxy, employing its network connection to interact with the pfSense webConfigurator, while the forward tunnels enables us to send SSH connection requests to our lab VMs via the SSH service of our jump box.

Continue the configuration setup by selecting on the entry labeled *Auth* in the Category pane, then click the browse button next to the input box titled *Private key file for authentication:*. In the Explorer window that opens up, browse to the private key file (the one with a ".ppk" extension) you created with PuTTYgen earlier, and double-click select it. On return to the *PuTTYNG Configuration* window, the input box should populate with the path to your private key file.



Options controlling SSH authentication

[ ] Bypass authentication entirely (SSH-2 only)
[✓] Display pre-authentication banner (SSH-2 only)

Authentication methods

[✓] Attempt authentication using Pageant
[ ] Attempt TIS or CryptoCard auth (SSH-1)
[✓] Attempt "keyboard-interactive" auth (SSH-2)

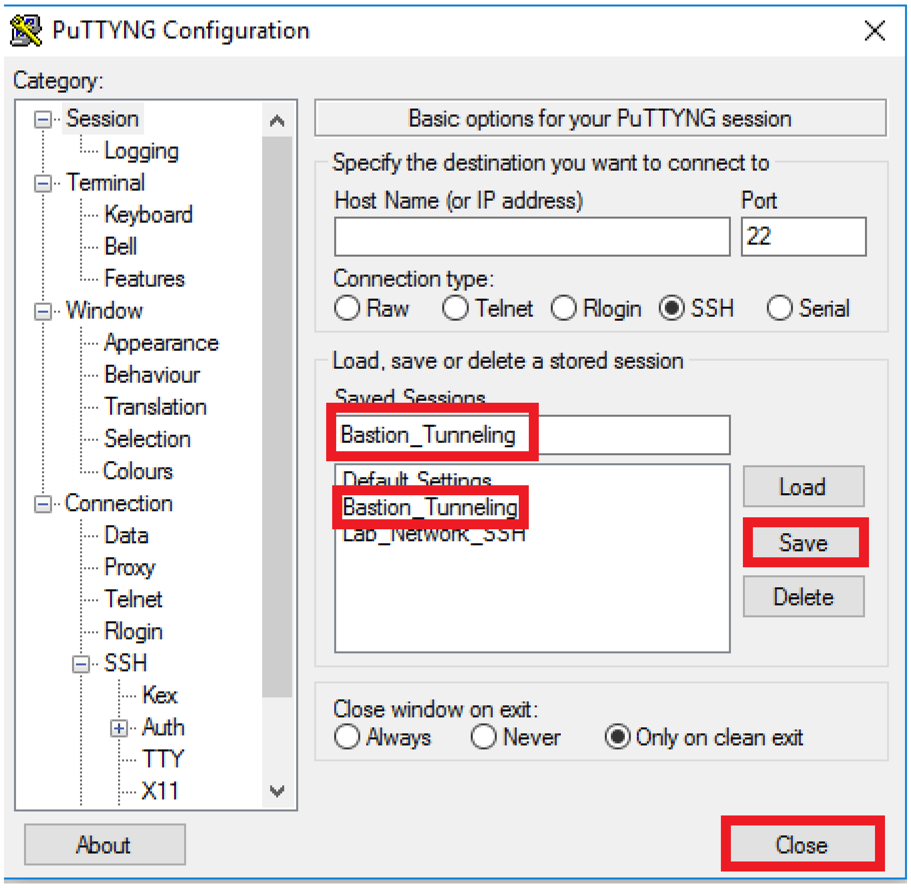
Authentication parameters

[ ] Allow agent forwarding
[ ] Allow attempted changes of username in SSH-2

Private key file for authentication:

D:\ayy_lmao\Documents\priv_key.ppk

Finally, scroll up on the Category pane and select *Session*. Find the box titled *Saved Sessions*, enter a short description for this PuTTY connection profile (I entered "Bastion_Tunneling"), then click the *Save* button. Finish this setup by clicking the *Close* button to exit *PuTTYNG Configuration*.



Port: 22

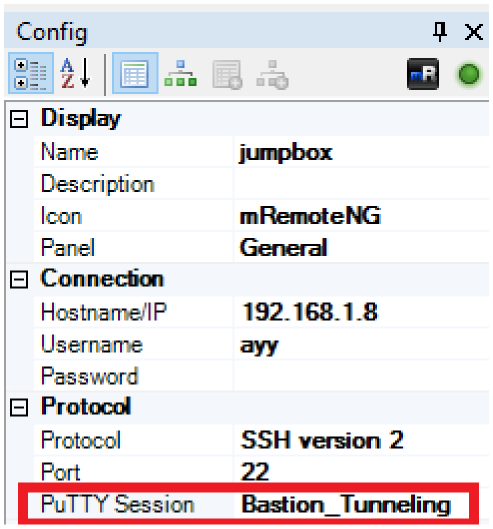
Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Saved Sessions: Bastion_Tunneling

Default Settings
Bastion_Tunneling
Lab_Network_SSH

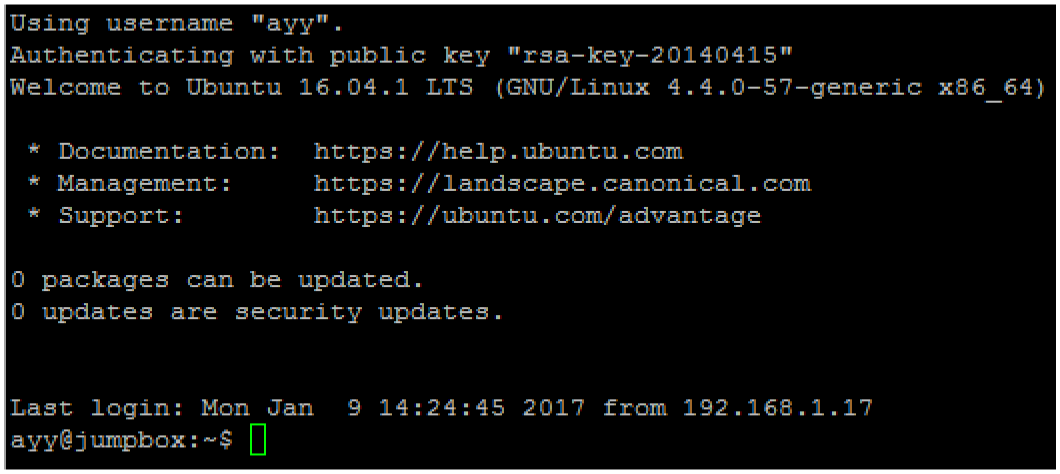
Close window on exit: Always Never [✓ Only on clean exit]

Back in mRemoteNG, close the *Options* menu window to return to the main screen. Under *Connections*, select the entry you created for your jump box. In my case, the session is named "jumpbox". In the *Config* portion, under *Protocol*, find the entry called *PuTTY Session*. When clicked on, a small box with a downward facing arrow appears. Click on this arrow to get a drop-down menu, listing all the PuTTY sessions mRemoteNG is configured with. Click on the one you previously created (which in my case is Bastion_Tunneling) to select it.



Display	
Name	jumpbox
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	192.168.1.8
Username	ayy
Password	
Protocol	
Protocol	SSH version 2
Port	22
PuTTY Session	Bastion_Tunneling

Finally, click *File > Save Connection File* to save your changes. Double click on your newly modified mRemoteNG session for your jump box to see that it works and you are able to authenticate. If everything was configured correctly, you should be greeted by a screen similar to the one in the pictured below.

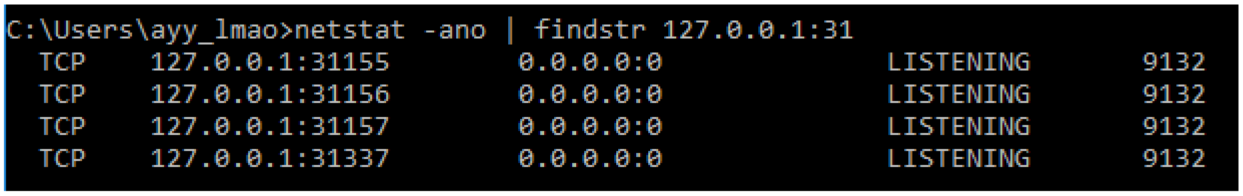


```
Using username "ayy".
Authenticating with public key "rsa-key-20140415"
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-57-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Mon Jan  9 14:24:45 2017 from 192.168.1.17
ayy@jumpbox:~$ 
```

If the connection attempt worked without problems, open up a command prompt on your Windows host, then run the following:

```
netstat -ano | findstr 127.0.0.1:31
```

Note: You might have to change the trailing "31" if you entered different values for the source port setup in your recently created PuTTY session configuration. In that case replace the 31 with the first two digits of the numbers you used.



```
C:\Users\ayy_lmao>netstat -ano | findstr 127.0.0.1:31
  TCP    127.0.0.1:31155        0.0.0.0:0              LISTENING       9132
  TCP    127.0.0.1:31156        0.0.0.0:0              LISTENING       9132
  TCP    127.0.0.1:31157        0.0.0.0:0              LISTENING       9132
  TCP    127.0.0.1:31337        0.0.0.0:0              LISTENING       9132
```

The resulting output (which for my setup looked like the one in the image above) shows all lines returned by the `netstat` command that contain "127.0.0.1:31". Since all of the current connection's tunnels open a port on 127.0.0.1 (localhost) on TCP ports 31155-57 and 31337, I searched for "31", which is the common denominator of the TCP ports I defined in the Bastion_Tunneling PuTTY session on my machine. Since the command has returned four entries, it appears that the tunnels are all functioning normally.

When you are finished here, you can proceed to Testing your Dynamic Tunnels with FoxyProxy.

Linux/BSD/OS X SSH Tunnels

Earlier, on our management workstation, we created an alias to establish SSH connections to our jump box and saved it to our user's `~/.bashrc` or `~/.bash_profile` configuration. Before you make any change to this file again, I highly recommend making a backup copy by running

```
cp ~/.bash_profile ~/.bash_profile_bak; cp ~/.bashrc ~/.bashrc_bak
```

just to cover your bases and make sure you have something to recover from in case your changes don't work as expected.

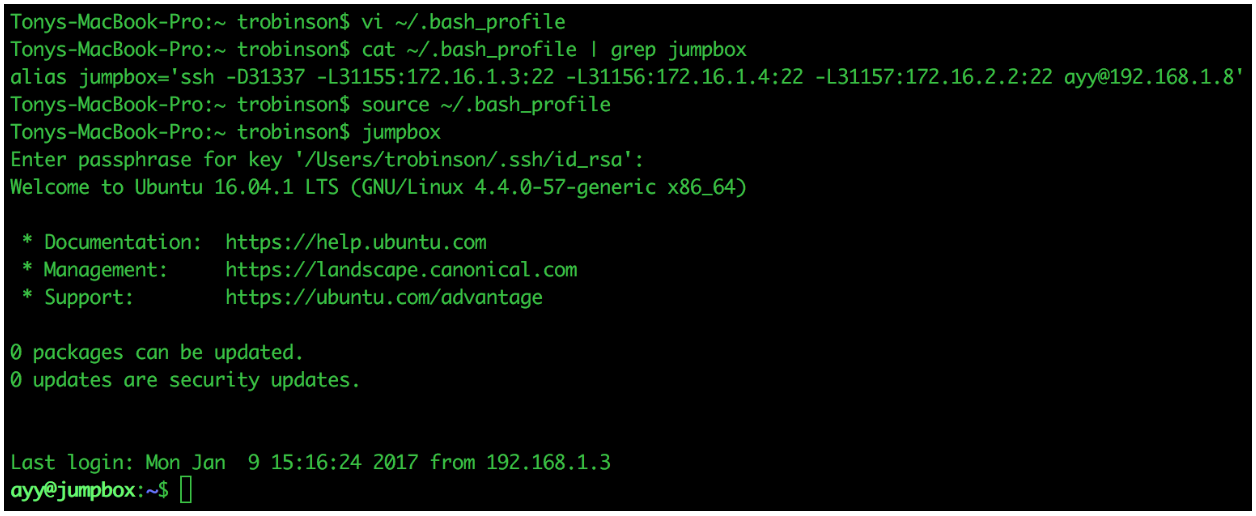
Next, using your favorite text editor, you need to modify the alias entry for your jumpbox system to:

```
alias jumpbox='ssh -D31337 -L31155:172.16.1.3:22 -L31156:172.16.1.4:22 -
L31157:172.16.2.2:22 ayy@192.168.1.8'
```

Note: In the alias above, the ports 31337, 31155-57 can be **any** TCP port **above** 1024, as long as the port is NOT currently in use by the operating system, and the port number chosen for each `-L` or `-D` option is NOT the same. I recommend using TCP ports 30000 or above. As a reminder, you may need to modify the finishing portion of your `ssh` alias from `ayy@192.168.1.8` to the username and IP address you configured for jump box in your environment.

This alias tries to establish an SSH connection to the bastion host, including one dynamic (`-D`) TCP tunnel, and three Local (`-L`) forward connect TCP tunnels. The dynamic tunnel mentioned above allows us to use our bastion host as a proxy, employing its network connection to interact with the pfSense webConfigurator, while the forward tunnels enables us to send SSH connection requests to our lab VMs via the SSH service of our jump box.

Save the changes to your `.bash_profile` or `.bashrc` file, then either use the `source` command (e.g. `source ~/.bashrc` OR `source ~/.bash_profile`) to reload the configuration, or simply reboot your management workstation. After that, verify you can connect to your jump box by using the adjusted alias.

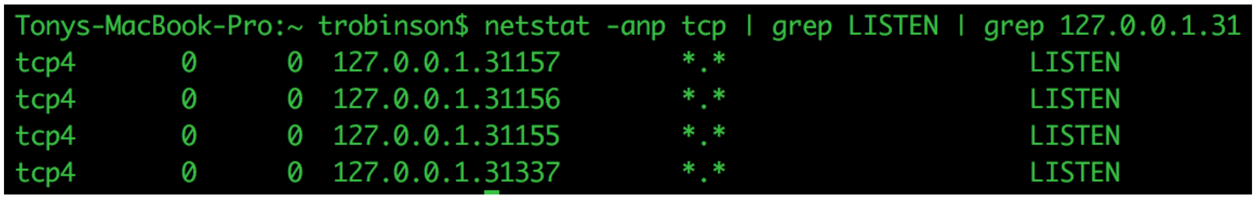


```
Tonys-MacBook-Pro:~ trobinson$ vi ~/.bash_profile
Tonys-MacBook-Pro:~ trobinson$ cat ~/.bash_profile | grep jumpbox
alias jumpbox='ssh -D31337 -L31155:172.16.1.3:22 -L31156:172.16.1.4:22 -L31157:172.16.2.2:22 ayy@192.168.1.8'
Tonys-MacBook-Pro:~ trobinson$ source ~/.bash_profile
Tonys-MacBook-Pro:~ trobinson$ jumpbox
Enter passphrase for key '/Users/trobinson/.ssh/id_rsa':
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-57-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Mon Jan  9 15:16:24 2017 from 192.168.1.3
ayy@jumpbox:~$ 
```

If the connection attempt worked without problems and you're using OS X/BSD, open a terminal session, then enter and execute the following:

```
netstat -anp tcp | grep LISTEN | grep 127.0.0.1.31
```

Note: You might have to change the trailing "31" if you entered different values for the ports in your alias. In that case replace the 31 with the first two digits of the numbers you used.



```
Tonys-MacBook-Pro:~ trobinson$ netstat -anp tcp | grep LISTEN | grep 127.0.0.1.31
tcp4       0      0  127.0.0.1.31157        *.*                    LISTEN
tcp4       0      0  127.0.0.1.31156        *.*                    LISTEN
tcp4       0      0  127.0.0.1.31155        *.*                    LISTEN
tcp4       0      0  127.0.0.1.31337        *.*                    LISTEN
```

This command runs `netstat -anp tcp`, which in this case, is exclusively looking for current TCP connections. We then use the `grep` command to check the results returned by the previously run program for lines that contain `LISTEN` (as in that port is listening for connections). We then use `grep` again to look for any lines within the results from the earlier execution that contain "127.0.0.1.31". This specifically looks for TCP sessions listening for connections on 127.0.0.1 (localhost). All of the TCP ports defined in the example `ssh` alias command above begin with "31", resulting in the `grep` command matching on 31155-31157 and 31337. Since the command has returned four entries, it appears that the tunnels are all functioning normally.

Linux users should run:

```
netstat -tlpn | grep 127.0.0.1:31
```

to receive similar results. Just like with OS X/BSD systems, you are expecting to have four lines returned, similar to what was explained above. When you are finished here, proceed to Testing your Dynamic Tunnels with FoxyProxy.

Testing your Dynamic Tunnels with FoxyProxy

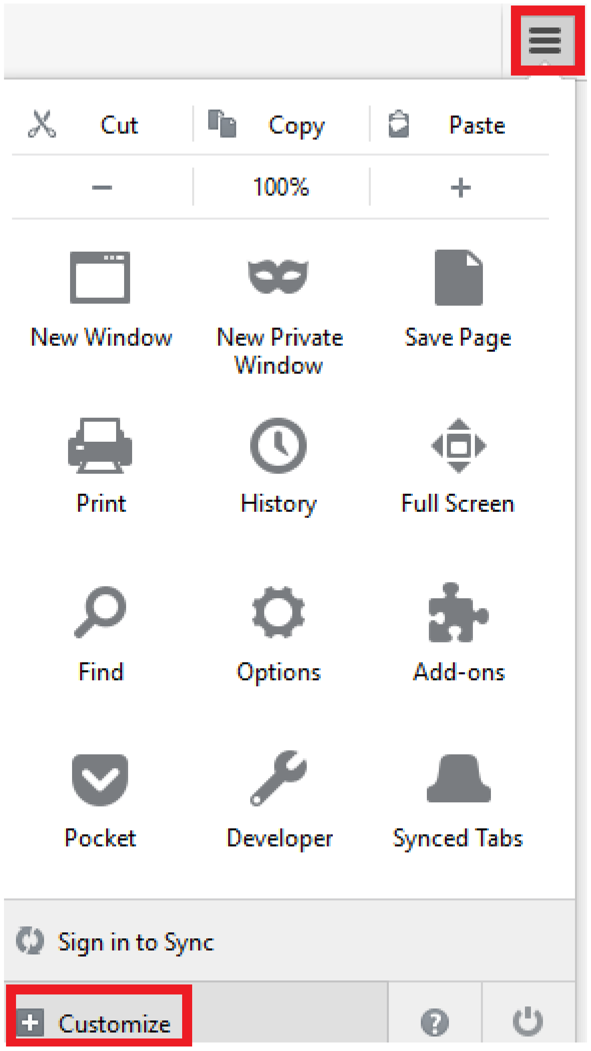
Before we move on to actually testing the dynamic tunnels we just prepared, we need to configure our management workstation's web browser to use the jump box as a proxy. Since they both support the browser extension FoxyProxy, I recommend using either the current version of the Google Chrome or Mozilla Firefox web browser. The FoxyProxy extension allows you to create multiple proxy configurations/sessions that can be easily switched between, and instantly applied to your web browser at any time. With the help of this tool and the necessary settings in place, we will be able to use our management workstation to access the pfSense webConfigurator via the dynamic SSH tunnel established to our jump box. For this guide, I used Mozilla Firefox on Windows 10, however, aside from the initial installation of the add-on and how to access it, configuring FoxyProxy for utilizing our dynamic tunnel should be nearly identical for users of both Chrome and Firefox, on Windows as well as on OS X, Linux and BSD.

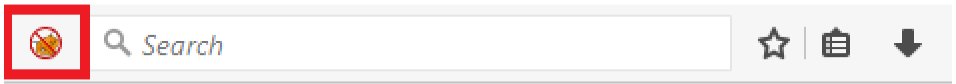
Note: Use of FoxyProxy, Google Chrome, or Mozilla Firefox isn't mandatory in order to use our Dynamic SSH tunnel, this just happens to be the easiest way, and the method I recommend. Most modern web browsers have configuration options that allow users to manually specify a proxy to connect to. Pay attention to the detail that the dynamic SSH tunnel should be registered as a SOCKS v5 proxy.

You can download

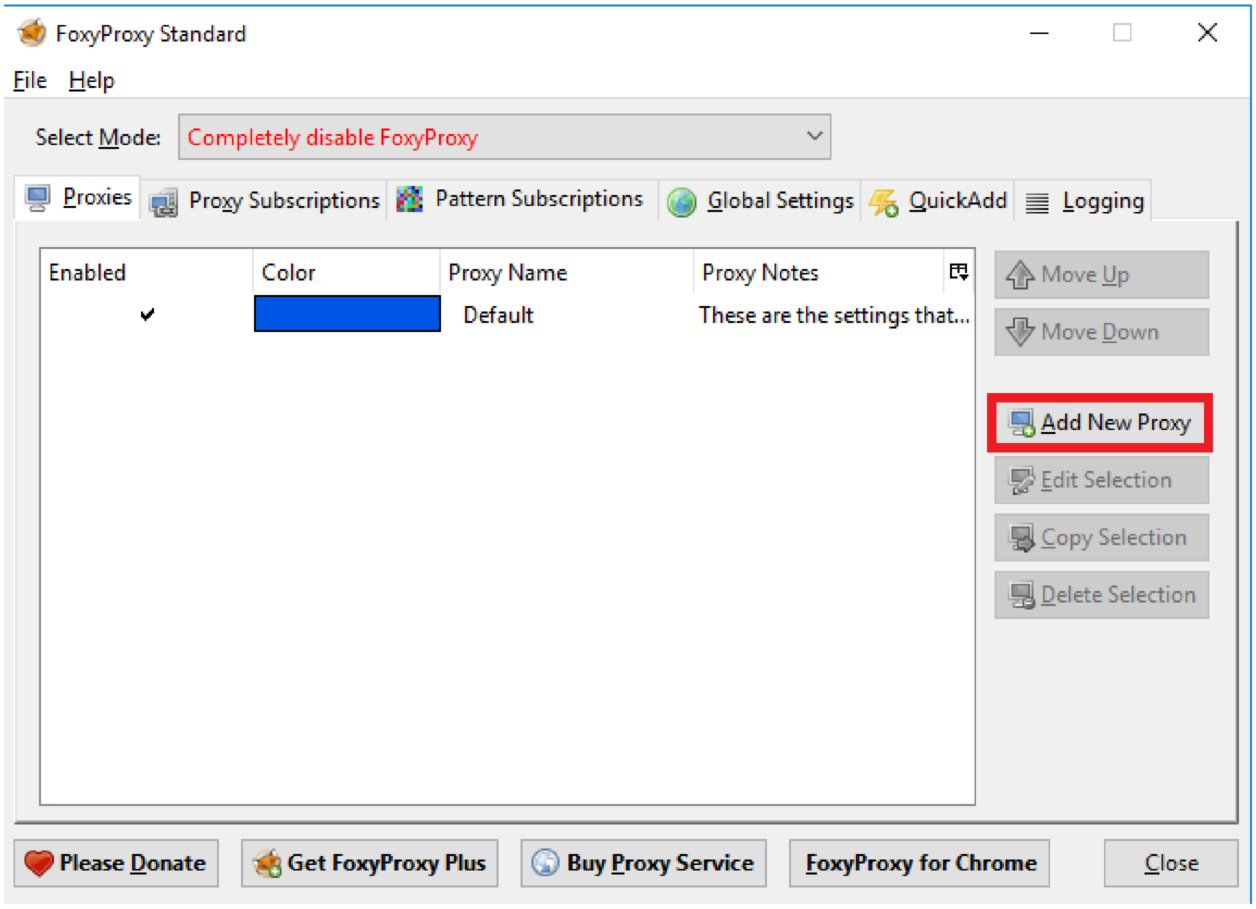
- Mozilla Firefox at https://www.mozilla.org/en-US/firefox/desktop/
- Google Chrome at https://www.google.com/chrome/
- FoxyProxy at https://getfoxyproxy.org/downloads/

After installing the extension, open your web browser, and find the the FoxyProxy settings to add a new proxy, and follow along. If you are using Firefox and haven't done so already, I highly recommend adding the icon for FoxyProxy to the web bar in Firefox. This can be done by clicking the *Settings* icon, then the *Customize* option in the menu, followed by dragging the FoxyProxy icon to the address bar.





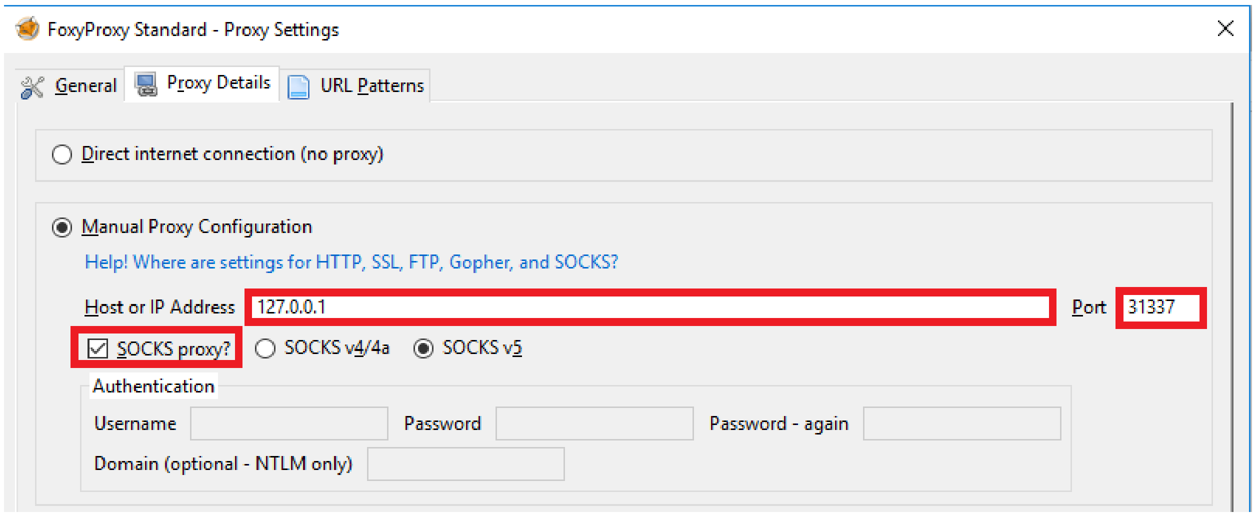
Click on the FoxyProxy icon (or otherwise, enter your browser's configuration menu and select FoxyProxy) to bring up the window titled *FoxyProxy Standard*, then click the *Add New Proxy* button on the right.



Select Mode: Completely disable FoxyProxy

Enabled	Color	Proxy Name	Proxy Notes
✓	🟦	Default	These are the settings that...

In the window labeled *FoxyProxy Standard - Proxy Settings*, select the *Proxy Details* tab. Ensure that the *Manual Proxy Configuration* radio button is selected, then enter “127.0.0.1” into the box next to the label *Host or IP Address* . Move on to the input box titled *Port* and insert the very same value you used for creating/configuring your dynamic tunnel via `ssh -D` or via mRemoteNG/PuTTYNG Configuration. In my case, this is port 31337. Finally, click the checkbox labeled “SOCKS proxy?” to check the box, and verify the radio button next to *SOCKS v5* is selected. Finish by clicking *OK* to save the new proxy and to return to the previous window.


FoxyProxy Standard - Proxy Settings

Direct internet connection (no proxy)

[✓ Manual Proxy Configuration]

Help! Where are settings for HTTP, SSL, FTP, Gopher, and SOCKS?

Host or IP Address: 127.0.0.1   Port: 31337

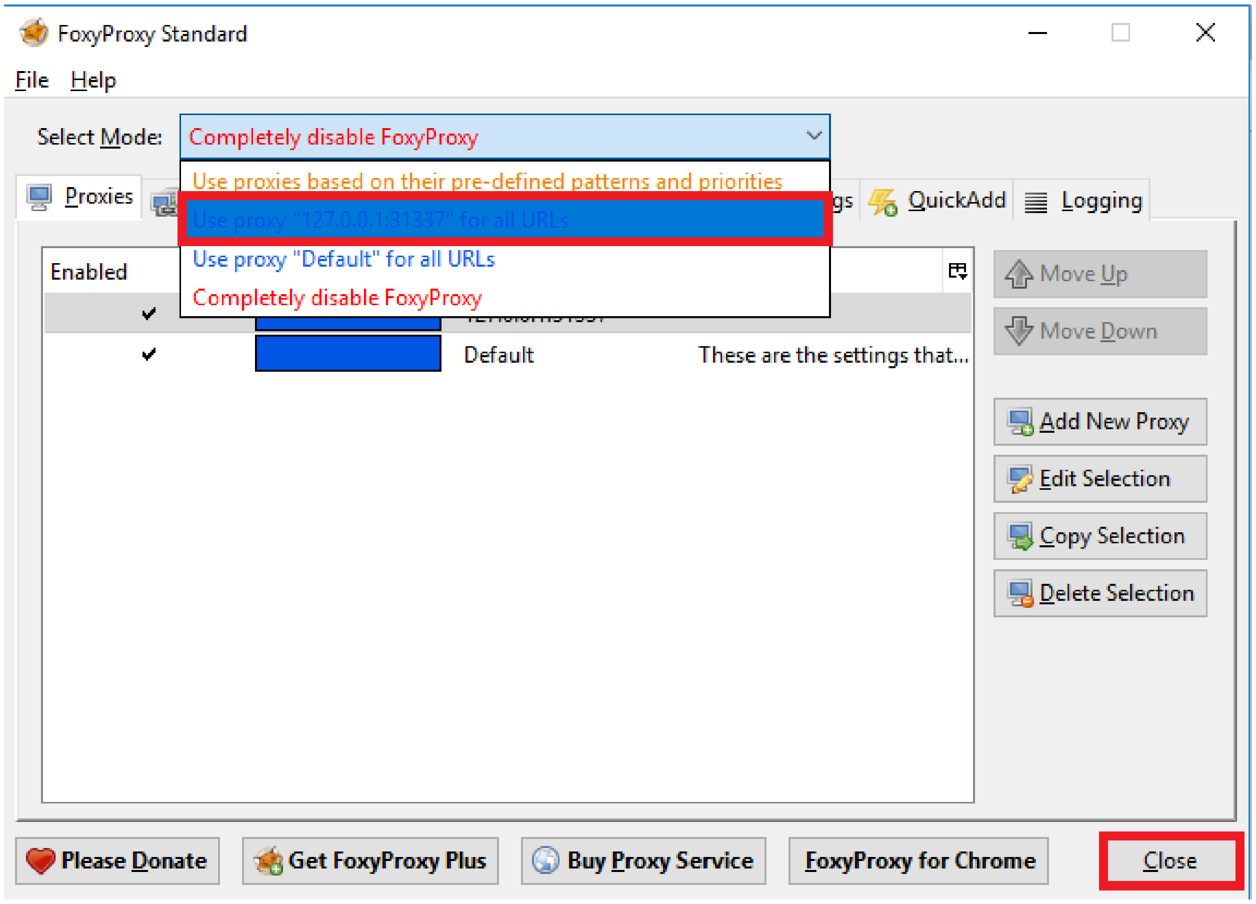
[✓ SOCKS proxy?]   SOCKS v4/4a   [✓ SOCKS v5]

Authentication

Username   Password   Password - again

Domain (optional - NTLM only)

Back in the main menu, find the drop-down box named *Select Mode:* with its default value set to *Completely disable FoxyProxy*. Click on this text to display a drop-down menu, and select the item *Use proxy "127.0.0.1:31337" for all URLs*. Exit the *FoxyProxy Standard* window by clicking the *Close* button in the lower right corner.


FoxyProxy Standard
File
Help
Select Mode:
Completely disable FoxyProxy
Use proxies based on their pre-defined patterns and priorities
Use proxy "127.0.0.1:31337" for all URLs
Use proxy "Default" for all URLs
Completely disable FoxyProxy
Proxies
QuickAdd
Logging
Enabled
Default
These are the settings that...
Move Up
Move Down
Add New Proxy
Edit Selection
Copy Selection
Delete Selection
Please Donate
Get FoxyProxy Plus
Buy Proxy Service
FoxyProxy for Chrome
Close

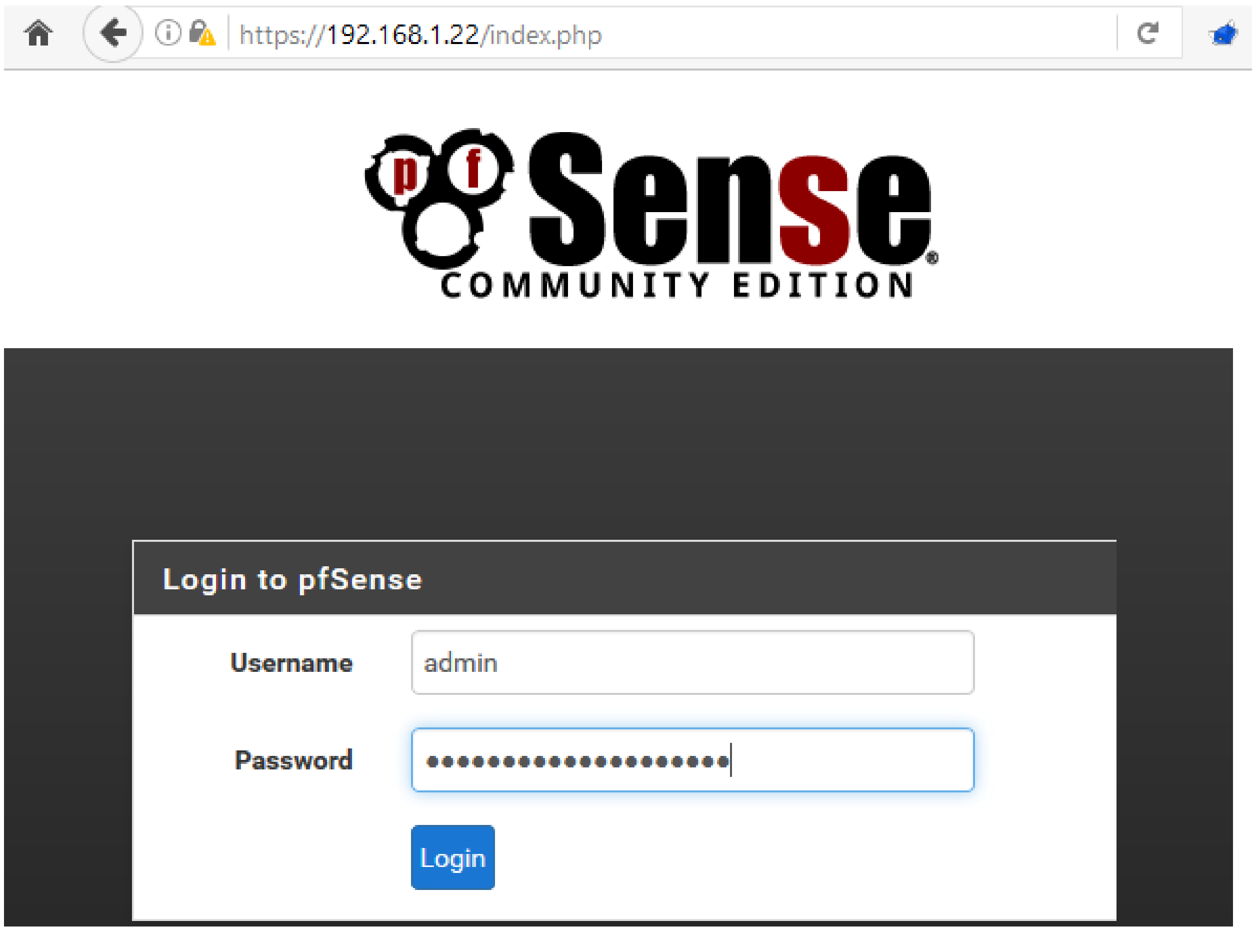
Note: When setting FoxyProxy's mode to *Use proxy "x.x.x.x:xxxxxx" for all URLs*, that is exactly what is going to happen: **all of your web traffic will be funneled through your jump box.** I would highly suggest experimenting with Firefox and Chrome profiles/sessions, along with creating a separate profile to be used exclusively for accessing the pfSense web UI through the jump box. Alternatively, you can simply switch between Using your dynamic SSH tunnel proxy and the option *Completely disable FoxyProxy* as needed.

For more on adding profiles to Chrome, visit https://support.google.com/chrome/answer/2364824?co=GENIE.Platform%3DDesktop&hl=en. Chrome grants you the ability swap profiles directly in the web browser.

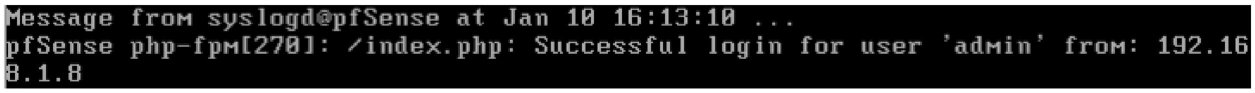
Firefox users should create a new profile via the profile manager, and make themselves familiar with Firefox's `-ProfileManager` and `--no-remote` command line options. For more information on this, see: https://developer.mozilla.org/en-US/docs/Mozilla/Command_Line_Options.

With FoxyProxy set up, we are ready to actually test and verify that the dynamic tunnel/proxy is working. If you closed the SSH session to your jump box that you established at the end of the previous section, start a new one. Make sure to use the newly modified connection profile or alias that creates the dynamic and forward tunnels we set up just a moment ago. This should be obvious, but **you must have an active connection to the jump box system in order to use it as a proxy/tunnel system.**

Next, try to connect to the IP address of the WAN interface on your pfSense system. If you created a firewall rule to allow your jump box to access the webConfigurator interface (or used `pfctl -d`), you should be greeted by its web prompt. If that is the case, proceed by trying to log in.

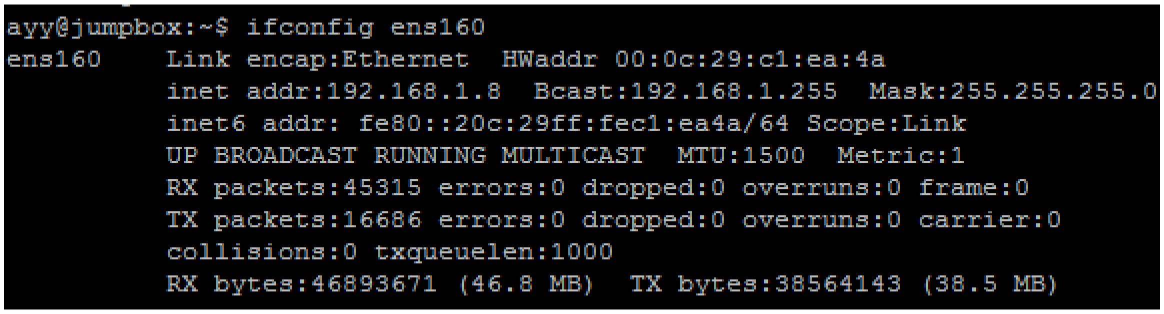

https://192.168.1.22/index.php

If you succeeded, then your dynamic tunnel should be working. If you're like me and you REALLY want to make sure your tunnel is working, log in to the ESXi web interface, and open up a console session to your pfSense VM. Look for the portion of the output that looks similar to the one in the picture below:



```
Message from syslogd@pfSense at Jan 10 16:13:10 ...
pfSense php-fpm[270]: /index.php: Successful login for user 'admin' from: 192.16
8.1.8
```

In my case, the IP address of the jump box system is 192.168.1.8, and my management workstation is 192.168.1.17, the log message on the pfSense console indicates that the login originated from 192.168.1.8, so this confirms that the dynamic tunnel is working properly.

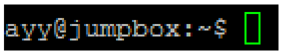


```
ayy@jumpbox:~$ ifconfig ens160
ens160    Link encap:Ethernet  HWaddr 00:0c:29:c1:ea:4a
          inet addr:192.168.1.8  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fec1:ea4a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:45315 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16686 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:46893671 (46.8 MB)  TX bytes:38564143 (38.5 MB)
```

Troubleshooting Dynamic Tunnels

If, instead of the pfSense webConfigurator login screen, your web browser displays an error message similar to `Proxy Server is refusing connections`, `Connection Timed out`, `Connection Refused`, or the pfSense console logs includes an IP address that is NOT the IP address of the jump box (e.g. your management workstation's IP address), don't panic. Instead, start investigating by following the suggested checks below:

Make sure that there is an active SSH session to your jump box. None of your tunnels will work without an active SSH session to your jump box. In mRemoteNG (or in one of your terminal sessions, if you are using Linux, OS X or BSD), look for the respective session prompt, similar to the one you can see in the picture below.



```
ayy@jumpbox:~$
```

While you are connected to your bastion host via SSH, on your management workstation, on Windows, run:

```
netstat -ano | findstr 127.0.0.1:31337 | findstr LISTEN
```

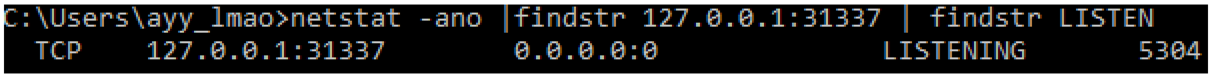
or on OS X/BSD, run:

```
netstat -anp tcp | grep 127.0.0.1.31337 | grep LISTEN
```

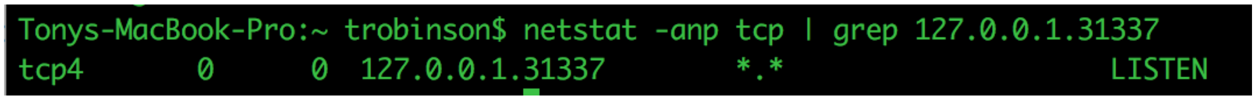
or on Linux, run:

```
netstat -tln | grep 127.0.0.1:31337 | grep LISTEN
```

Note that if you specified another TCP port, replace 31337 with the value you used to make your dynamic tunnel instead. If the respective command returned an output similar to those depicted below, the dynamic tunnel is up and ready to accept connections from your management workstation.

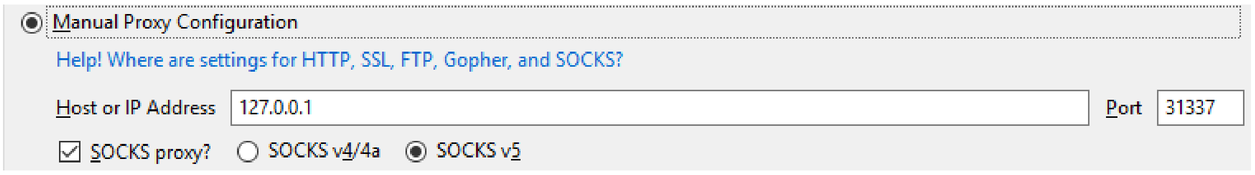


```
C:\Users\ayy_lmao>netstat -ano |findstr 127.0.0.1:31337 | findstr LISTEN
  TCP    127.0.0.1:31337        0.0.0.0:0              LISTENING       5304
```



```
Tonys-MacBook-Pro:~ trobinson$ netstat -anp tcp | grep 127.0.0.1.31337
tcp4       0      0  127.0.0.1.31337        *.*                    LISTEN
```

Verify that the proxy server settings have been successfully added to FoxyProxy. If you used a different number than 31337 for the TCP port of your dynamic SSH tunnel, check that you specified that number, along with the IP address “127.0.0.1” in your FoxyProxy configuration. Make sure that the *SOCKS proxy?* checkbox is checked as well, and that the radio button next to *SOCKS v5* is selected.



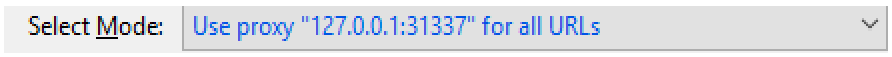
● Manual Proxy Configuration

Help! Where are settings for HTTP, SSL, FTP, Gopher, and SOCKS?

Host or IP Address: 127.0.0.1 Port: 31337

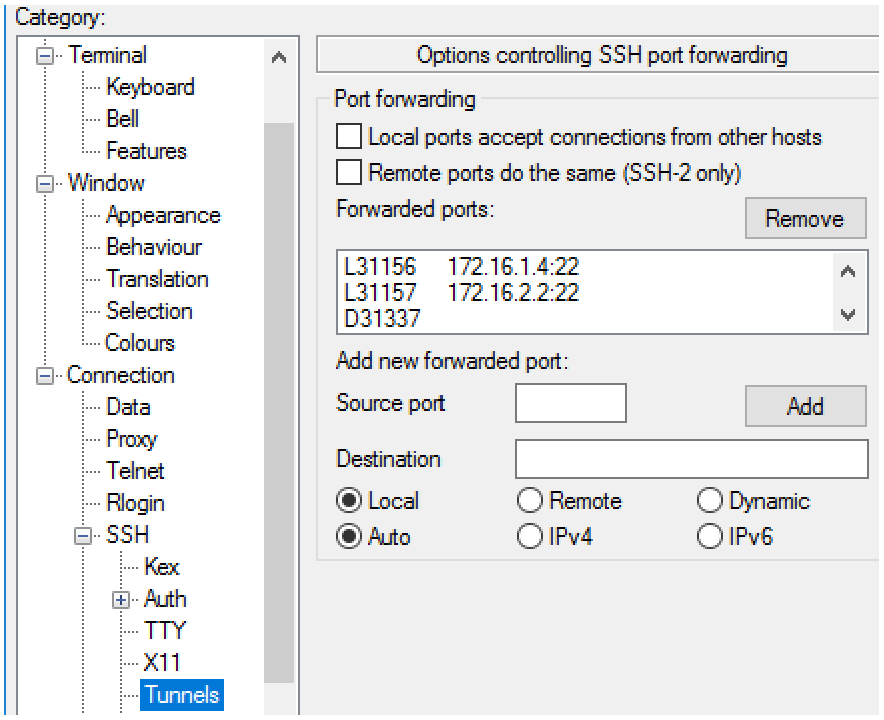
☑ SOCKS proxy? ○ SOCKS v4/4a ● SOCKS v5

Verify FoxyProxy’s mode is set to *Use proxy “127.0.0.1:31337” for all URLs.* (Again, the number that follows the IP will be a different one in case you decided to chose a different port for your dynamic tunnel setup.) Your web browser will only send its traffic through your jump box if you selected this profile as the active one in FoxyProxy.



Select Mode: Use proxy "127.0.0.1:31337" for all URLs

If you’re on Windows, verify that you created the respective PuTTY session with a dynamic tunnel to port 31337, (or the alternate TCP port you specified for it), that you SAVED the corresponding configuration, and that you APPLIED that saved PuTTY session to your mRemoteNG connection item for the jump box system. To finish, close and reestablish the SSH connection to your jump box, using that particular mRemoteNG connection item.



Options controlling SSH port forwarding

Port forwarding

- [ ] Local ports accept connections from other hosts
- [ ] Remote ports do the same (SSH-2 only)

Forwarded ports:

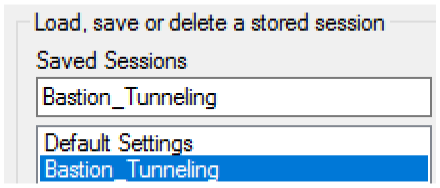
```
L31156    172.16.1.4:22
L31157    172.16.2.2:22
D31337
```

Add new forwarded port:

Source port	
Destination	

[✓ Local] Remote Dynamic

[✓ Auto] IPv4 IPv6

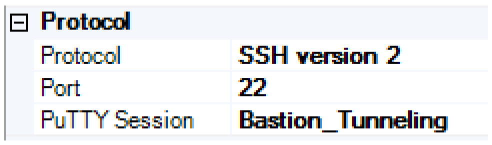


Load, save or delete a stored session

Saved Sessions

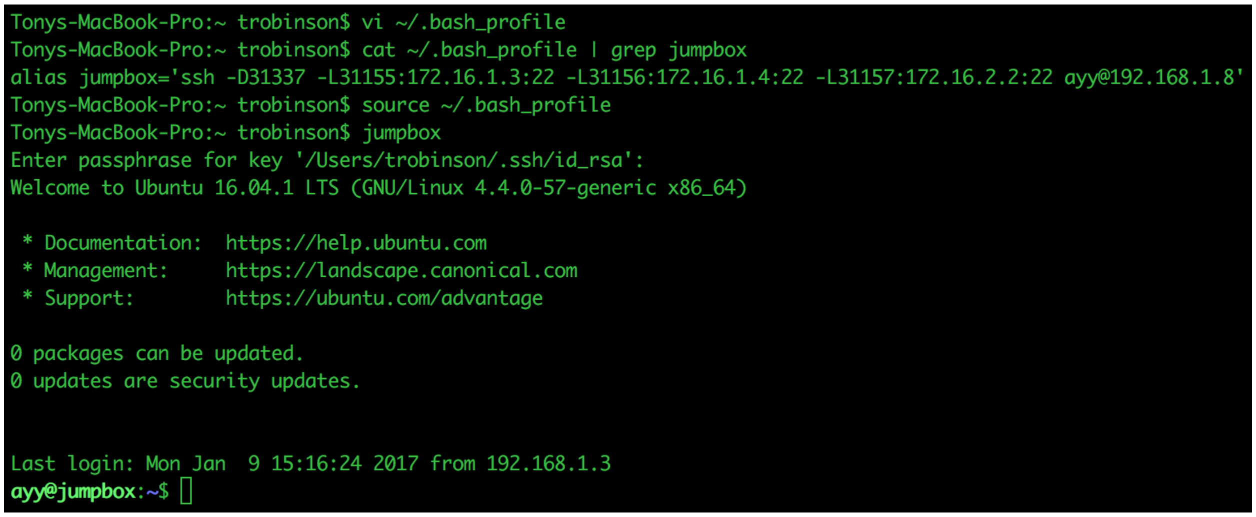
Bastion_Tunneling

Default Settings
Bastion_Tunneling



Protocol	
Protocol	**SSH version 2**
Port	**22**
PuTTY Session	**Bastion_Tunneling**

If you’re on Linux/OS X/BSD, verify that your alias was created properly (see the alias in the illustration below, changing the TCP port, 31337 by default, as necessary if you chose a different port number for your dynamic tunnel), that you added it to `~/.bashrc` or `~/.bash_profile`, and ensure that you sourced `~/.bashrc` or `~/.bash_profile` properly. Finally, close and reestablish the SSH connection to your jump box using the specific alias.



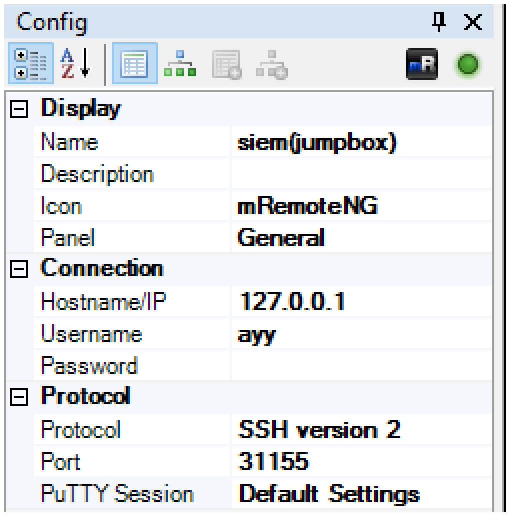
```
Tonys-MacBook-Pro:~ trobinson$ vi ~/.bash_profile
Tonys-MacBook-Pro:~ trobinson$ cat ~/.bash_profile | grep jumpbox
alias jumpbox='ssh -D31337 -L31155:172.16.1.3:22 -L31156:172.16.1.4:22 -L31157:172.16.2.2:22 ayy@192.168.1.8'
Tonys-MacBook-Pro:~ trobinson$ source ~/.bash_profile
Tonys-MacBook-Pro:~ trobinson$ jumpbox
Enter passphrase for key '/Users/trobinson/.ssh/id_rsa':
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-57-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Mon Jan  9 15:16:24 2017 from 192.168.1.3
ayy@jumpbox:~$ 
```

Testing Your Forward Tunnels

In order to proceed and test whether or not the forward tunnels are working and able to forward SSH requests to your lab VMs, the following preconditions apply: you created all of the lab VMs, assigned them the correct IP addresses, and configured their SSH services to be listening for connections. Additionally, you need an active SSH session to your jump box from your management workstation, with your forward listeners configured. For my environment, I set up three forward TCP listeners on my management workstation: 31155 forwards to 172.16.1.3:22 (SSH on the SIEM VM), 31156 forwards to 172.16.1.4:22 (SSH on the IPS VM), and 31157 forwards to 172.16.2.2:22 (SSH on the Kali VM, assuming you enabled SSH on the Kali VM).

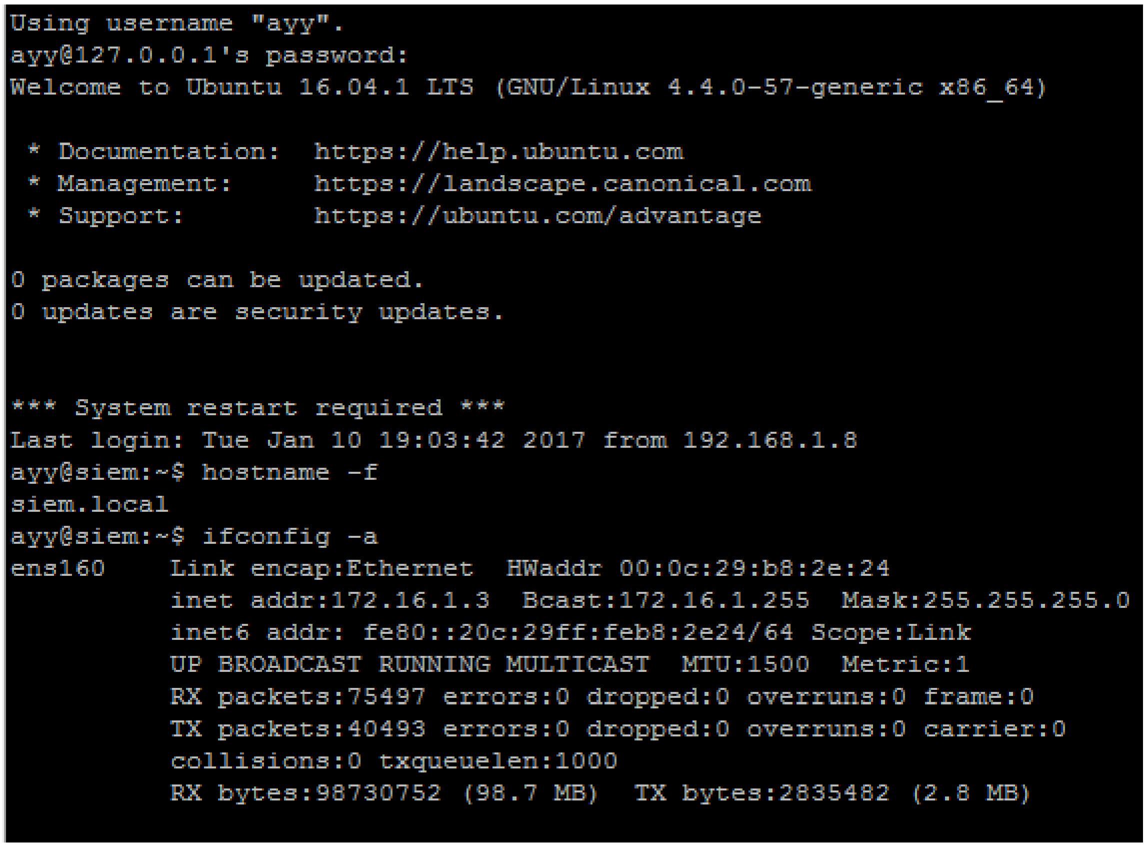
Windows

If you're using Windows on your management workstation, create a new mRemoteNG connection profile and name it "siem(jumpbox)". Enter "127.0.0.1" as the IP address, and set the username to the one you configured when you created the SIEM VM. In the *Protocol* section, choose *SSH version 2*, and change the TCP port number to 31155 (or to the respective value you configured for your forward connect to 172.16.1.3:22). For now, you can set the PuTTY Session to *Default Settings*. The Config for the new item should look similar to the one shown in the image below.



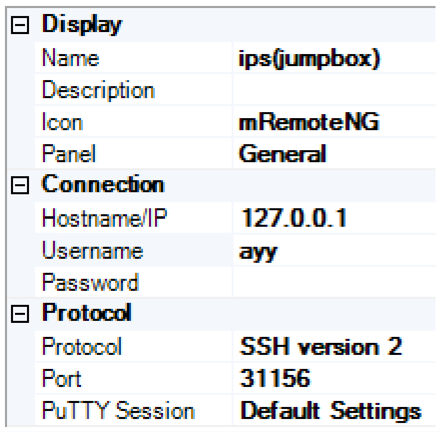
Display	
Name	siem(jumpbox)
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	127.0.0.1
Username	ayy
Password	
Protocol	
Protocol	SSH version 2
Port	31155
PuTTY Session	Default Settings

If you haven't already, open a connection to your jump box system, ensuring that the putty connection profile you used to create your forward and dynamic tunnels is selected. With your SSH session to your jump box still active, and the corresponding tunnels enabled, double click on the previously created connection item. If everything is configured correctly and working as intended, you should reach the SIEM VM, logged in with the user account specified.

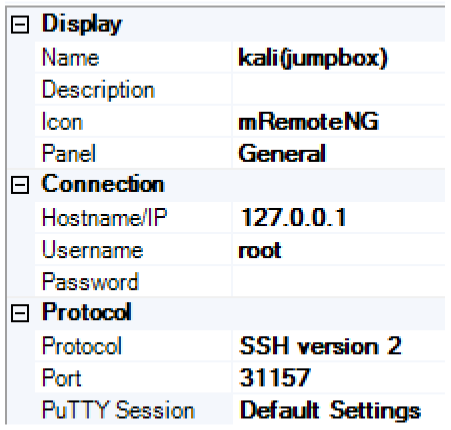


```
Using username "ayy".
ayy@127.0.0.1's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-57-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Tue Jan 10 19:03:42 2017 from 192.168.1.8
ayy@siem:~$ hostname -f
siem.local
ayy@siem:~$ ifconfig -a
ens160    Link encap:Ethernet  HWaddr 00:0c:29:b8:2e:24
          inet addr:172.16.1.3  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb8:2e24/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:75497 errors:0 dropped:0 overruns:0 frame:0
          TX packets:40493 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:98730752 (98.7 MB)  TX bytes:2835482 (2.8 MB)
```

Repeat the above process to create two more connection profiles, with the port on one set to 31156 and to 31157 on the other, in order to establish SSH connections to the IPS VM and Kali VM respectively. (Again, use the values you decided to assign to the listener ports instead of 31156 and 31157.) The images below should give you the idea what the Config sections of the new connection items should look like.

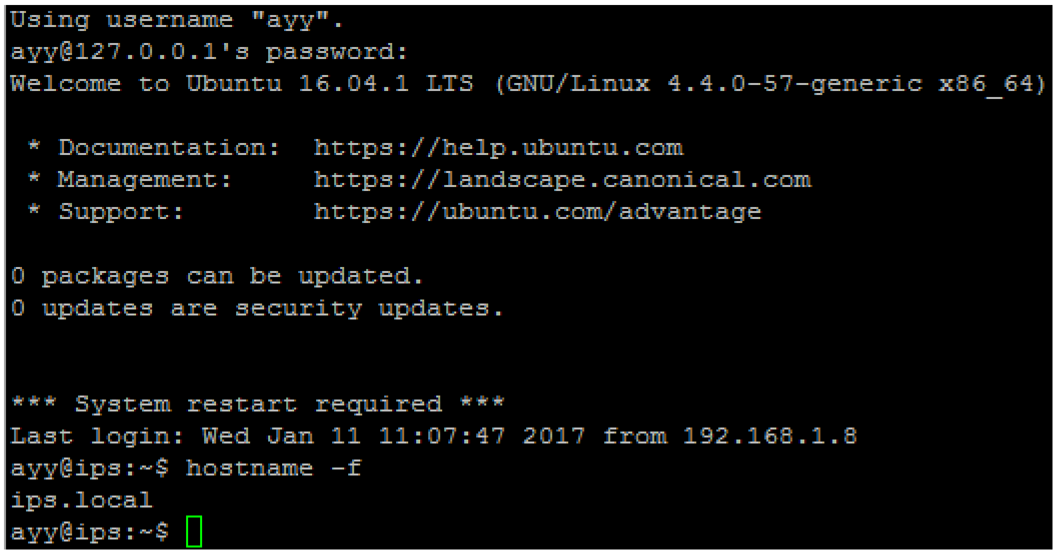


Display	
Name	ips(jumpbox)
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	127.0.0.1
Username	ayy
Password	
Protocol	
Protocol	SSH version 2
Port	31156
PuTTY Session	Default Settings

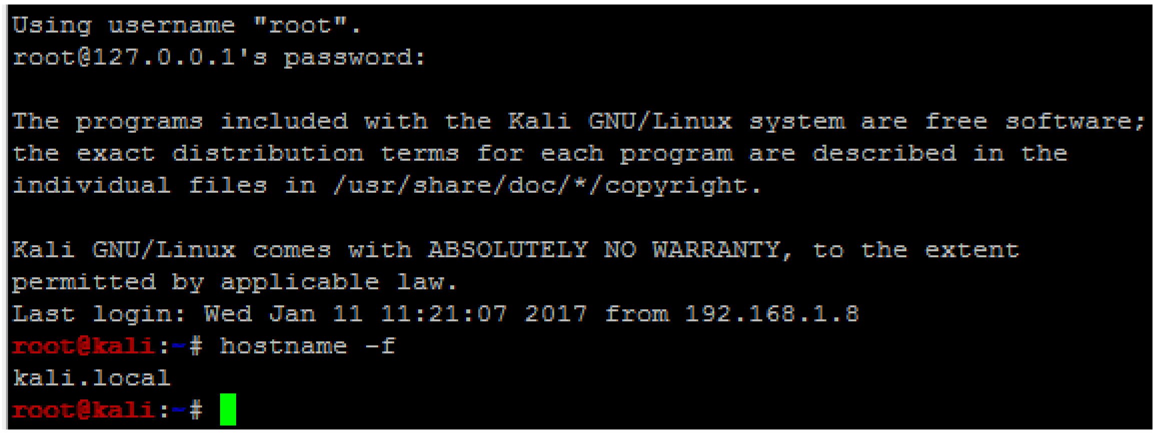


Display	
Name	kali(jumpbox)
Description	
Icon	mRemoteNG
Panel	General
Connection	
Hostname/IP	127.0.0.1
Username	root
Password	
Protocol	
Protocol	SSH version 2
Port	31157
PuTTY Session	Default Settings

When finished, test the newly created connections for the IPS, and Kali VM, just like you did with the one for the SIEM VM.



```
Using username "ayy".
ayy@127.0.0.1's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-57-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Wed Jan 11 11:07:47 2017 from 192.168.1.8
ayy@ips:~$ hostname -f
ips.local
ayy@ips:~$
```



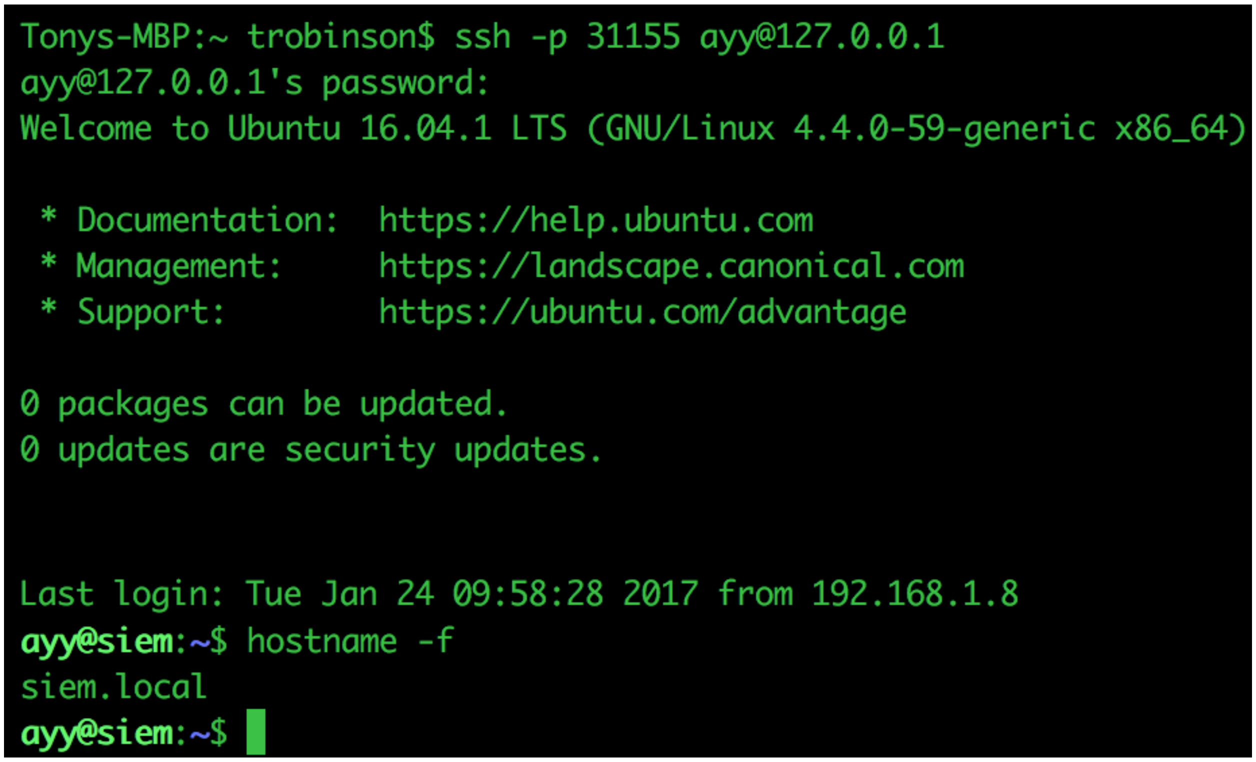
```
Using username "root".
root@127.0.0.1's password:

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jan 11 11:21:07 2017 from 192.168.1.8
root@kali:~# hostname -f
kali.local
root@kali:~#
```

Linux/OS X/BSD

If you're using Linux, OS X or BSD on your management workstation, with your SSH session to your jump box still active and the corresponding tunnels enabled, open another terminal session on your local system, then enter and run the following command:

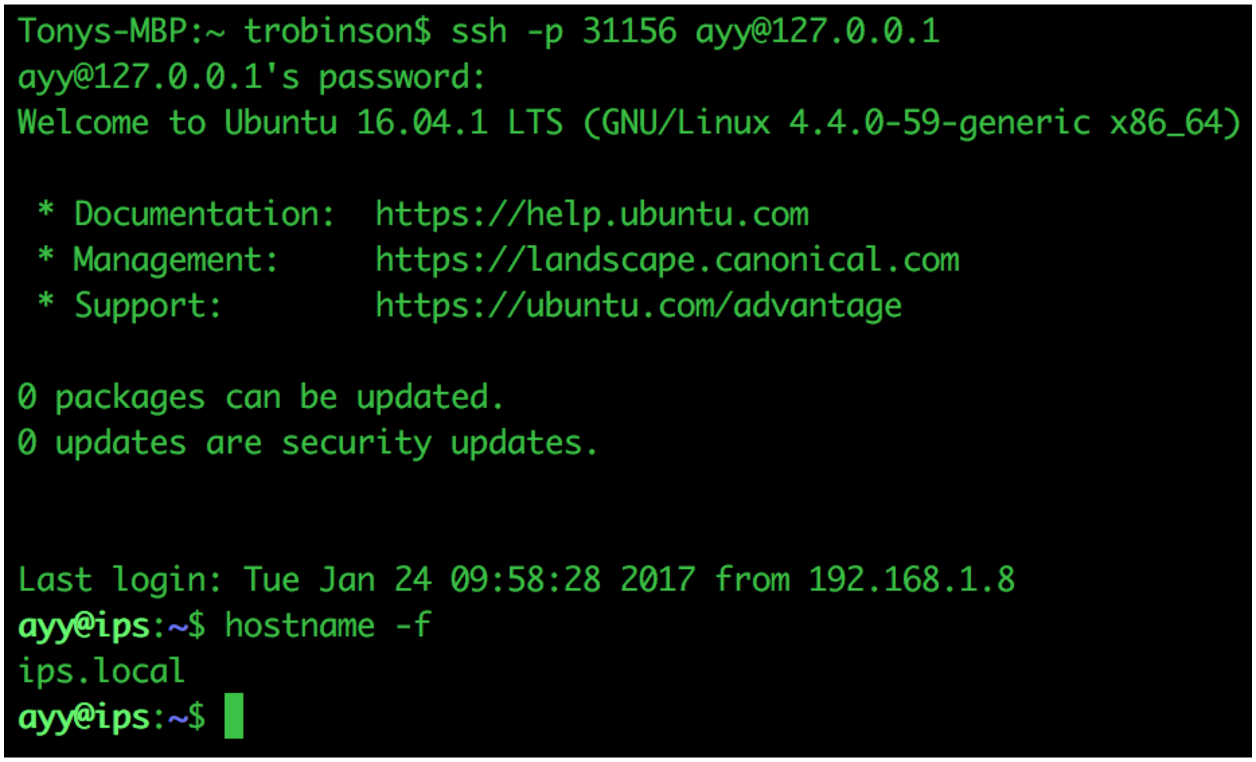
```
ssh -p 31155 ayy@127.0.0.1
```

For your environment, change the value of the `-p` option with the number you assigned to the tunneling port intended to be used with the SIEM VM, and replace `ayy` with the name of the user you configured when you first created this virtual machine. If everything is configured correctly and working as intended, you should reach the remote system, logged in with the user account specified.

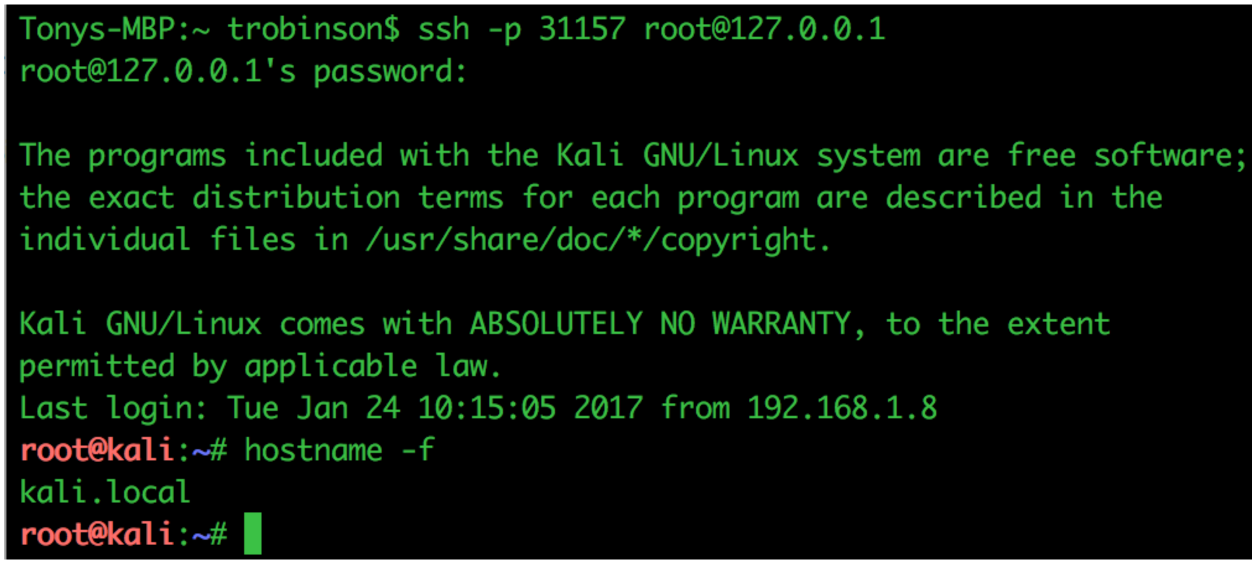


```
Tonys-MBP:~ trobinson$ ssh -p 31155 ayy@127.0.0.1
ayy@127.0.0.1's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-59-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Tue Jan 24 09:58:28 2017 from 192.168.1.8
ayy@siem:~$ hostname -f
siem.local
ayy@siem:~$
```

After confirming that you are able to log in to the SIEM VM, repeat the process and connect to port 31156 (`ssh -p 31156 ayy@127.0.0.1`) and port 31157 (`ssh -p 31157 ayy@127.0.0.1`) to verify connectivity to the IPS and Kali VMs, respectively. Apply what has been said about the port numbers and the username accordingly

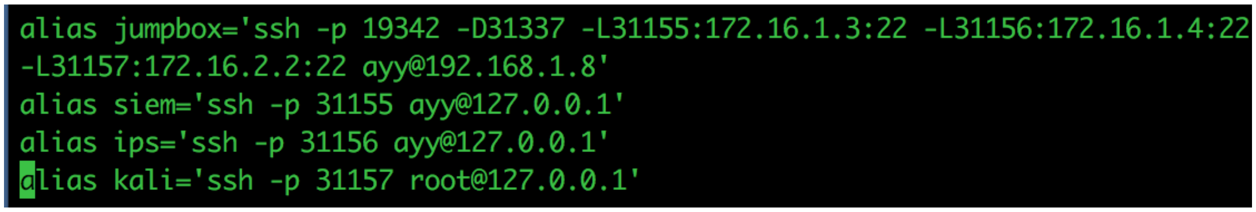


```
Tonys-MBP:~ trobinson$ ssh -p 31156 ayy@127.0.0.1
ayy@127.0.0.1's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-59-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.


Last login: Tue Jan 24 09:58:28 2017 from 192.168.1.8
ayy@ips:~$ hostname -f
ips.local
ayy@ips:~$
```



```
Tonys-MBP:~ trobinson$ ssh -p 31157 root@127.0.0.1
root@127.0.0.1's password:

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 24 10:15:05 2017 from 192.168.1.8
root@kali:~# hostname -f
kali.local
root@kali:~#
```

If you were able to reach all three VMs, consider adding aliases based on the command lines above (exemplarily depicted below), to make establishing an SSH connection with TCP tunneling/forwarding to the jump box and connecting to the VMs tunneled through the jump box easier.

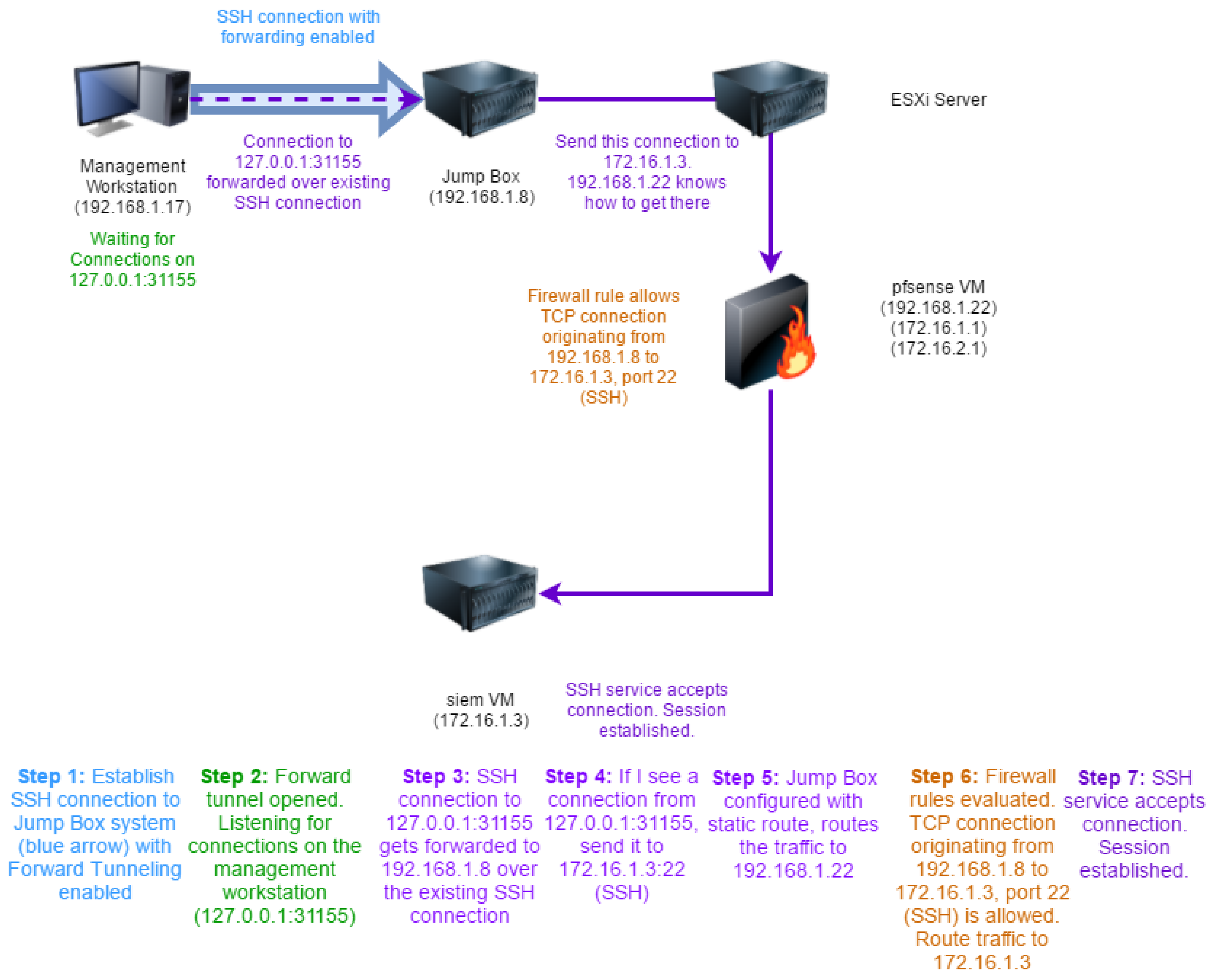


```
alias jumpbox='ssh -p 19342 -D31337 -L31155:172.16.1.3:22 -L31156:172.16.1.4:22
-L31157:172.16.2.2:22 ayy@192.168.1.8'
alias siem='ssh -p 31155 ayy@127.0.0.1'
alias ips='ssh -p 31156 ayy@127.0.0.1'
alias kali='ssh -p 31157 root@127.0.0.1'
```

Understanding SSH Tunnels

I just connected to 127.0.0.1, how am I connected to the SIEM VM? SSH forward tunnels are dark magic.

While it may be a bit hard to understand at first how they actually work, rest assured that there is absolutely no magic involved at all. Review the diagram below:


SSH connection with forwarding enabled
Management Workstation (192.168.1.17)
Waiting for Connections on 127.0.0.1:31155
Connection to 127.0.0.1:31155 forwarded over existing SSH connection
Jump Box (192.168.1.8)
Send this connection to 172.16.1.3. 192.168.1.22 knows how to get there
ESXi Server
pfsense VM (192.168.1.22) (172.16.1.1) (172.16.2.1)
Firewall rule allows TCP connection originating from 192.168.1.8 to 172.16.1.3, port 22 (SSH)
siem VM (172.16.1.3)
SSH service accepts connection. Session established.

Step 1: Establish SSH connection to Jump Box system (blue arrow) with Forward Tunneling enabled

Step 2: Forward tunnel opened. Listening for connections on the management workstation (127.0.0.1:31155)

Step 3: SSH connection to 127.0.0.1:31155 gets forwarded to 192.168.1.8 over the existing SSH connection

Step 4: If I see a connection from 127.0.0.1:31155, send it to 172.16.1.3:22 (SSH)

Step 5: Jump Box configured with static route, routes the traffic to 192.168.1.22

Step 6: Firewall rules evaluated. TCP connection originating from 192.168.1.8 to 172.16.1.3, port 22 (SSH) is allowed. Route traffic to 172.16.1.3

Step 7: SSH service accepts connection. Session established.

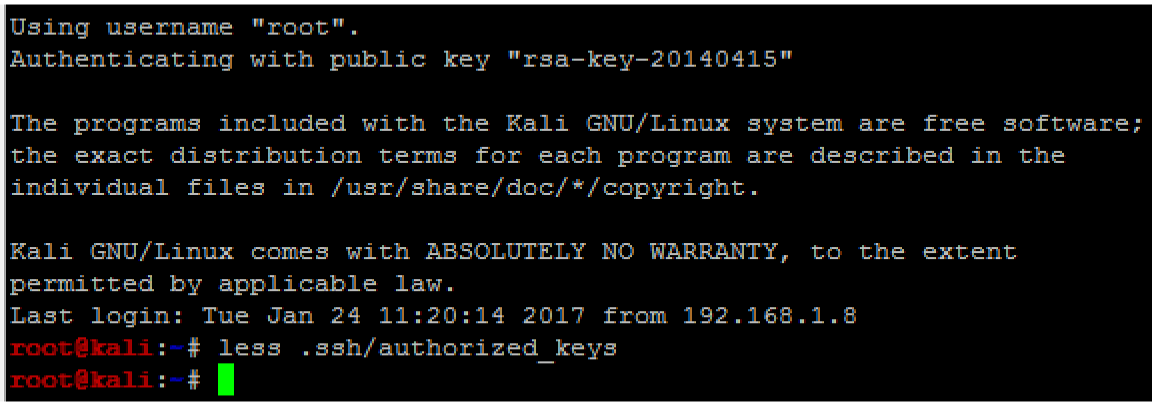
We start by connecting to the jump box system at 192.168.1.8 with your forward and dynamic tunnels configured. These tunnels open listening network ports (in our case, TCP port 31155) on your management workstation, specifically on its loopback IP address, 127.0.0.1. We then create an SSH connection to 127.0.0.1 on port 31155. The listener accepts our connection, and forwards our TCP data over our already existing SSH connection to the jump box on the other end. The jump box then forwards the connection to 172.16.1.3 on on TCP port 22 (SSH). This makes it appear as though the connection is originating from our jump box at 192.168.1.8. Since the jump box has static routes to the 172.16.1.0/24 network via 192.168.1.22, and we have a firewall rule that allows 192.168.1.8 to connect to 172.16.1.3 on port 22/TCP, the connection is forwarded to the SIEM VM and allowed, granting us a successful connection.

Closing Notes on Jump Boxing

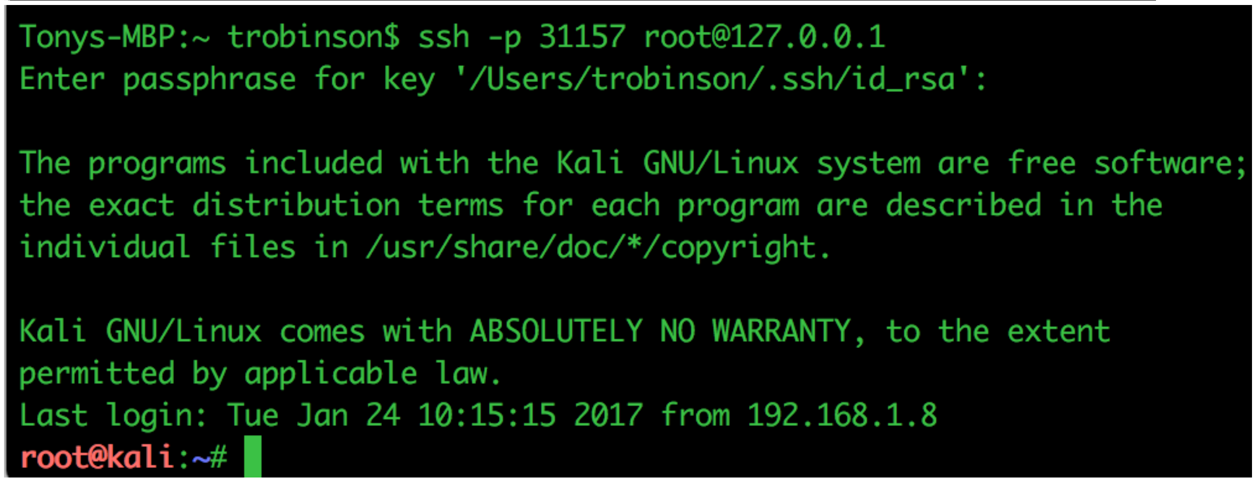
Let’s talk about key-based authentication, and managing your SSH keys with jump boxes/bastion hosts.

Key-Based Authentication: Managing SSH Keys for Tunneled Connections

Just like with hosted hypervisors, you can configure key-based authentication for accessing the jump box and/or your tunneled SSH sessions to the lab VMs. Key-based authentication to the lab virtual machines through SSH tunnels works perfectly fine, regardless of whether you are using Windows (PuTTY/mRemoteNG) or Linux/OS X/BSD.



```
Using username "root".
Authenticating with public key "rsa-key-20140415"

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 24 11:20:14 2017 from 192.168.1.8
root@kali:~# less .ssh/authorized_keys
root@kali:~#
```



```
Tonys-MBP:~ trobinson$ ssh -p 31157 root@127.0.0.1
Enter passphrase for key '/Users/trobinson/.ssh/id_rsa':

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 24 10:15:15 2017 from 192.168.1.8
root@kali:~#
```

If you want to establish key-based authentication, simply follow the Windows Remote Access or Linux/OS X/BSD Remote Access guides, specifically, the portions on SSH key generation, and copying your SSH public key, generated on your management workstation, to the jumpbox and lab VMs. You can use key-based authentication for both the jump box, as well as your connections to the lab VMs through the jumpbox with no problems whatsoever.

Note: Depending on whether or not you value security or redundancy, you may wish to generate an SSH key on your jump box, and copy the public key of the respective user of this system to the Kali, SIEM, and IPS VM as well. It is considered good security practice to NOT generate or store SSH private keys on a system that is meant to act as a bastion host (e.g. you should not run `ssh-keygen` on your jump box system, and copy its public keys to the lab virtual machines). Instead it is recommend to use only the public/private SSH key pair on your local management workstation instead. This practice reduces the possibility of attackers gaining access to poorly secured SSH keys on a jump box/bastion host system, and using that jump box as a pivot to compromise other hosts.

On the other hand, having the ability to directly establish SSH connections to your lab VMs from an SSH session on your jump box system provides some degree of redundancy in the event you aren't able to get SSH tunnels or TCP forwarding to work as expected. A compromise between these two would be to password protect any SSH keys you generate on the jump box system via `ssh-keygen` before copying the jump box public SSH key to your lab VMs, if you desire redundant access to your lab virtual machines.

In addition to enabling key-based authentication, you can also choose to disable password authentication for SSH connections to the jumpbox and lab VMs (which I very highly recommend), and/or enable key-based authentication as root on both the jump box and lab VMs, if you're so inclined. Typically, I don't enable key-based authentication as root on the bastion host system because TCP forwarding (which allows us to establish tunnels to the lab VMs) doesn't require root privileges, and it is a tenant of good security to only use the privileges that are necessary to perform functions you require.

IPS Installation Guide

If you have worked through the previous parts of this guide, you should have a mostly functional lab environment by now, consisting of five essentially usable virtual machines. Furthermore, there should be firewall rules in place that, among other things, basically enable you to remotely manage those VMs.This installation guide pertains primarily to the IPS VM.

Before getting into the matter of this section, I recommend that you complete the Remote Lab Management chapter. Enabling SSH and SCP access for the IPS virtual machine will make system management and configuration much easier for you. Apart from this recommendation, your hypervisor network and/or the IPS VM (depending on the hypervisor you chose), ***must be configured to allow promiscuous mode***, and in some cases (such as with Client Hyper-V, or ESXi), must be set to permit MAC spoofing/forged transmits. If you followed the instructions laid out in the corresponding setup guide for your hypervisor of choice, you should be all set. With that being said, let's turn to the actual installation.

Currently the most popular rule-/signature-based IDS/IPS solutions are Snort, (http://www.snort.org) and Suricata (http://www.suricata-ids.org). To avoid the impression of favoring the one over the other (in spite of having used Snort most of my career), I'm going to instruct you on how to install both solutions. Both Snort and Suricata can be used to supply an AFPACKET, fail-close bridge, which makes them both viable choices for our lab network. Ultimately, both systems are expected to operate in almost the same fashion, with the only difference being the default rules provided by the solutions.

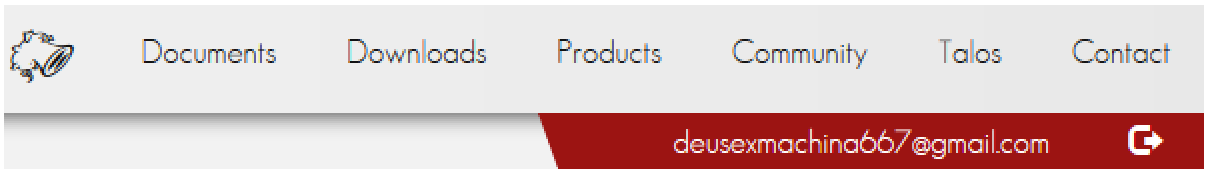
Installing and configuring Snort (via Autosnort)

In the past, installing Snort, its supporting libraries, and rule management tools was a pretty daunting task that would take a significant amount of time. Fortunately for you, this is something I have automated for your convenience. There is a small project of mine on GitHub called "Autosnort" (I'm not one for particularly creative names) that has evolved over the years . The scripts (multiple versions to address different Linux distributions) are designed to install the prerequisites needed to successfully compile and run Snort, including the DAQ (Data AcQuisition) libraries, the PulledPork rule manager, and the alert spooler Barnyard2. Additionally, my scripts also provide the user with the option of configuring different output interfaces and mechanisms (e.g. send alerts to syslog, utilize the sguil/squert alert interfaces, etc.).

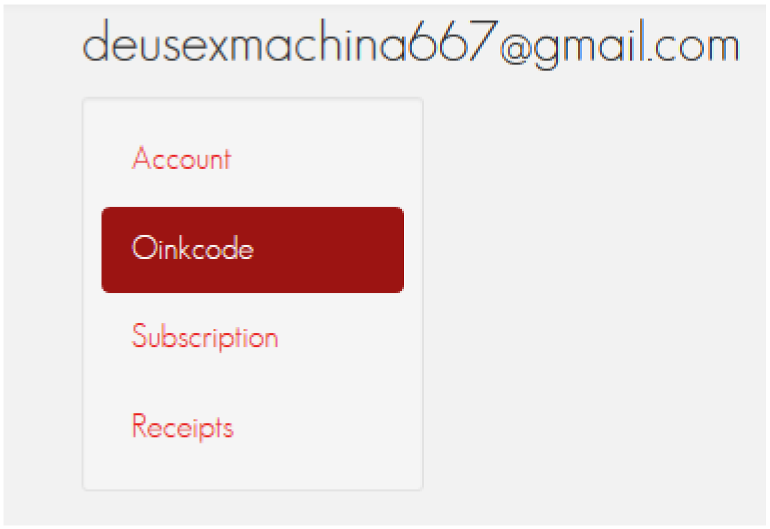
I made a special edition version of Autosnort for this book, which implements some important changes compared to the standard versions of Autosnort. First and foremost, it's very stripped down. This new, special version only installs Snort and PulledPork. Additionally, the script assumes that we will be running our sensor in what is called inline mode, using AFPACKET, which is exactly what we want (and need) for passing traffic between the IPS 1 and IPS 2 networks. Finally, this adapted Autosnort script does NOT install Barnyard2 or any kind of an output interface for viewing alerts. Instead, we are going to set up Splunk and forward our IPS logs to our Splunk server for review and storage.

To get started, visit snort.org and register a user account. If you don't want to use your real email address, you can probably use 10 Minute Mail or another service that offers disposable email addresses to create your account. Be sure to save the credentials you used for this registration.

After logging in, click on your e-mail address in the upper right corner. This will bring you to the settings page of your account.



On the left side of this page, underneath your email address, click on Oinkcode to have your license key generated and displayed to you in the center of the screen. Copy the string of letters and numbers (you'll need it momentarily), then you may logout of your Snort.org account.



After getting your Oinkcode, log in to your IPS VM with the respective account you created during the initial installation. To do so, use either the virtual machine's console via your hypervisor's management interface, or, if you did the Remote Lab Management configuration exercises, connect to the VM over SSH,. Next, at the session's command prompt, run

```
sudo su -
```

and put in your user’s password to get a `root` shell. Autosnortis going to muck with a lot of system settings, **running the script as `root` is a requirement.**

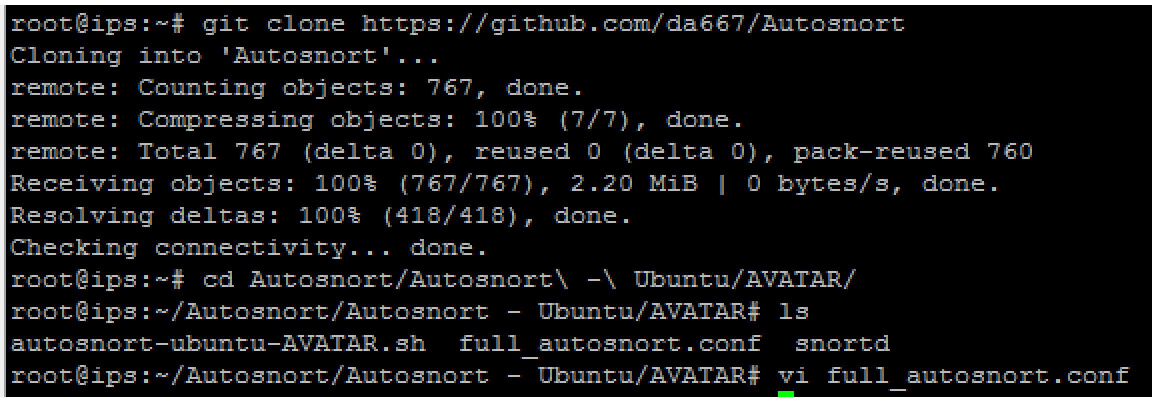
Note: If you installed and configured the Squid HTTP proxy as per the Network Configuration - Core Network Services guide, you need to enter and execute the following lines BEFORE running any of the commands or scripts below:

```
export HTTP_PROXY=http://172.16.1.1:3128
export http_proxy=http://172.16.1.1:3128
export HTTPS_PROXY=
export https_proxy=
```

The commands above set the HTTP_PROXY, http_proxy, HTTPS_PROXY, and https_proxy environment variables for the current user session. They need to be present and set for command line utilities to be aware of the fact that a proxy server is in place and has to be used to gain external network access for the protocols specified. If you don’t set these variables, you may notice that the `git` command as well as the Autosnort script hangs (and times out eventually) as soon as a download is attempted.

Once you opened the `root` shell, get the Autosnort script by running the command

```
git clone https://github.com/da667/Autosnort
```



```
root@ips:~# git clone https://github.com/da667/Autosnort
Cloning into 'Autosnort'...
remote: Counting objects: 767, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 767 (delta 0), reused 0 (delta 0), pack-reused 760
Receiving objects: 100% (767/767), 2.20 MiB | 0 bytes/s, done.
Resolving deltas: 100% (418/418), done.
Checking connectivity... done.
root@ips:~# cd Autosnort/Autosnort\ -\ Ubuntu/AVATAR/
root@ips:~/Autosnort/Autosnort - Ubuntu/AVATAR# ls
autosnort-ubuntu-AVATAR.sh  full_autosnort.conf  snortd
root@ips:~/Autosnort/Autosnort - Ubuntu/AVATAR# vi full_autosnort.conf
```

If for some reason, you do not have the `git` command available (it should have been installed by default), run

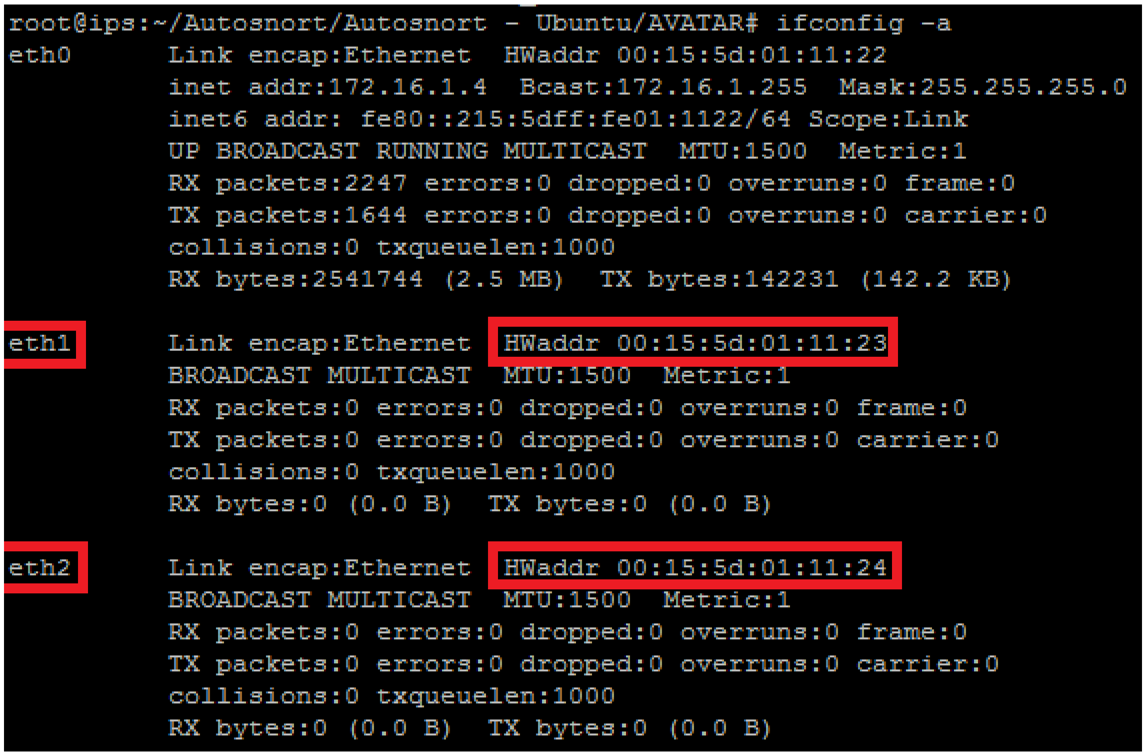
```
apt-get -y install git
```

to install it. After downloading Autosnort, run

```
ifconfig -a
```

to identify which interfaces are available for use on your system. If you followed the respective section in the setup guide for the hypervisor of your choice, your IPS VM should be configured with three virtual network cards. Each of these NICs should be connected to a certain virtual network, and it is crucial to clearly identify these relations, as we need this information to make the Autosnort script work properly.

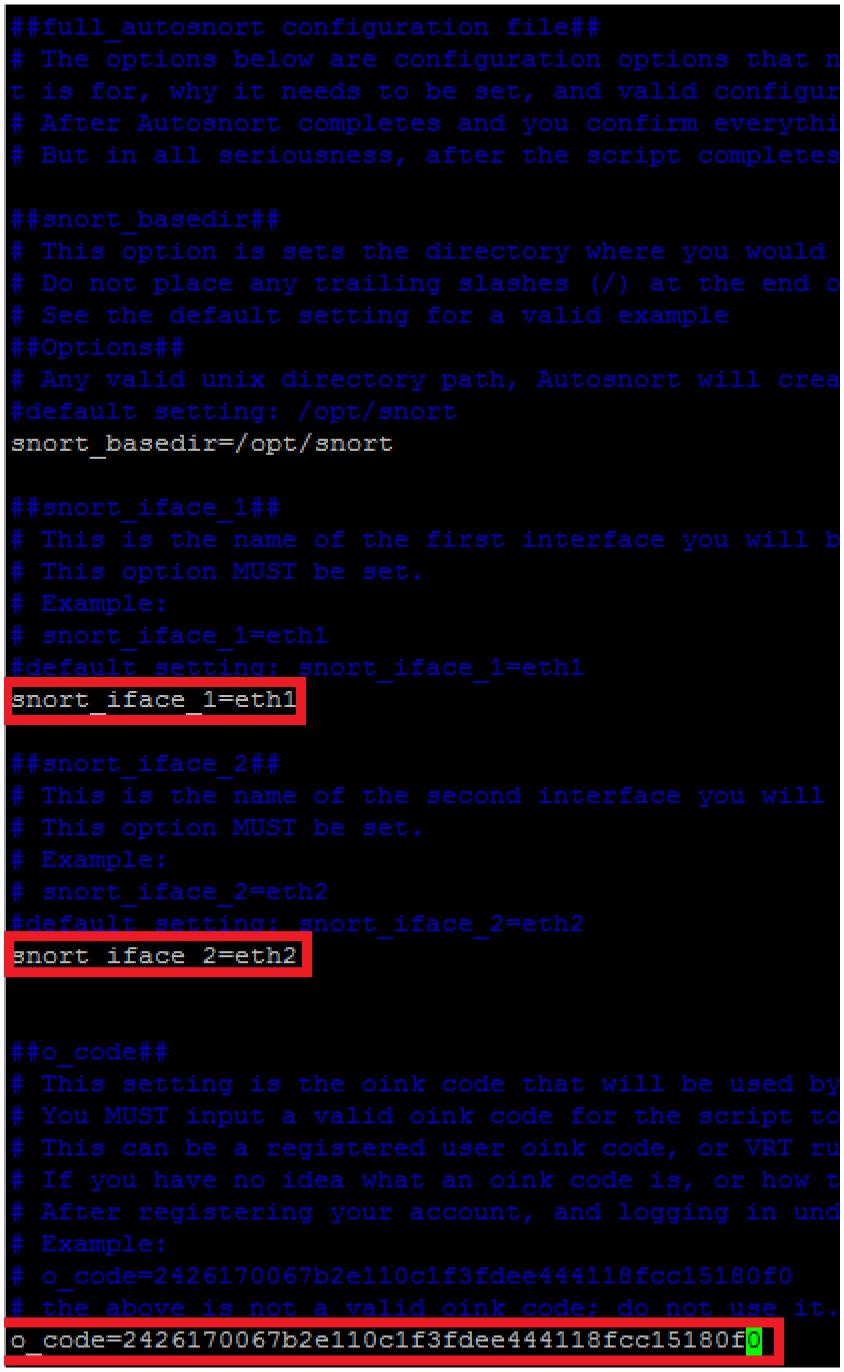
As you can see in the picture below, the network interface I used for establishing an SSH connection to the IPS VM was `eth0` (easily identified by the IP address used/assigned). My virtual machine also had `eth1` and `eth2` available. `Eth0` should be connected to the *Management* network, while `eth1` and `eth2` should be connected to the *IPS1* and *IPS2* networks, respectively. Check the `HWaddr` field for each of the NICs and compare their values with those of the MAC addresses in your notes for the IPS VM setup to confirm which interface is connected to which virtual network. We will be using this information momentarily.



```
root@ips:~/Autosnort/Autosnort - Ubuntu/AVATAR# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:15:5d:01:11:22
          inet addr:172.16.1.4  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: fe80::215:5dff:fe01:1122/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2247 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1644 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2541744 (2.5 MB)  TX bytes:142231 (142.2 KB)

eth1      Link encap:Ethernet  HWaddr 00:15:5d:01:11:23
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth2      Link encap:Ethernet  HWaddr 00:15:5d:01:11:24
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Next, change directories into `Autosnort/Autosnort - Ubuntu/AVATAR`. In this folder, there should be four files, with one of them named `full_autosnort.conf`. We need to edit this file and change a couple of settings before the Autosnort script itself is run. Using the text editor you are most comfortable with, adjust the following settings:

- `snort_iface_1`: by default, this is set to `eth1`, the second listed network interface. In my case, I didn't have to change this.
- `snort_iface_2`: by default, this is set to `eth2`, the third listed network interface. Again, in my case, I did not have to change this.
- `o_code`: by default this is blank. This line should contain the Oinkcode you copied from snort.org earlier. Make sure that there are **no blank spaces in this string**.



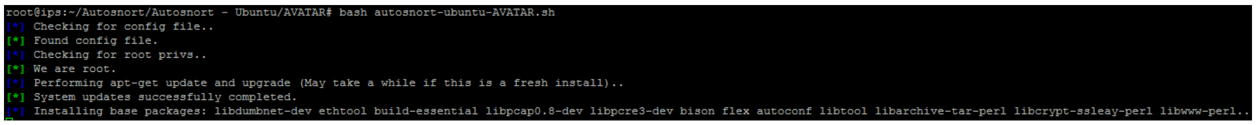
```
##full_autosnort configuration file##
# The options below are configuration options that n
t is for, why it needs to be set, and valid configur
# After Autosnort completes and you confirm everythi
# But in all seriousness, after the script completes

##snort_basedir##
# This option is sets the directory where you would
# Do not place any trailing slashes (/) at the end o
# See the default setting for a valid example
##Options##
# Any valid unix directory path, Autosnort will crea
#default setting: /opt/snort
snort_basedir=/opt/snort

##snort_iface_1##
# This is the name of the first interface you will b
# This option MUST be set.
# Example:
# snort_iface_1=eth1
#default setting: snort_iface_1=eth1
snort_iface_1=eth1

##snort_iface_2##
# This is the name of the second interface you will
# This option MUST be set.
# Example:
# snort_iface_2=eth2
#default setting: snort_iface_2=eth2
snort_iface_2=eth2


##o_code##
# This setting is the oink code that will be used by
# You MUST input a valid oink code for the script to
# This can be a registered user oink code, or VRT ru
# If you have no idea what an oink code is, or how t
# After registering your account, and logging in und
# Example:
# o_code=2426170067b2e110c1f3fdee444118fcc15180f0
# the above is not a valid oink code; do not use it.
o_code=2426170067b2e110c1f3fdee444118fcc15180f0
```

After you are done editing, save the file and exit the editor, then enter and execute the following

```
bash autosnort-ubuntu-AVATAR.sh
```

to begin running the Autosnort installer.

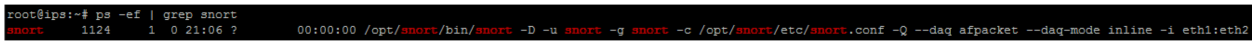
The script keeps you notified as to what tasks it is running at any given time. (You can see an example of the output it provides in the image below.) Please be patient and let it run its course. When the script has finished running successfully, your IPS VM will reboot automatically to finish the installation.



```
root@ips:~/Autosnort/Autosnort - Ubuntu/AVATAR# bash autosnort-ubuntu-AVATAR.sh
[*] Checking for config file..
[*] Found config file.
[*] Checking for root privs..
[*] We are root.
[*] Performing apt-get update and upgrade (May take a while if this is a fresh install)..
[*] System updates successfully completed.
[*] Installing base packages: libdumbnet-dev ethtool build-essential libpcap0.8-dev libpcre3-dev bison flex autoconf libtool libarchive-tar-perl libcrypt-ssleay-perl libwww-perl..
```

If the script fails for some reason, the log file `/var/log/autosnort_install.log` should contain clues as to what went wrong. After identifying and applying the fix needed, you can try to run the script again. Alternatively, revert the virtual machine to a previous snapshot, resolve the problem, snapshot the VM again, then attempt to run the script again. Upon successful completion, the script will reboot the virtual machine. As soon as it is back up and running again, log back into the IPS VM and execute the command line

```
ps -ef | grep snort
```



```
root@ips:~# ps -ef | grep snort
snort     1124     1  0 21:06 ?        00:00:00 /opt/snort/bin/snort -D -u snort -g snort -c /opt/snort/etc/snort.conf -Q --daq afpacket --daq-mode inline -i eth1:eth2
```

The output above shows that Snort is running out of `/opt/snort/bin` with the options:

- `-D` (daemonize),
- `-u snort`, `-g snort` (user and group to drop privileges to),
- `-c /opt/snort/etc/snort.conf` (points to the configuration file to use),
- `-Q` (inline mode),
- `--daq afpacket` (tells snort what DAQ module to use),
- `--daq-mode inline` (tells snort to run our DAQ module with support for inline mode),
- `-i eth1:eth2` (the interface pair to use for running snort inline).

If the output you received on running the command line above looks like this, then Snort should be running and passing traffic just fine. If you need to restart/reload it for any reason (e.g. adding/removing rules, changing configuration options, etc), the init script `snortd` exists and supports the stop, start, restart, and status options for managing the `snort` process. If you want to confirm that Snort is running properly, proceed to the section on testing your IPS bridge.

Installing and configuring Suricata (via Autosuricata)

Not unlike automating the installation and configuration of Snort, I created a script named Autosuricata (again, I'm not terribly creative with names) that prepares Suricata in a similar way. Just like Autosnort, it needs to be run with `root` privileges, and leaves the user with a minimal Suricata installation. Unlike Autosnort, Autosuricata does NOT require you to have an Oinkcode, unless you have an Emerging Threats Pro rule subscription that you would like to utilize for your lab environment.

Log in to your IPS VM with the respective account you created during the initial installation. To do so, use either the virtual machine's console via your hypervisor's management interface, or, if you did the Remote Lab Management configuration exercises, connect to the VM over SSH. Next, at the session's command prompt, run

```
sudo su -
```

and put in your user's password, to get a root shell. As Autosuricata is going to muck with a lot of system settings, running the script as root is a requirement.

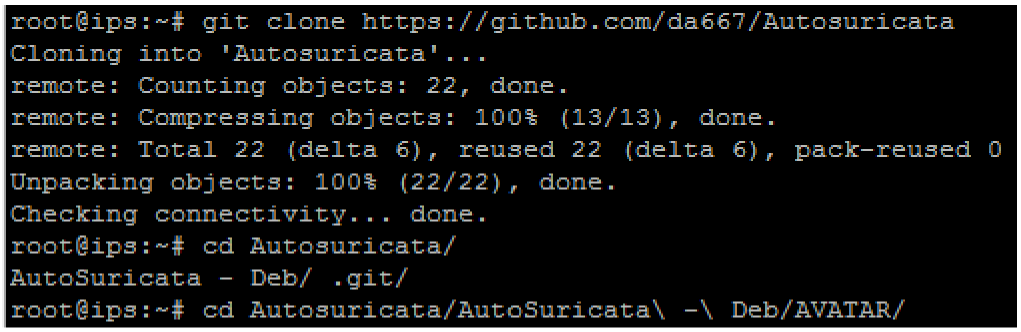
Note: If you installed and configured the Squid HTTP proxy as per the Network Configuration - Core Network Services guide, you need to enter and execute the following lines BEFORE running any of the commands or scripts below:

```
export HTTP_PROXY=http://172.16.1.1:3128
export http_proxy=http://172.16.1.1:3128
export HTTPS_PROXY=
export https_proxy=
```

The commands above set the http_proxy, HTTP_PROXY, HTTPS_PROXY, and https_proxy variables for the current user session. They need to be present and set for command line utilities to be aware of the fact that a proxy server is in place and has to be used to gain external network access for the protocols specified. If you don't set these variables, you will notice that the `git` command as well as the Autosuricata script hangs (and times out eventually) as soon as a download is attempted.

Once you opened the `root` shell, get the Autosuricata script by entering and running the command

```
git clone https://github.com/da667/Autosuricata
```



```
root@ips:~# git clone https://github.com/da667/Autosuricata
Cloning into 'Autosuricata'...
remote: Counting objects: 22, done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 22 (delta 6), reused 22 (delta 6), pack-reused 0
Unpacking objects: 100% (22/22), done.
Checking connectivity... done.
root@ips:~# cd Autosuricata/
AutoSuricata - Deb/ .git/
root@ips:~# cd Autosuricata/AutoSuricata\ -\ Deb/AVATAR/
```

If for some reason, you do not have the `git` command available (it should have been installed by default), run

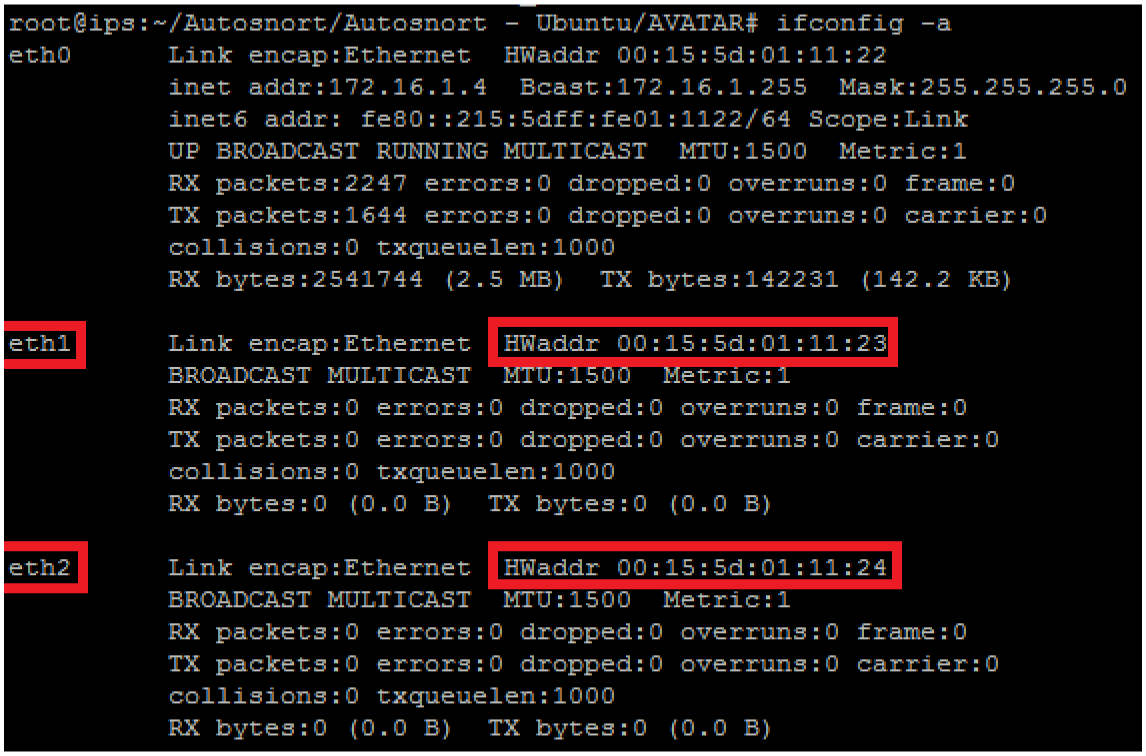
```
apt-get -y install git
```

to install it. After downloading `Autosuricata`, run

```
ifconfig -a
```

to identify which interfaces are available for use on your system. If you followed the respective section in the setup guide for the hypervisor of your choice, your IPS VM should be configured with three virtual network cards. Each of these NICs should be connected to a certain virtual network, and it is crucial to clearly identify these relations, as we need this information to make the Autosuricata script work properly.

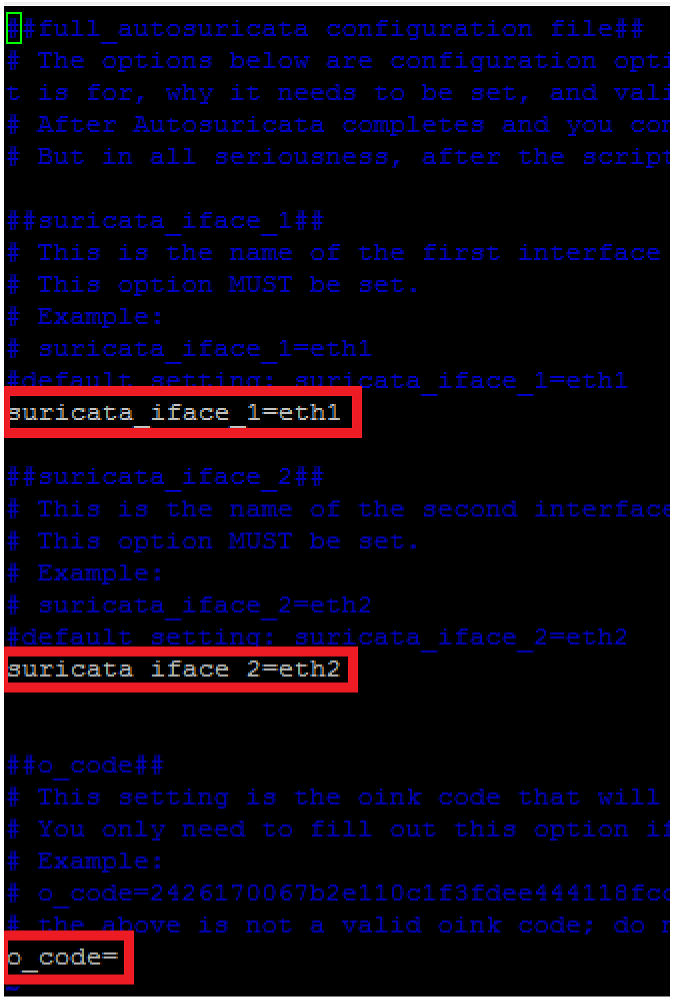
As you can see in the picture below, the network interface I used for establishing an SSH connection was `eth0` (easily identified by the IP address used/assigned). My virtual machine also had `eth1`, and `eth2`. `Eth0` should be connected to the *Management* network, while `eth1` and `eth2` should be linked to the *IPS1* and *IPS2* networks, respectively. Check the `HWaddr` field for each of the NICs and compare their values with those of the MAC addresses in your notes for the IPS VM setup to confirm which network interface is connected to which virtual network. We will be using this information momentarily.



```
root@ips:~/Autosnort/Autosnort - Ubuntu/AVATAR# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:15:5d:01:11:22
          inet addr:172.16.1.4  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: fe80::215:5dff:fe01:1122/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2247 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1644 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2541744 (2.5 MB)  TX bytes:142231 (142.2 KB)

eth1      Link encap:Ethernet  HWaddr 00:15:5d:01:11:23
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth2      Link encap:Ethernet  HWaddr 00:15:5d:01:11:24
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Next, change directories into `Autosuricata/AutoSuricata - Deb/AVATAR`. In this folder, there should be five files,with one of them named `full_autosuricata.conf`. We need to edit this file and change a couple of settings before the Autosuricata script itself is run. Using the text editor you are most comfortable with, adjust the following settings:

- `suricata_iface_1`: by default, this is set to `eth1`, the second listed network interface. In my case, I didn't have to change this.
- `suricata_iface_2`: by default, this is set to `eth2`, the third listed network interface. Again, in my case, I did not have to change this.
- `o_code`: by default this is blank. This field is optional. You only need to enter a valid Oinkcode if you are using the Emerging Threats Pro ruleset.



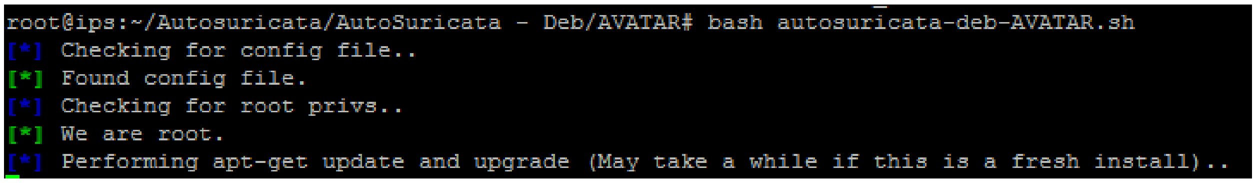
```
##full_autosuricata configuration file##
# The options below are configuration opti
t is for, why it needs to be set, and vali
# After Autosuricata completes and you cor
# But in all seriousness, after the script

##suricata_iface_1##
# This is the name of the first interface
# This option MUST be set.
# Example:
# suricata_iface_1=eth1
#default setting: suricata_iface_1=eth1
suricata_iface_1=eth1

##suricata_iface_2##
# This is the name of the second interface
# This option MUST be set.
# Example:
# suricata_iface_2=eth2
#default setting: suricata_iface_2=eth2
suricata_iface_2=eth2


##o_code##
# This setting is the oink code that will
# You only need to fill out this option if
# Example:
# o_code=2426170067b2e110c1f3fdee444118fcc
# the above is not a valid oink code; do n
o_code=
```

After you are done editing, save the file and exit the editor, then enter and execute the following

```
bash autosuricata-deb-AVATAR.sh
```

to begin running the Autosuricata installer.

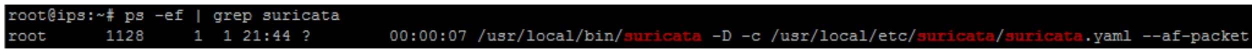
The script keeps you notified as to what tasks it is running at any given time. (You can see an example of the output it provides in the image below.) Please be patient and let it run its course. When the script is finished running successfully, your VM will reboot.



```
root@ips:~/Autosuricata/AutoSuricata - Deb/AVATAR# bash autosuricata-deb-AVATAR.sh
[*] Checking for config file..
[*] Found config file.
[*] Checking for root privs..
[*] We are root.
[*] Performing apt-get update and upgrade (May take a while if this is a fresh install)..
```

If the script fails for some reason, the log file `/var/log/autosuricata_install.log` should contain clues as to what went wrong. After identifying and applying the fix needed, you can try to run the script again. Alternatively, revert the virtual machine to a previous snapshot, resolve the problem, snapshot the VM again, then attempt to run the script again. Upon successful completion, the script will automatically reboot the virtual machine. As soon as it is back up and running again, log back into the IPS VM and run the command

```
ps -ef | grep suricata
```



```
root@ips:~# ps -ef | grep suricata
root      1128     1  1 21:44 ?        00:00:07 /usr/local/bin/suricata -D -c /usr/local/etc/suricata/suricata.yaml --af-packet
```

The output above shows that Suricata is running out of `/usr/local/bin` with the options:

- `-D` (daemonize),
- `-c /usr/local/etc/suricata/suricata.yaml` (points to the configuration file to use),
- `--af-packet` (instructs Suricata to operate in AFPACKET mode).

Since the Autosuricata script created an `af-packet.yaml` file in `/usr/local/etc/suricata/af-packet.yaml`, and, in addition to that, there is an "include:" statement pointing to this file in `suricata.yaml`, Suricata is configured to use af-packet bridging between `eth1` and `eth2` (via the `full_autosuricata.conf` file).

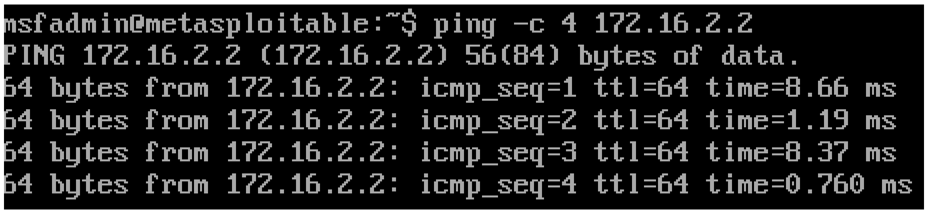
If the output you received on running the command line above looks like this, then Suricata should be running and passing traffic just fine. If you need to restart/reload it for any reason (e.g. adding/removing rules, changing configuration options, etc), the init script `suricatad` exists and supports the stop, start, restart, and status options for managing the `suricata` process. If you want to confirm that Suricata is running properly, proceed to the section on testing your IPS bridge below.

Testing your IPS Bridge

After installing either the Snort or Suricata IPS, we need to test the AFPACKET, fail-close bridge that should be in place now. To do so, start by powering on your Kali VM and your Metasploitable2 VM. Remember that Kali is located on the *IPS1* network, with an IP address of 172.16.2.2, while Metasploitable 2 is connected to the *IPS2* network, with 172.16.2.3 as its corresponding static DHCP mapping, which we haven't been able to verify yet.

As soon as both VMs have finished booting, log in to the Metasploitable2 virtual machine via the default username and password of `msfadmin`, then run:

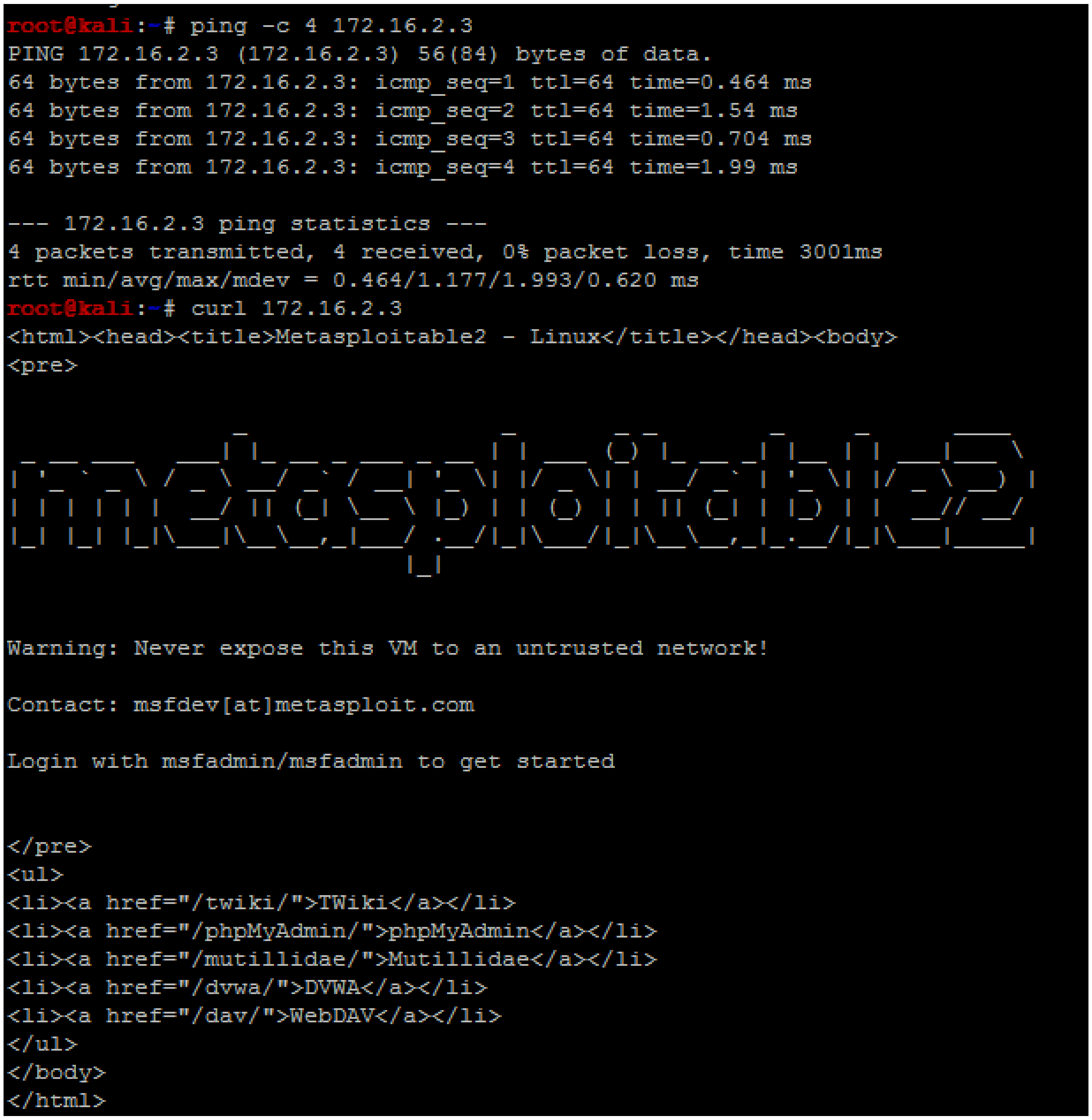
```
ping -c 4 172.16.2.2
```



```
msfadmin@metasploitable:~$ ping -c 4 172.16.2.2
PING 172.16.2.2 (172.16.2.2) 56(84) bytes of data.
64 bytes from 172.16.2.2: icmp_seq=1 ttl=64 time=8.66 ms
64 bytes from 172.16.2.2: icmp_seq=2 ttl=64 time=1.19 ms
64 bytes from 172.16.2.2: icmp_seq=3 ttl=64 time=8.37 ms
64 bytes from 172.16.2.2: icmp_seq=4 ttl=64 time=0.760 ms
```

If you get results similar to the output depicted above, that means the Metasploitable2 VM on the *IPS2* network is able to reach the Kali VM on the *IPS1* network. Next, log in to the Kali VM and try the following:

```
ping -c 4 172.16.2.3
curl 172.16.2.3
```



```
root@kali:~# ping -c 4 172.16.2.3
PING 172.16.2.3 (172.16.2.3) 56(84) bytes of data.
64 bytes from 172.16.2.3: icmp_seq=1 ttl=64 time=0.464 ms
64 bytes from 172.16.2.3: icmp_seq=2 ttl=64 time=1.54 ms
64 bytes from 172.16.2.3: icmp_seq=3 ttl=64 time=0.704 ms
64 bytes from 172.16.2.3: icmp_seq=4 ttl=64 time=1.99 ms

--- 172.16.2.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.464/1.177/1.993/0.620 ms
root@kali:~# curl 172.16.2.3
<html><head><title>Metasploitable2 - Linux</title></head><body>
<pre>

                _                  _       _ _        _     _      ____
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


</pre>
<ul>
<li><a href="/twiki/">TWiki</a></li>
<li><a href="/phpMyAdmin/">phpMyAdmin</a></li>
<li><a href="/mutillidae/">Mutillidae</a></li>
<li><a href="/dvwa/">DVWA</a></li>
<li><a href="/dav/">WebDAV</a></li>
</ul>
</body>
</html>
```

If the output you recieve is similar to the illustration above, then congratulations: your AFPACKET bridge and the virtual networking (including the promiscuous mode and/or forged transmit settings) are all set up correctly, and the Kali Linux VM on the *IPS1* network can reach Metasploitable2 on the *IPS2* network.

At this point, I very highly recommend creating another snapshot of the IPS VM, since, as of now, it is running with a known good working configuration.

Splunk Installation Guide

Splunk is the SIEM (which is shorthand for Security Intrusion Events Manager) software that we are going to install on the SIEM VM for our lab. While there are a variety of open-source projects/solutions available (e.g. ELK -- Elastic Search, Logstash, Kibana), some of the considerable advantages of Splunk are, that it is dead-easy to set up, has extremely robust extension/app support, a large user community, and it will more than fit the bill for our small lab environment.

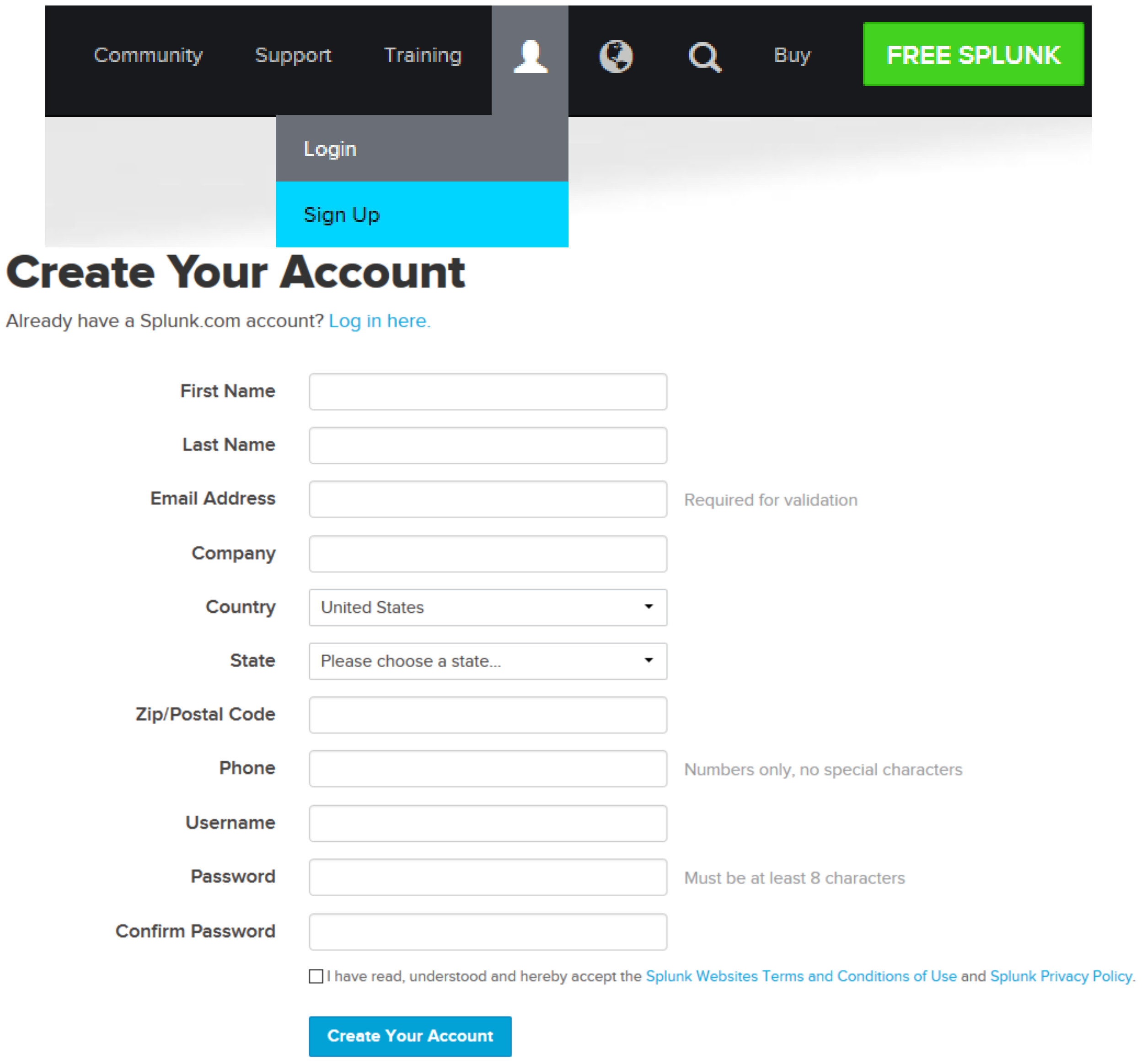
Splunk isn't truly a SIEM in and of itself, but a log aggregation system that, through a huge log parsing ecosystem, has been beaten into submission.

The one concern most critics of Splunk have is the limitation on the amount of data that can be logged/used on a daily basis. By default, the free version allows to collect up to 500mb of logs per day. Anything beyond that requires a license, and commercial licensing for Splunk tends to be very pricey. However, there is a developer program available, and after registering with Splunk, you can request access to a development license that allows you up to collect/process 10GB of data per day. This license is good for one year, and available for absolutely no cost to you. Having said this, it is VERY important to mention that this license is EXCLUSIVELY meant for training purposes, the development of tools, and personal skill improvement, and thus should NOT be used in a production environment, due to possible legal and ethical concerns. The bottom line is that you should make use of the development license for training if necessary, but **do not abuse the privilege.**

Depending on the size of the infrastructure that has to be overseen, Splunk can consist of several components, distributed across different systems. these components include the search head (the user interface used to query collected logs), multiple indexers (responsible for organizing and storing collected logs), and a variety of forwarders (universal forwarders, heavy forwarders, syslog servers, etc., used to collect logs off the systems we wish to monitor). Since we have a small lab, our configuration will consist of a single Splunk enterprise installation, with both the index and search head installed on the same system (our SIEM VM), and a single universal forwarder installed on our IPS VM, with either the Snort or Suricata log parsing applications.

Initial Setup (Server Installation)

Start by visiting www.splunk.com. Find the icon that looks like a person on the right of the top menu bar and hover the mouse pointer over it. From the drop-down list that appears, select “Sign Up” to go to the registration page. If you’re paranoid, you can choose to fill in fake information for most of the fields, however, you need to associate your account with a somewhat permanent e-mail address (especially if you plan on using a developer license).

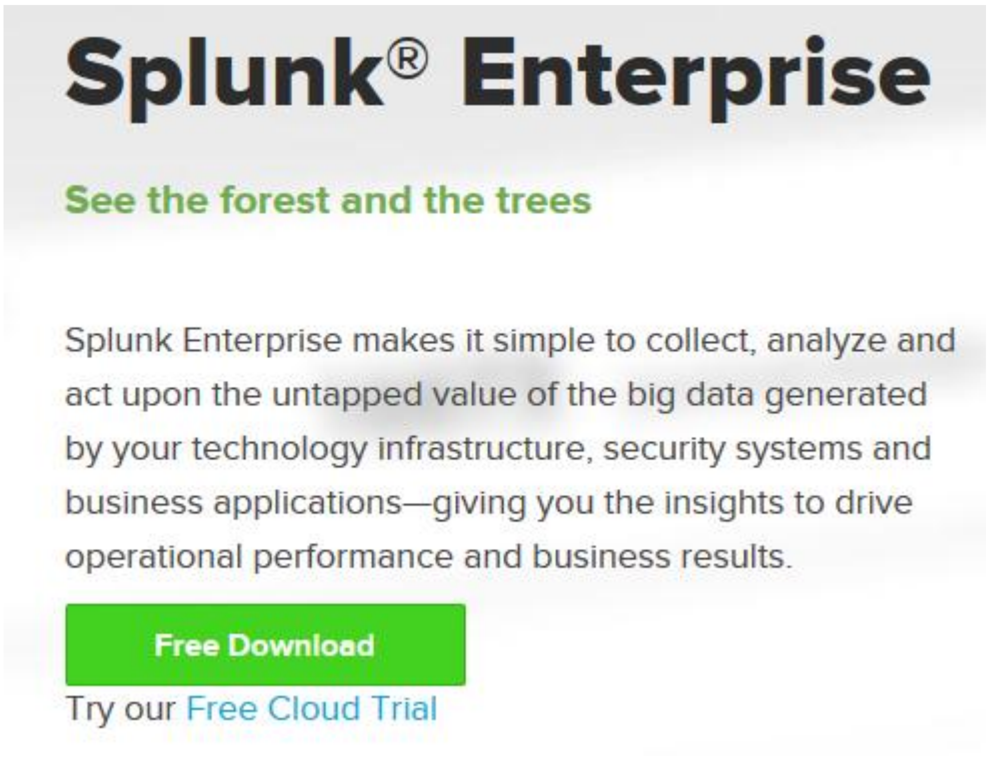


Already have a Splunk.com account? Log in here.

First Name		
Last Name		
Email Address		Required for validation
Company		
Country	United States	
State	Please choose a state...	
Zip/Postal Code		
Phone		Numbers only, no special characters
Username		
Password		Must be at least 8 characters
Confirm Password		

☐ I have read, understood and hereby accept the Splunk Websites Terms and Conditions of Use and Splunk Privacy Policy.

Once you have signed up and logged in, navigate to *Products* > *Core Products* > *Splunk Enterprise*. Click the *Free Download* button.

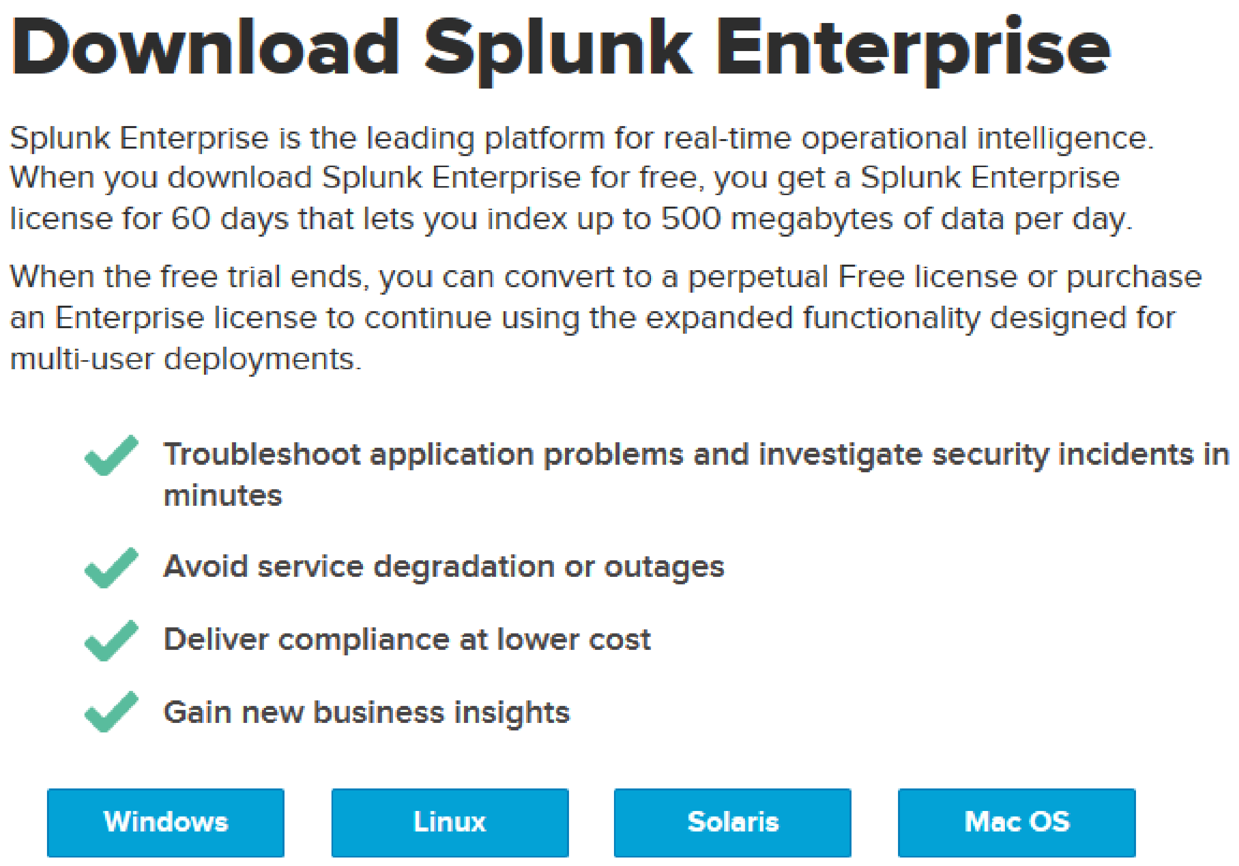


Splunk® Enterprise

See the forest and the trees

Splunk Enterprise makes it simple to collect, analyze and act upon the untapped value of the big data generated by your technology infrastructure, security systems and business applications—giving you the insights to drive operational performance and business results.

On the next page, you are offered download options for a couple of operating systems. Find the button labeled *Linux*, and click on it..



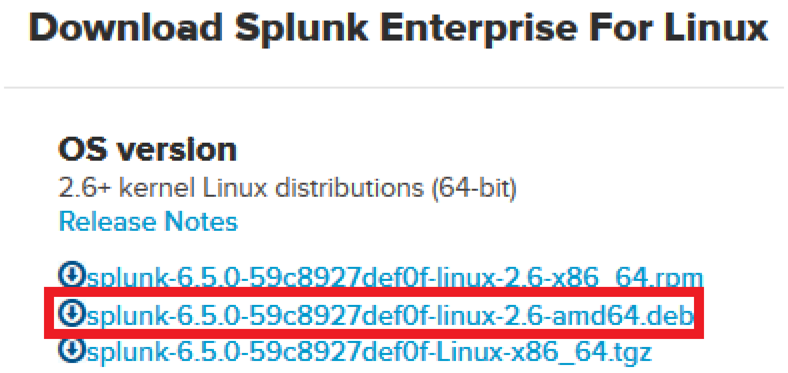
Download Splunk Enterprise

Splunk Enterprise is the leading platform for real-time operational intelligence. When you download Splunk Enterprise for free, you get a Splunk Enterprise license for 60 days that lets you index up to 500 megabytes of data per day.

When the free trial ends, you can convert to a perpetual Free license or purchase an Enterprise license to continue using the expanded functionality designed for multi-user deployments.

- ✓ Troubleshoot application problems and investigate security incidents in minutes
- ✓ Avoid service degradation or outages
- ✓ Deliver compliance at lower cost
- ✓ Gain new business insights

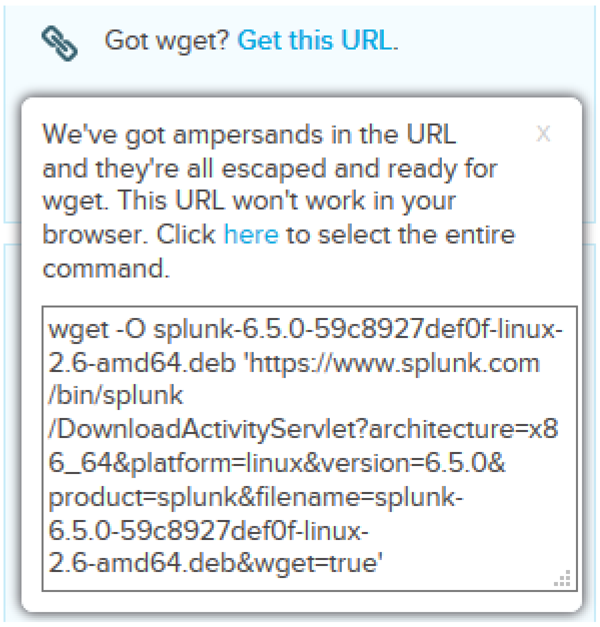
A pop-up appears, asking you to select which file to download. At the time of this writing, the current version of Splunk Enterprise is version 6.5.0. Of course, this is likely to change as new updates are made available. Make sure you download the newest release available. Click on the link that shows a filename with a `.deb` extension.


Download Splunk Enterprise For Linux

OS version
2.6+ kernel Linux distributions (64-bit)
Release Notes

splunk-6.5.0-59c8927def0f-linux-2.6-x86_64.rpm
splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb
splunk-6.5.0-59c8927def0f-Linux-x86_64.tgz

Clicking on the `.deb` link sends you to a splash page that thanks you for downloading Splunk. Your web browser should open a dialogue box, asking you where you want to save this file. If you choose to download it in this manner now, the file will be stored locally on the machine you are currently browsing the Splunk site with. As a consequence, you will have to copy this download to your SIEM VM via SCP. Alternatively, click cancel on the download file dialogue. On the splash page thanking you for downloading Splunk, look for the text that says *Got wget? Get this URL*. If you click on it, a text box pops up (similar to the one depicted below), displaying a `wget` command you can copy to your local machine's clipboard.


Got wget? Get this URL.

We've got ampersands in the URL and they're all escaped and ready for wget. This URL won't work in your browser. Click here to select the entire command.

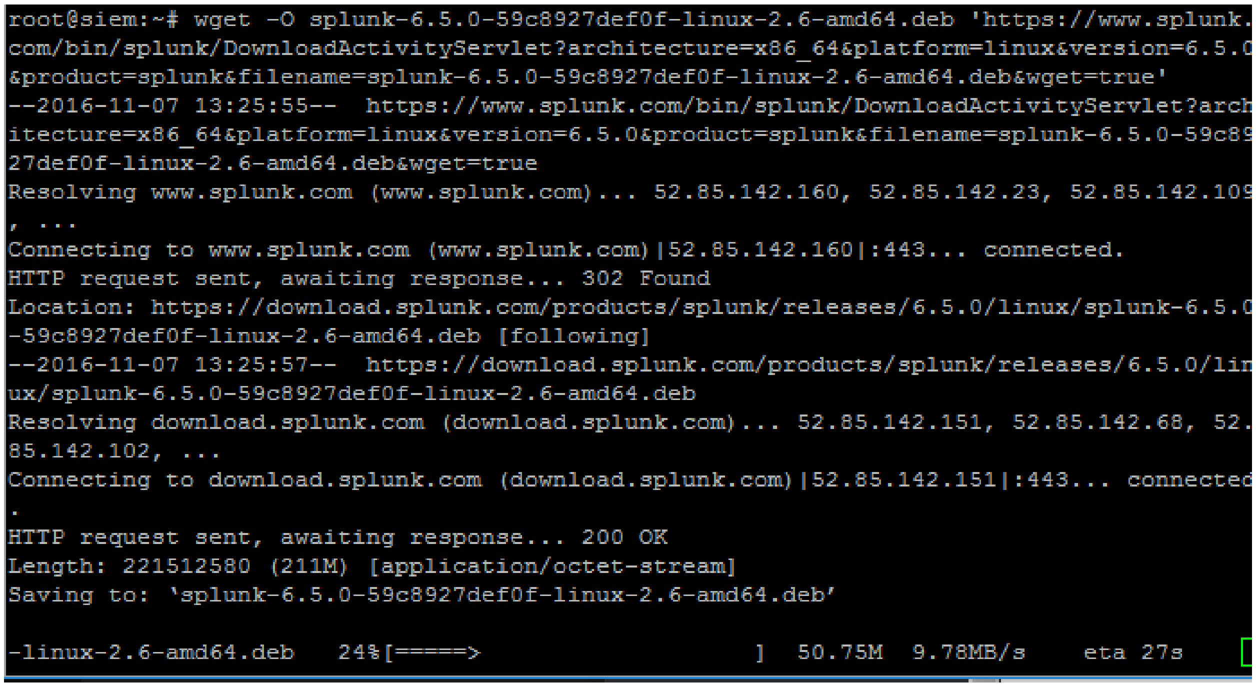
```
wget -O splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb 'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=6.5.0&product=splunk&filename=splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb&wget=true'
```

Next, log in to your SIEM VM with the respective account you created during the initial installation. To do so, use either the virtual machine's console via your hypervisor's management interface, or, if you did the Remote Lab Management configuration exercises, connect to the VM over SSH. Next, at the session's command prompt, run

```
sudo su -
```

and put in your user’s password, to get a `root` shell.

If you are using SSH, you can paste the `wget` command line you copied from splunk.com and run it If you are using your hypervisor's console to interact with the SIEM VM, you will have to manually copy the wget command. Be aware, that any typos in the command will result in failure to download the Splunk installation package. The output you receive from running the `wget` command should look similar to the one you can see in the picture below.



```
root@siem:~# wget -O splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb 'https://www.splunk.
com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=6.5.0
&product=splunk&filename=splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb&wget=true'
--2016-11-07 13:25:55--  https://www.splunk.com/bin/splunk/DownloadActivityServlet?arch
itecture=x86_64&platform=linux&version=6.5.0&product=splunk&filename=splunk-6.5.0-59c89
27def0f-linux-2.6-amd64.deb&wget=true
Resolving www.splunk.com (www.splunk.com)... 52.85.142.160, 52.85.142.23, 52.85.142.109
, ...
Connecting to www.splunk.com (www.splunk.com)|52.85.142.160|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://download.splunk.com/products/splunk/releases/6.5.0/linux/splunk-6.5.0
-59c8927def0f-linux-2.6-amd64.deb [following]
--2016-11-07 13:25:57--  https://download.splunk.com/products/splunk/releases/6.5.0/lin
ux/splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb
Resolving download.splunk.com (download.splunk.com)... 52.85.142.151, 52.85.142.68, 52.
85.142.102, ...
Connecting to download.splunk.com (download.splunk.com)|52.85.142.151|:443... connected
.
HTTP request sent, awaiting response... 200 OK
Length: 221512580 (211M) [application/octet-stream]
Saving to: 'splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb'

-linux-2.6-amd64.deb   24%[=====>                  ]  50.75M  9.78MB/s    eta 27s
```

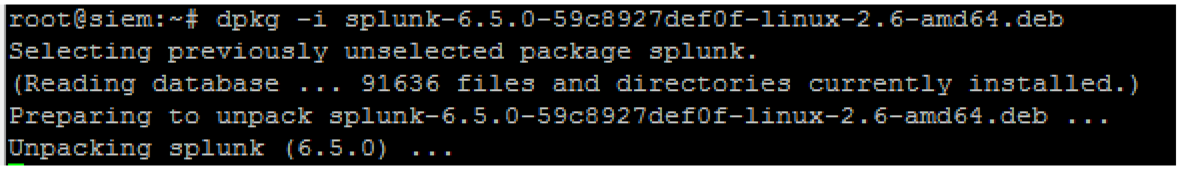
When the download is completed, there should be a file in root's home directory that starts with "splunk" and ends in ".deb". In my case, the file was named `splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb`.

To install Splunk from the downloaded software package, we have to use `dpkg`, the Debian (and Ubuntu) package manager. The command

```
dpkg -i [filename]
```

tells `dpkg` to try and install the file specified. In my case, the command line was

```
dpkg -i splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb
```

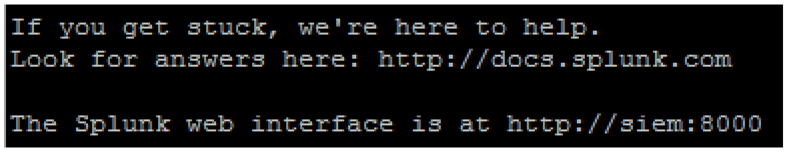


```
root@siem:~# dpkg -i splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb
Selecting previously unselected package splunk.
(Reading database ... 91636 files and directories currently installed.)
Preparing to unpack splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb ...
Unpacking splunk (6.5.0) ...
```

At this point, Splunk should be installed, but not started yet. To change that, try running:

```
/opt/splunk/bin/splunk start --accept-license
```

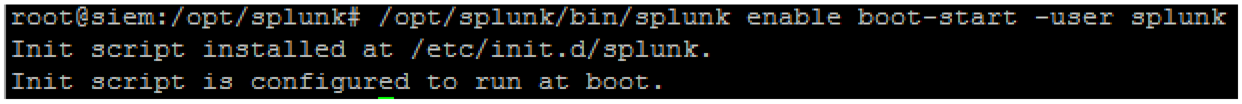
This command performs all of the initial setup tasks to get Splunk up and running, with the respective option set to accept the usage license. On completion, you should receive the notification depicted below, given everything proceeded without any errors.



```
If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com

The Splunk web interface is at http://siem:8000
```

The screen capture above offers really good advice. This guide is meant to get you through the initial setup, and confirm that data is sent to/received by Splunk normally. Beyond that, however, you will have to learn how to actually use the tool. That is where docs.splunk.com comes into place. Underneath that reference to the online documentation, we are informed that Splunk's web interface is up and running on port TCP 8000. If you are running a hosted hypervisor, you should be able to directly connect to http://172.16.1.3:8000 from that hosting machine. If you are using a bare-metal hypervisor, you need to create firewall rules on the pfSense system to allow your management workstation (or jump box) to connect to the Splunk web interface on the SIEM VM over port 8000/tcp. Before addressing that however, there is one more thing to be done on the console/SSH session before moving on. Run the following command:

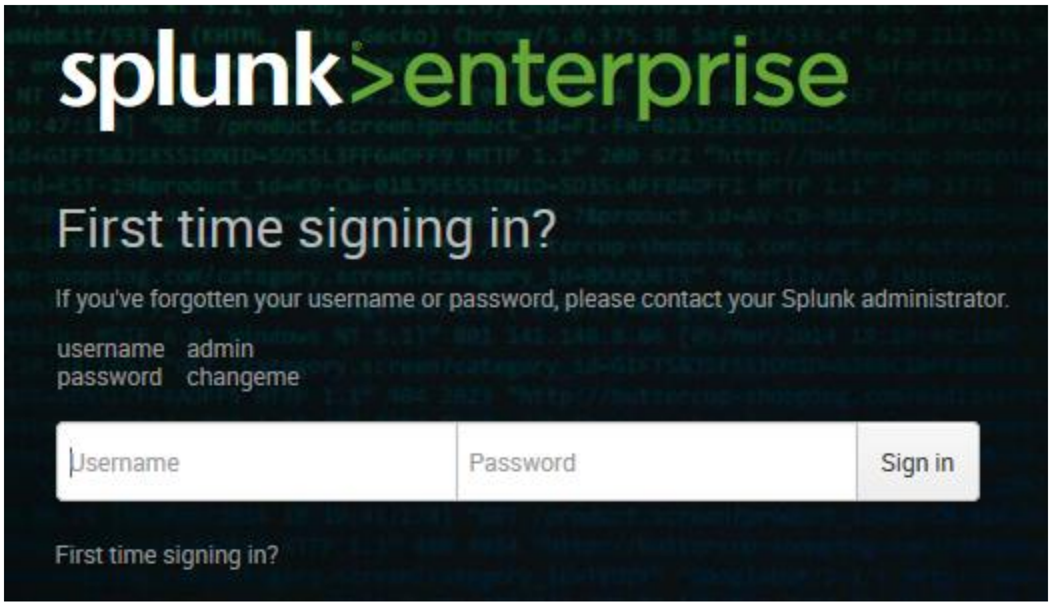
```
/opt/splunk/bin/splunk enable boot-start
```



```
root@siem:/opt/splunk# /opt/splunk/bin/splunk enable boot-start -user splunk
Init script installed at /etc/init.d/splunk.
Init script is configured to run at boot.
```

This command installs an init script to ensure that the `splunk` service starts up on system boot. In this way, if you have to reboot the SIEM VM after installing updates, due to maintenance etc., the `splunk` service should be started on boot automatically.

Next, using your favorite web browser, go to http://172.16.1.3:8000. You should be greeted with a login prompt like the one shown in the image below.



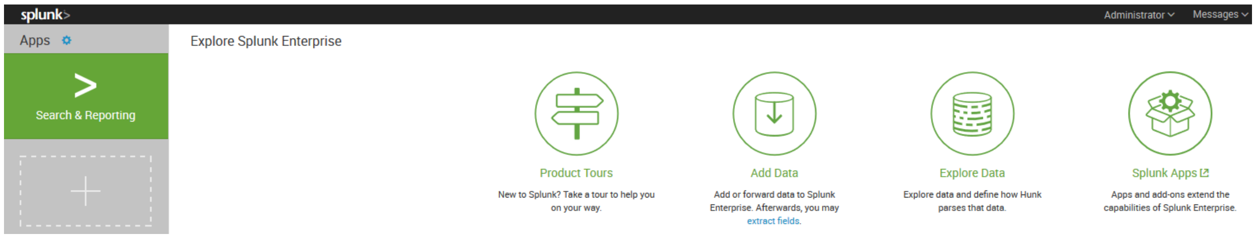
splunk>enterprise

First time signing in?

If you've forgotten your username or password, please contact your Splunk administrator.

username admin
password changeme

As the text implies, use the username `admin` and the password `changeme` to log in to the web UI for the first time. Splunk makes you select a new password for the admin user. After changing the password, you will be logged in. A small pop-up window with two checkboxes appears. You are asked whether you wish to submit data to make Splunk better, and whether you agree to sending license usage data to Splunk. Check both of these boxes, then click *OK*. Finally, you're greeted by the default splash page of your Splunk system.



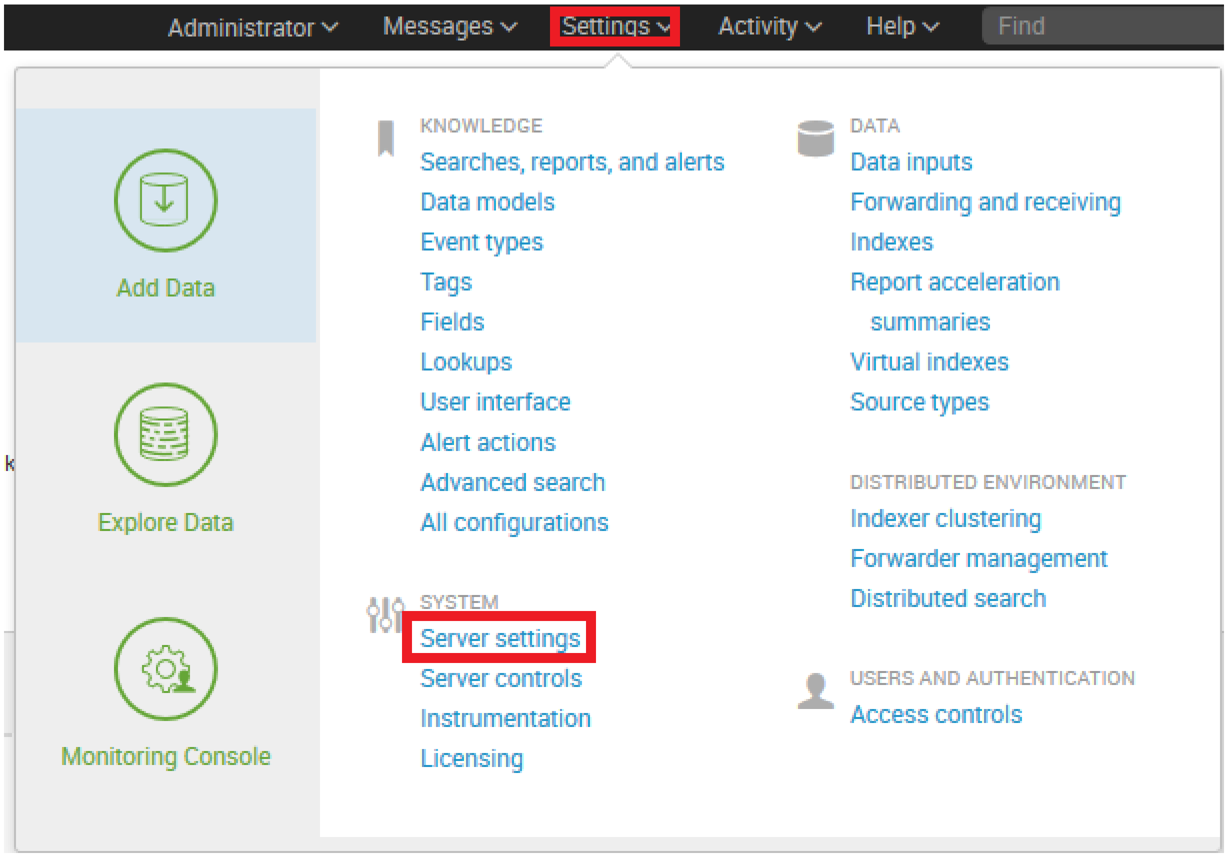
New to Splunk? Take a tour to help you on your way.

Add or forward data to Splunk Enterprise. Afterwards, you may extract fields.

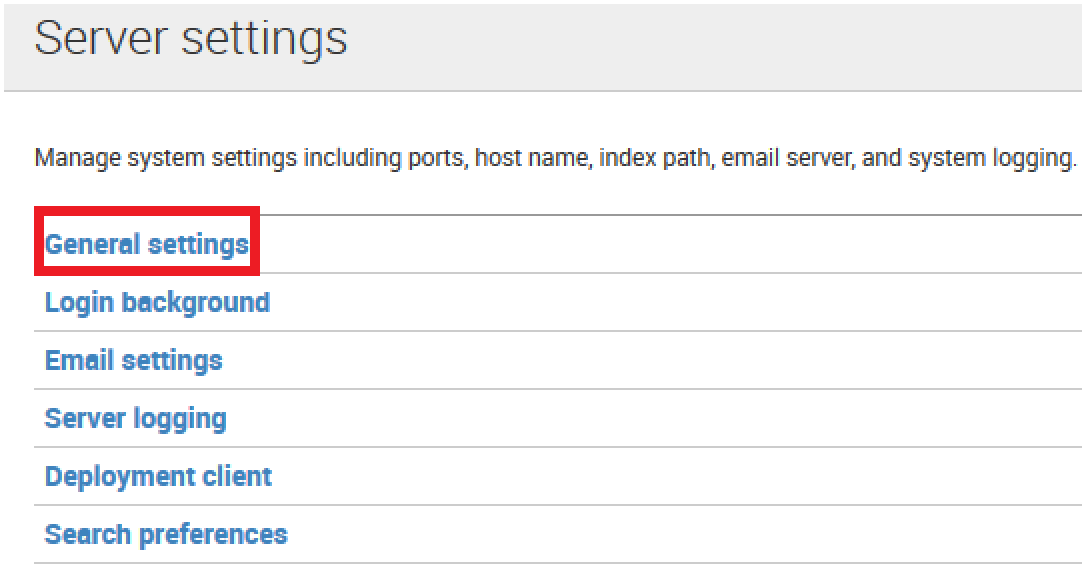
Explore data and define how Hunk parses that data.

Apps and add-ons extend the capabilities of Splunk Enterprise.

Congratulations, Splunk is up and running. But before we're done here, there is one more change we need to make. In the upper right corner of the screen, click on *Settings*. On the sub-menu that appears, click on *Server settings*.



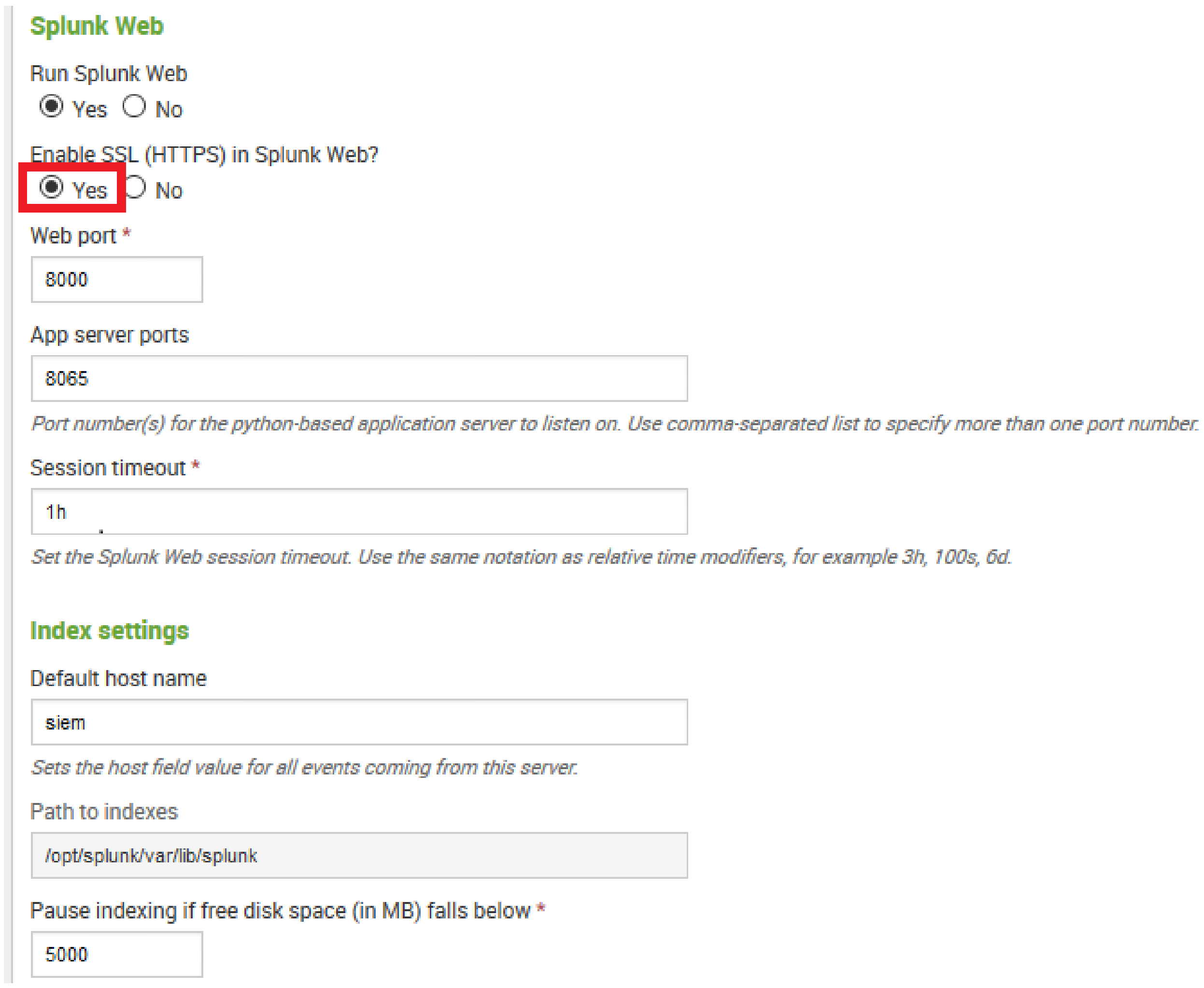
On the new page that pops up, click on *General settings*.



Server settings

Manage system settings including ports, host name, index path, email server, and system logging.

On this screen, there are two settings you need to be aware of: the *Splunk Web* and the *Index Settings*. Splunk Web settings control how the web interface operates. Change the setting under *Enable SSL (HTTPS) in Splunk Web?* to *Yes*. Further down, be aware that the Index Settings are set to stop Splunk if the free storage space on this system falls below 5GB. You can adjust that setting here, if it becomes necessary in the future.


Splunk Web

Run Splunk Web

[✓ Yes] No

Enable SSL (HTTPS) in Splunk Web?

[✓ Yes] No

Web port *

8000

App server ports

8065

Port number(s) for the python-based application server to listen on. Use comma-separated list to specify more than one port number.

Session timeout *

1h

Set the Splunk Web session timeout. Use the same notation as relative time modifiers, for example 3h, 100s, 6d.

Index settings

Default host name

siem

Sets the host field value for all events coming from this server.

Path to indexes

/opt/splunk/var/lib/splunk

Pause indexing if free disk space (in MB) falls below *

5000

After enabling SSL access, click the green *Save* button on the bottom of the page. To test things out, exit the Splunk web interface and reboot the SIEM VM. After the reboot is complete, try logging back into the web UI at https://172.16.1.3:8000.

If you are able to access the Splunk web interface, and manage to log in, everything is working correctly. In that case, proceed to the next section. If the login attempt fails, log in to the SIEM VM console and run

```
service splunk status
```

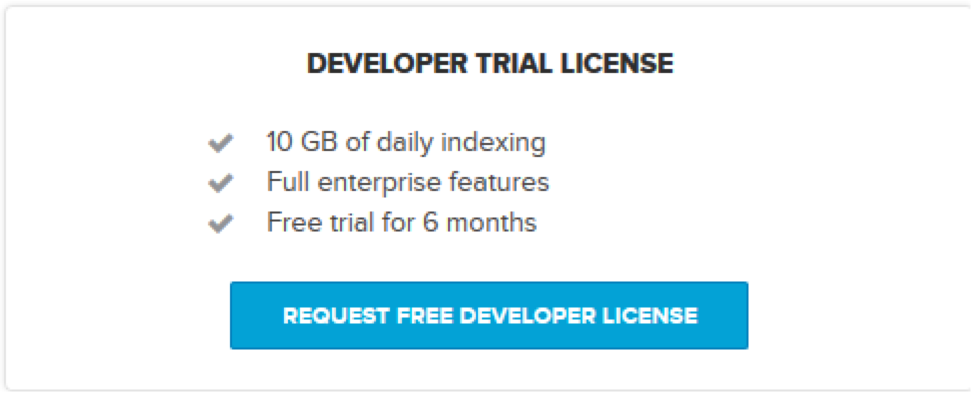
to get a status report. This usually includes an error message that may provide clues about the underlying issue.

(Optional) Requesting and Implementing a Splunk Dev License

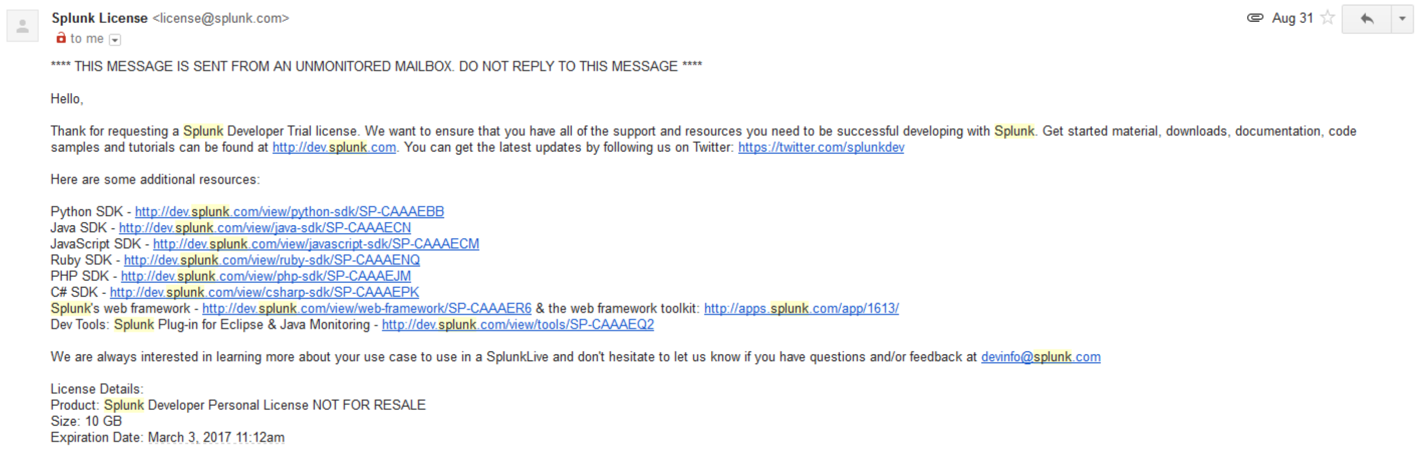
Note: Even though this section is marked as optional, I highly recommend following through with this. The increased data limit of 10GB per day will definitely come in handy, especially if you decide to expand your lab network and/or want to perform log and statistical analysis in the future.

As previously mentioned, by default, the free version of Splunk Enterprise edition allows you store up to 500MB of data in Splunk per day. However, you can request access to a developer license to increase that limit significantly. Be aware that developer licenses are meant for home or testing environments; they are NOT intended for commercial or production use. Frankly, I think it's awesome that Splunk allows this sort of flexibility, essentially handing these licenses out like candy for anyone who wants to install and run the software at home/in a lab environment, to learn how to configure and use it and/or develop cool tools for it. But as the saying goes, that it only takes one bad apple to spoil the bunch, all it takes is a number of license abuse cases for Splunk to shut down this particular program. So, what I'm getting at here is: **act ethically and responsibly.**

In your favorite web browser, navigate to https://splunkbase.splunk.com/develop/. Use the information of the splunk.com account you created earlier to log in, then find and click on the *REQUEST FREE DEVELOPER LICENSE* button.



On the next page, under *2. Get your developer license.*, click on the link labeled *request your developer license*. At this point, all you can do is wait to get your license sent to you. As soon as your request has been approved, an e-mail from license@splunk.com (similar to the one shown in the picture below) will arrive in your inbox.



Splunk License <license@splunk.com>
to me
Aug 31

**** THIS MESSAGE IS SENT FROM AN UNMONITORED MAILBOX. DO NOT REPLY TO THIS MESSAGE ****

Hello,

Thank for requesting a Splunk Developer Trial license. We want to ensure that you have all of the support and resources you need to be successful developing with Splunk. Get started material, downloads, documentation, code samples and tutorials can be found at http://dev.splunk.com. You can get the latest updates by following us on Twitter: https://twitter.com/splunkdev

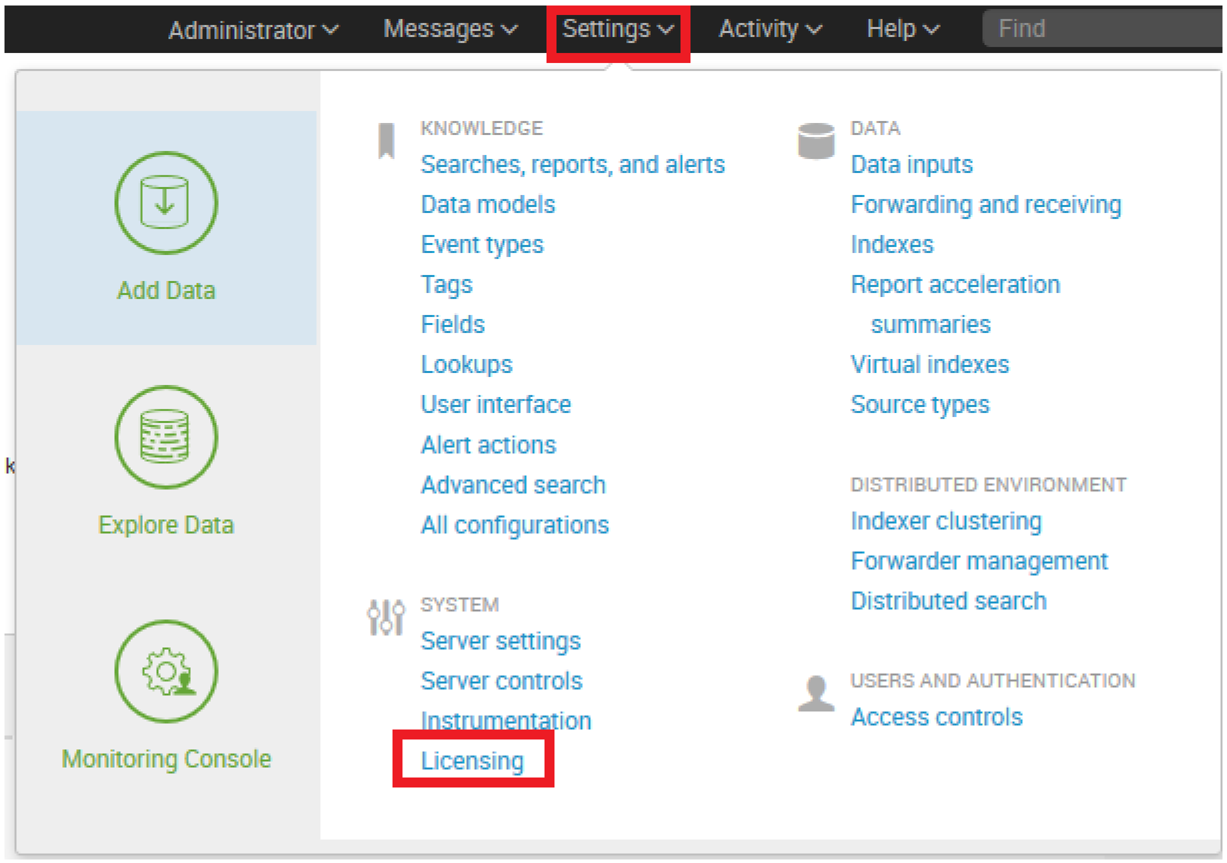
Here are some additional resources:

Python SDK - http://dev.splunk.com/view/python-sdk/SP-CAAAEBB
Java SDK - http://dev.splunk.com/view/java-sdk/SP-CAAAECN
JavaScript SDK - http://dev.splunk.com/view/javascript-sdk/SP-CAAAECM
Ruby SDK - http://dev.splunk.com/view/ruby-sdk/SP-CAAAENQ
PHP SDK - http://dev.splunk.com/view/php-sdk/SP-CAAAEJM
C# SDK - http://dev.splunk.com/view/csharp-sdk/SP-CAAAEPK
Splunk's web framework - http://dev.splunk.com/view/web-framework/SP-CAAAER6 & the web framework toolkit: http://apps.splunk.com/app/1613/
Dev Tools: Splunk Plug-in for Eclipse & Java Monitoring - http://dev.splunk.com/view/tools/SP-CAAAEQ2

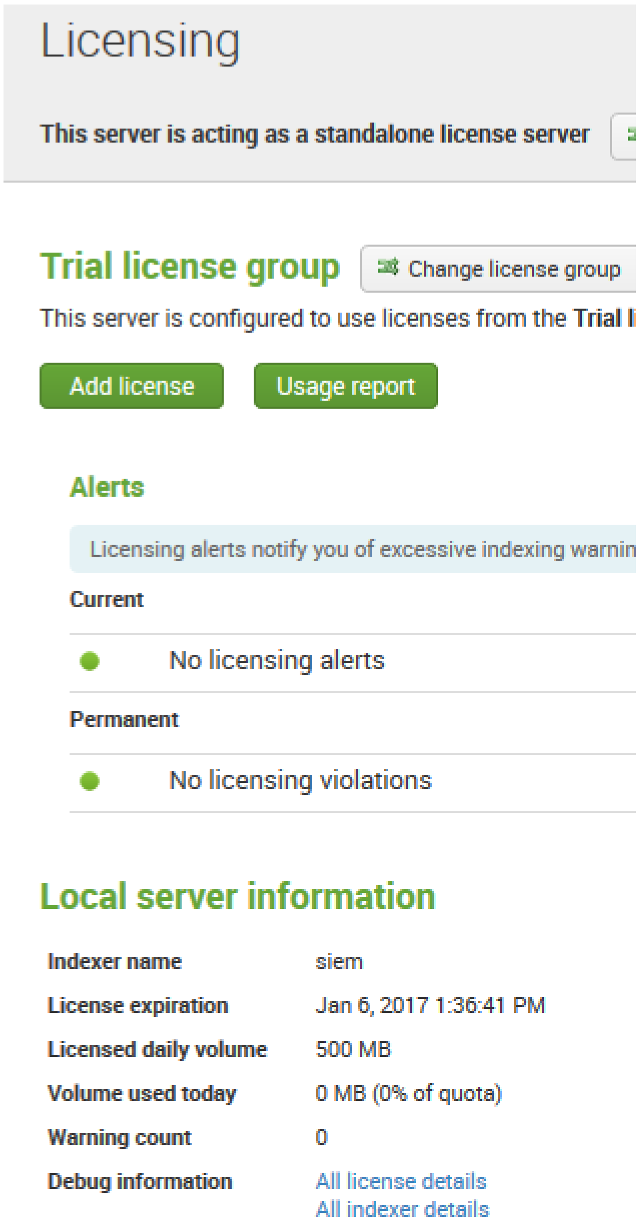
We are always interested in learning more about your use case to use in a SplunkLive and don't hesitate to let us know if you have questions and/or feedback at devinfo@splunk.com

License Details:
Product: Splunk Developer Personal License NOT FOR RESALE
Size: 10 GB
Expiration Date: March 3, 2017 11:12am

Attached to this message, you receive a Splunk license file, `splunk.license`. Download this file to your machine, then use your web browser to navigate to your Splunk search head at https://172.16.1.3:8000. Log in as `admin`, open the *Settings* menu, then click on *Licensing*.



On the page that appears, captioned *Licensing*, click the *Add license* button.



Licensing

This server is acting as a standalone license server

Trial license group

This server is configured to use licenses from the Trial l

Alerts

Licensing alerts notify you of excessive indexing warnin

Current

● No licensing alerts

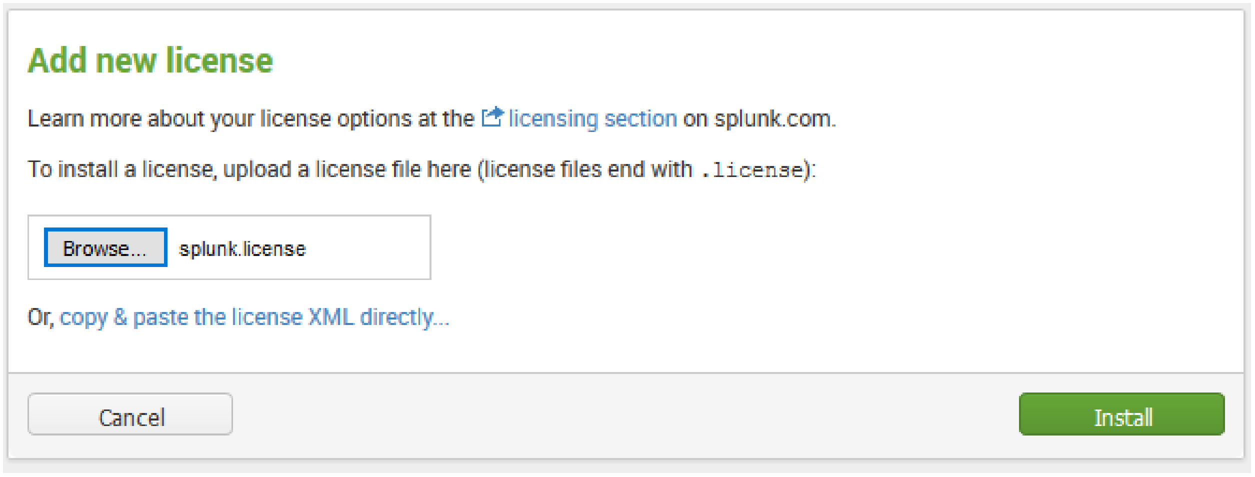
Permanent

● No licensing violations

Local server information

Indexer name	siem
License expiration	Jan 6, 2017 1:36:41 PM
Licensed daily volume	500 MB
Volume used today	0 MB (0% of quota)
Warning count	0
Debug information	All license details All indexer details

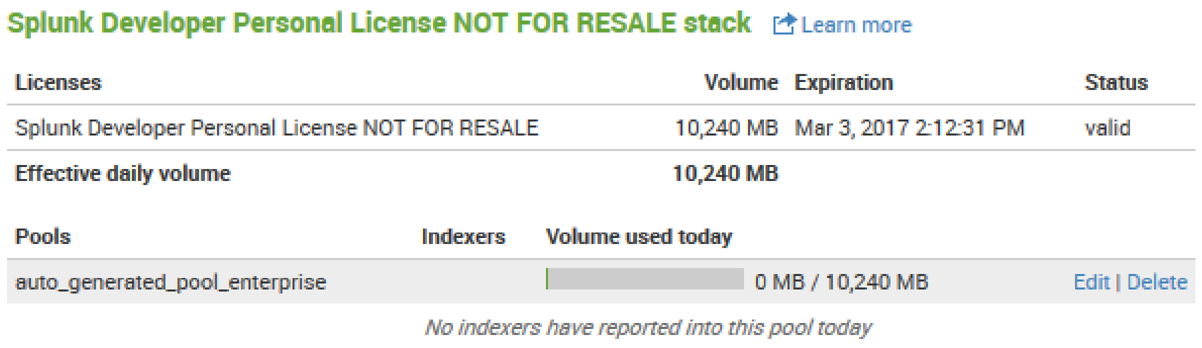
In the *Add new license* window that pops up, click on the *Browse…* button to open an Explorer window. Navigate to where you stored the `splunk.license` file and select it. Back in the *Add new license* window, click the green *Install* button in the lower right corner.


Add new license

Learn more about your license options at the licensing section on splunk.com.

To install a license, upload a license file here (license files end with `.license`):

The next page asks you to restart Splunk to apply the license. As you don't have any data being collected or indexed at the moment, there's no reason to not select *Restart now*. As soon as Splunk is up and available again, log back in as `admin`. The licensing page should have been updated, displaying the newly acquired and installed development license. Log off if you were able to verify this.


Splunk Developer Personal License NOT FOR RESALE stack

Licenses	Volume	Expiration	Status
Splunk Developer Personal License NOT FOR RESALE	10,240 MB	Mar 3, 2017 2:12:31 PM	valid
Effective daily volume	**10,240 MB**		

Pools	Indexers	Volume used today	
auto_generated_pool_enterprise		0 MB / 10,240 MB	Edit | Delete

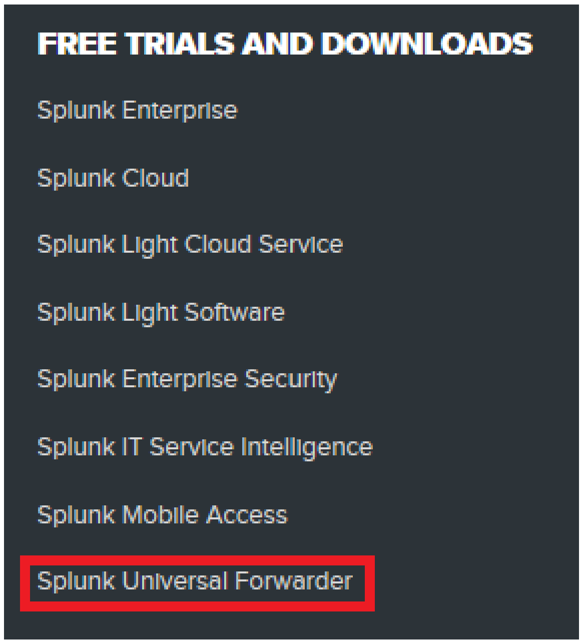
No indexers have reported into this pool today

At this point, the Splunk service on the SIEM VM is all but set up. **Take a snapshot of the SIEM VM.** As soon as you are ready, let's move on to setting up a Universal Forwarder on the IPS VM.

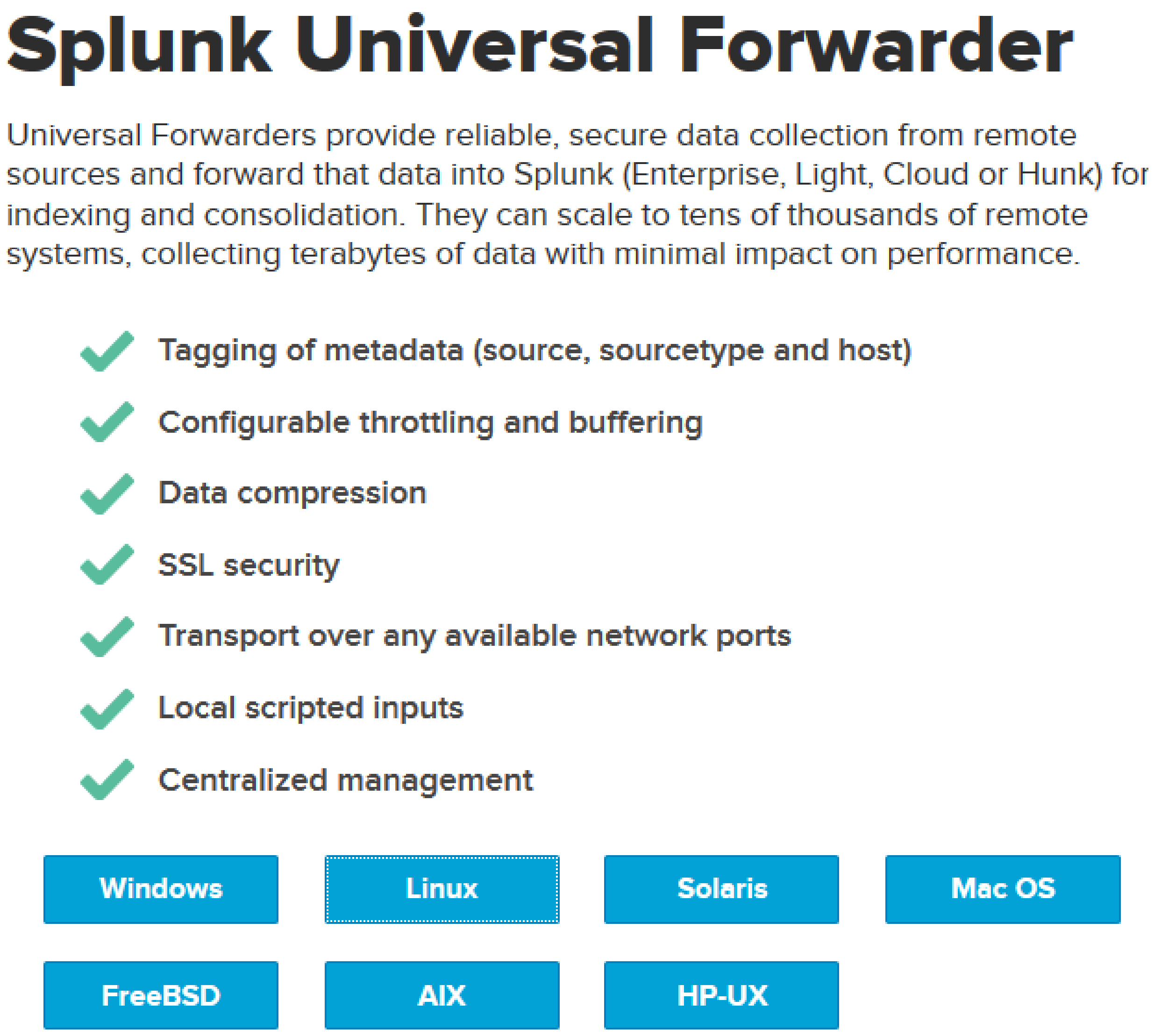
Universal Forwarder Setup

The Splunk universal forwarder is responsible for sending logs from a given system to a selected heavy forwarder or indexer for collection and organization. For our lab, we'll download the universal forwarder package for Linux, install it on our IPS VM, and configure it to send the IDS logs from Snort or Suricata (whichever you chose to install) to Splunk indexer on our SIEM VM for processing.

First, using your favorite web browser, navigate to splunk.com and log in with your account. On the bottom of the main page, under the list captioned *Free Trials and Downloads*, click on the link entitled *Splunk Universal Forwarder*.


FREE TRIALS AND DOWNLOADS
Splunk Enterprise
Splunk Cloud
Splunk Light Cloud Service
Splunk Light Software
Splunk Enterprise Security
Splunk IT Service Intelligence
Splunk Mobile Access
Splunk Universal Forwarder

Not unlike with the Splunk Enterprise installation files, the universal forwarder can be downloaded for and installed on a variety of operating systems.

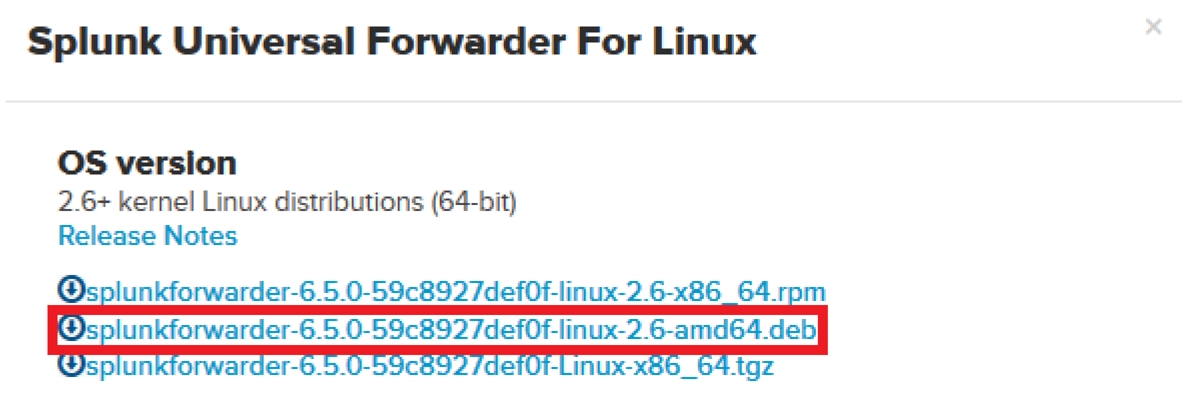


Splunk Universal Forwarder

Universal Forwarders provide reliable, secure data collection from remote sources and forward that data into Splunk (Enterprise, Light, Cloud or Hunk) for indexing and consolidation. They can scale to tens of thousands of remote systems, collecting terabytes of data with minimal impact on performance.

- ✓ Tagging of metadata (source, sourcetype and host)
- ✓ Configurable throttling and buffering
- ✓ Data compression
- ✓ SSL security
- ✓ Transport over any available network ports
- ✓ Local scripted inputs
- ✓ Centralized management

On the download page for the universal forwarder, click the *Linux* button. On the list of package options that appears, select the 64-bit package with the `.deb` file extension.


Splunk Universal Forwarder For Linux

OS version

2.6+ kernel Linux distributions (64-bit)

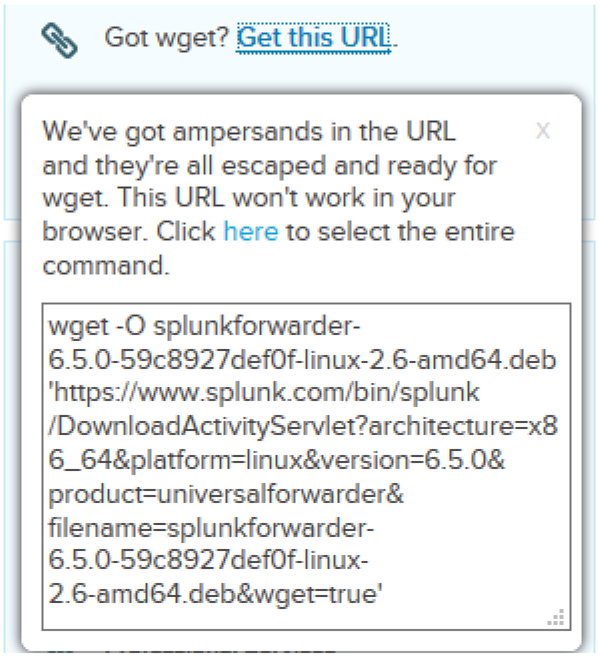
Release Notes

splunkforwarder-6.5.0-59c8927def0f-linux-2.6-x86_64.rpm

splunkforwarder-6.5.0-59c8927def0f-linux-2.6-amd64.deb

splunkforwarder-6.5.0-59c8927def0f-Linux-x86_64.tgz

After accepting the EULA that pops up, your browser will display a splash page, thanking you for downloading the universal forwarder, just like the one we were shown when downloading the Splunk Enterprise installer. And just like with the aforementioned package, there are two options to get this `.deb` file onto the IPS VM. You can download the installer via your web browser, save it locally, and use SCP to copy it to the IPS VM, or you can reveal the `wget` command to download the universal forwarder, copy it to your clipboard, then paste and run it on the IPS virtual machine itself (if you are using SSH to connect to the IPS VM. Otherwise, you will have to manually copy the `wget` command in order to download the package via `wget`). Choose the option you feel more comfortable with and go ahead.


Got wget? Get this URL.

We've got ampersands in the URL and they're all escaped and ready for wget. This URL won't work in your browser. Click here to select the entire command.

```
wget -O splunkforwarder-6.5.0-59c8927def0f-linux-2.6-amd64.deb 'https://www.splunk.com/bin/splunk/DownloadActivityServlet?architecture=x86_64&platform=linux&version=6.5.0&product=universalforwarder&filename=splunkforwarder-6.5.0-59c8927def0f-linux-2.6-amd64.deb&wget=true'
```

Once you have acquired the universal forwarder `.deb` package and placed it on the IPS VM, if you haven't already, log in to this virtual machine with the respective account you created during the initial installation (by means of SSH, or via the hypervisor's management interface). Next, at the session's command prompt, run

```
sudo su -
```

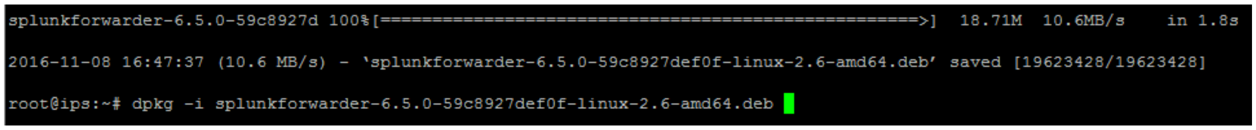
and put in your user's password, to get a `root` shell. After gaining `root` access, use the command

```
dpkg -i [filename]
```

to install the universal forwarder package. In my case, as indicated in the picture below, I used SSH key-based authentication to log in to the IPS virtual machine, executed the `wget` command line in a `root` shell to download the file, then ran

```
dpkg -i splunkforwarder-6.5.0-59c8927def0f-linux-2.6-amd64.deb
```

to install the universal forwarder.



```
splunkforwarder-6.5.0-59c8927d 100%[===================================================>]  18.71M  10.6MB/s    in 1.8s

2016-11-08 16:47:37 (10.6 MB/s) - 'splunkforwarder-6.5.0-59c8927def0f-linux-2.6-amd64.deb' saved [19623428/19623428]

root@ips:~# dpkg -i splunkforwarder-6.5.0-59c8927def0f-linux-2.6-amd64.deb
```

By default, the universal forwarder gets installed to `/opt/splunkforwarder`. For the time being, leave this console/SSH session open/connected. We will need our current session to finish the installation.

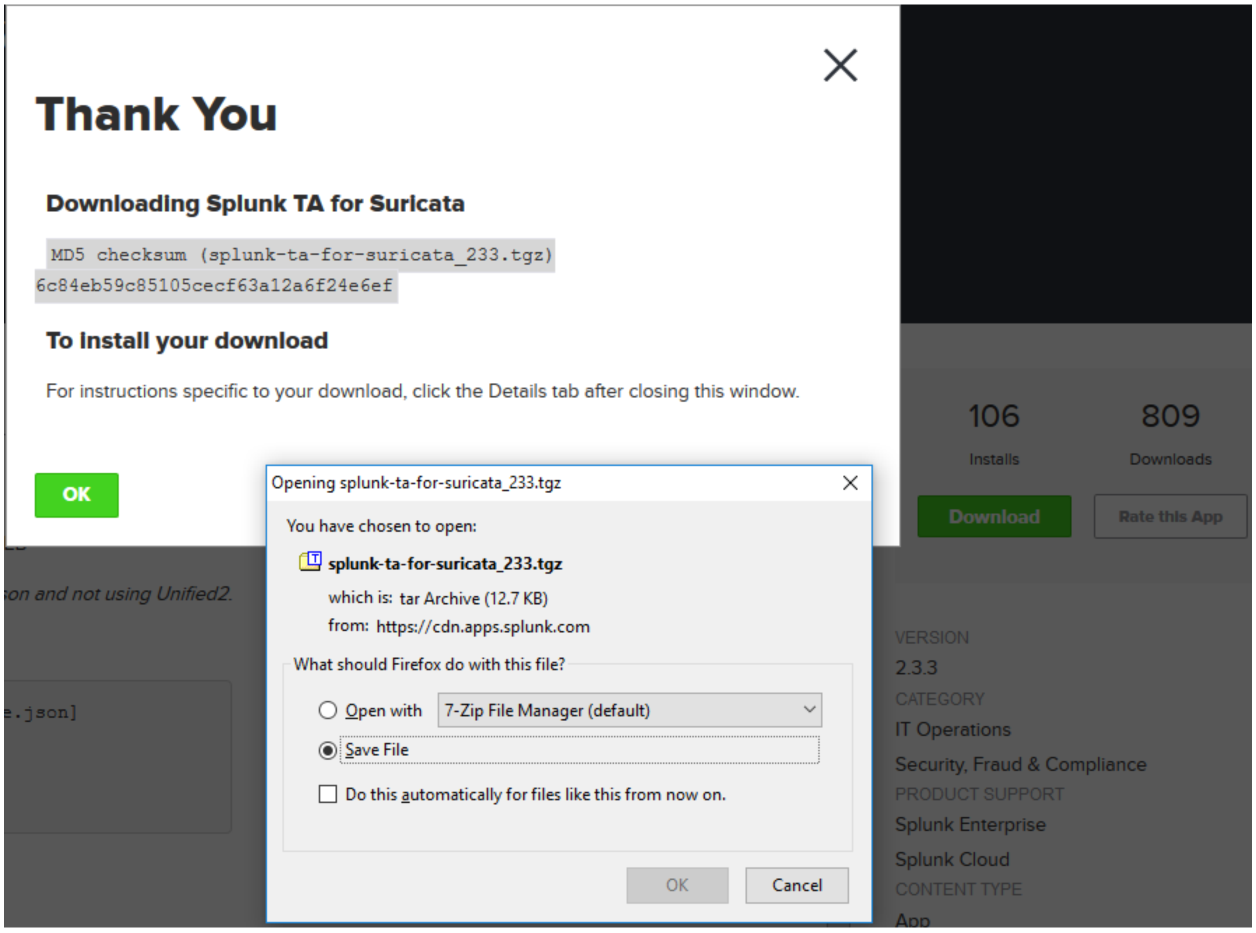
Before starting the universal forwarder, depending on which IPS software you chose to install and run, you need to acquire and install either the Splunk TA for Suricata, or the Hurricane Labs Add-On for Unified2 (if you're running Snort).

Splunk TA for Suricata

Splunk features what are known as technical applications or add-ons (also known as "TAs") for universal forwarders. These TAs enable the universal forwarder to parse a variety of different log types easily.

Splunk's various TAs and add-ons are available at splunkbase.splunk.com. Use your favorite browser to go there, and log in with your splunk.com credentials. The application we are looking for, "Splunk TA for Suricata", is currently located at splunkbase.splunk.com/app/2760/, with the

current version as of this writing being 2.3.3.
Click the Download button, accept the license terms, and save the file (which in my case was `splunk-ta-for-suricata_233.tgz`) to your workstation. Click OK to close the Thank You message box in your web browser, then log out of SplunkBase again.


Thank You

Downloading Splunk TA for Suricata

```
MD5 checksum (splunk-ta-for-suricata_233.tgz)
6c84eb59c85105cecf63a12a6f24e6ef
```

To install your download

For instructions specific to your download, click the Details tab after closing this window.

Opening splunk-ta-for-suricata_233.tgz

You have chosen to open:

splunk-ta-for-suricata_233.tgz

which is: tar Archive (12.7 KB)

from: https://cdn.apps.splunk.com

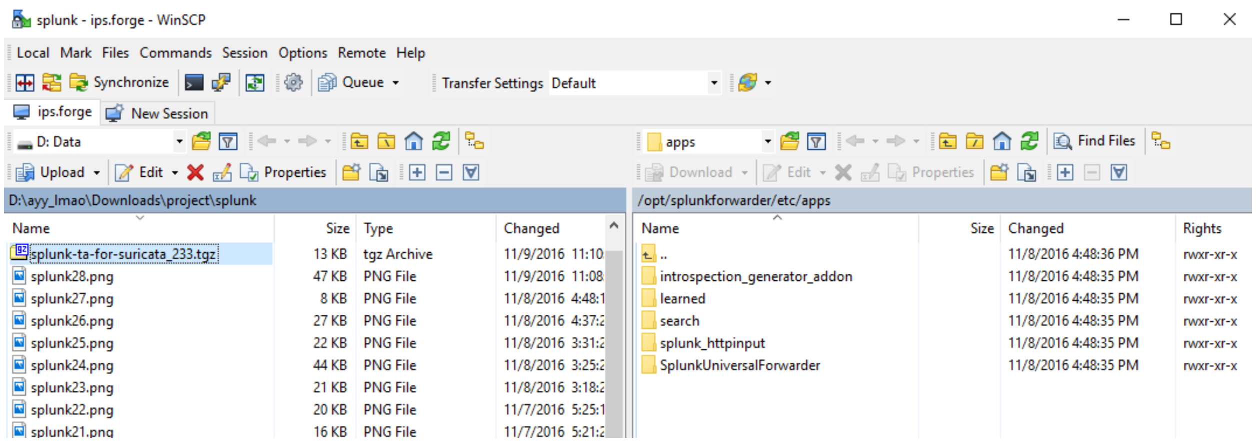
What should Firefox do with this file?

Open with 7-Zip File Manager (default) [✓ Save File]

Do this automatically for files like this from now on.

To continue, you will be required to use SCP to copy the file to the IPS VM. If you haven't already, I recommend setting up key-based authentication for your virtual machines, and allowing key-based authentication as the root user (in addition to enabling key-based authentication for WinSCP if you are using a Windows workstation, or hypervisor host).

If you are using Windows, open WinSCP, connect to the IPS VM as the `root` user, using key-based authentication, and copy the locally stored file `splunk-ta-for-suricata_233.tgz` to the remote directory `/opt/splunkforwarder/etc/apps`.


splunk - ips.forge - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
ips.forge
New Session
D: Data
apps
Find Files
Upload
Edit
Properties
Download

D:\ayy_lmao\Downloads\project\splunk

Name	Size	Type	Changed
splunk-ta-for-suricata_233.tgz	13 KB	tgz Archive	11/9/2016 11:10
splunk28.png	47 KB	PNG File	11/9/2016 11:08
splunk27.png	8 KB	PNG File	11/8/2016 4:48:1
splunk26.png	27 KB	PNG File	11/8/2016 4:37:2
splunk25.png	22 KB	PNG File	11/8/2016 3:31:2
splunk24.png	44 KB	PNG File	11/8/2016 3:25:2
splunk23.png	21 KB	PNG File	11/8/2016 3:18:2
splunk22.png	20 KB	PNG File	11/7/2016 5:25:1
splunk21.png	16 KB	PNG File	11/7/2016 5:21:2

/opt/splunkforwarder/etc/apps

Name	Size	Changed	Rights
..		11/8/2016 4:48:36 PM	rwxr-xr-x
introspection_generator_addon		11/8/2016 4:48:35 PM	rwxr-xr-x
learned		11/8/2016 4:48:35 PM	rwxr-xr-x
search		11/8/2016 4:48:35 PM	rwxr-xr-x
splunk_httpinput		11/8/2016 4:48:35 PM	rwxr-xr-x
SplunkUniversalForwarder		11/8/2016 4:48:35 PM	rwxr-xr-x

If you are on Linux/OS X, open your favorite terminal app, change directory (`cd`) into the folder you downloaded the `.tgz` package to (which normally is `~/Downloads`, but may differ for you), and run:

```
scp splunk-ta-for-suricata_233.tgz root@172.16.1.4:/opt/splunkforwarder/etc/apps
```

Assuming that key-based authentication has been configured for the root user on the IPS VM, the `scp` command should be able to utilize your SSH key to copy the file.

Note: the instructions above assume that you have enabled SSH access to the IPS VM, and that you have enabled SSH access to the IPS VM as the root user with key-based authentication. If you are not comfortable doing this, copy the `splunk-ta-for-suricata_233.tgz` file to the home directory of the user you created when you first installed Ubuntu on the IPS VM. For example, in my case, my default user's home directory would be `/home/ayy`. If you are using winSCP, simply log open an SCP session as that user. If you are using Linux, OS X or BSD, your scp command will differ slightly from the directory that contains the .tgz file we want to transfer (usually `~/Downloads`) run:

```
scp splunk-ta-for-suricata_233.tgz ayy@172.16.1.4:~/
```

This will place the file in that user's home directory. After copying the file to the IPS VM, either via a console session to the IPS VM, or an SSH session, run the following commands:

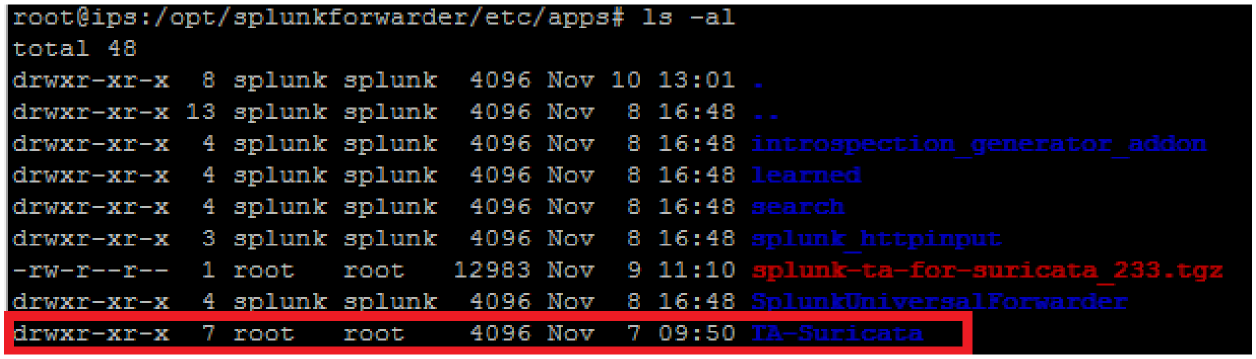
```
sudo su -
chown root:root /home/ayy/splunk-ta-for-suricata_233.tgz
mv /home/ayy/splunk-ta-for-suricata_233.tgz
```

`/opt/splunkforwarder/etc/apps/splunk-ta-for-suricata_233.tgz`
These commands have you become the root user (you will need root permissions for the remainder of this exercise), change the file permissions for `splunk-ta-for-suricata_233.tgz` so that root owns the file, then moves the file from the user's home directory, to `/opt/splunkfowarder/etc/apps` which is necessary for the next section. Please be aware that the `mv` command should be all on one line, but was broken into two lines here due to formatting.

After having copied the `.tgz` file to `/opt/splunkforwarder/etc/apps`, return to the terminal with the active console/SSH session connected to the IPS virtual machine. If you are not already the `root` user, run `sudo su -` to login as root, then run:

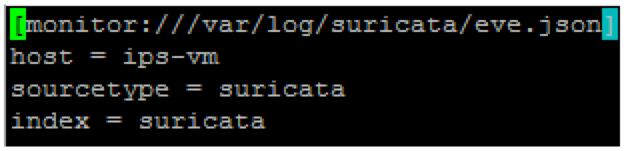
```
tar -xzvf splunk-ta-for-suricata_233.tgz
```

This will decompress the app and effectively "install" it to the `TA-Suricata` directory.



```
root@ips:/opt/splunkforwarder/etc/apps# ls -al
total 48
drwxr-xr-x  8 splunk splunk  4096 Nov 10 13:01 .
drwxr-xr-x 13 splunk splunk  4096 Nov  8 16:48 ..
drwxr-xr-x  4 splunk splunk  4096 Nov  8 16:48 introspection_generator_addon
drwxr-xr-x  4 splunk splunk  4096 Nov  8 16:48 learned
drwxr-xr-x  4 splunk splunk  4096 Nov  8 16:48 search
drwxr-xr-x  3 splunk splunk  4096 Nov  8 16:48 splunk_httpinput
-rw-r--r--  1 root   root   12983 Nov  9 11:10 splunk-ta-for-suricata_233.tgz
drwxr-xr-x  4 splunk splunk  4096 Nov  8 16:48 SplunkUniversalForwarder
drwxr-xr-x  7 root   root    4096 Nov  7 09:50 TA-Suricata
```

Change directory into `/opt/splunkforwarder/etc/apps/TA-Suricata/default`, and open the `inputs.conf` file for editing, using your favorite text editor. By running the Autosuricata script, `suricata.yaml` has been reconfigured to store the `eve.json` file in `/var/log/suricata`, which is the standard value set in the `inputs.conf` file. The default setting for `sourcetype` should be fine too, the only settings to modify are the values for `host` and `index`. I suggest to change the setting for `host` to `ips-vm` or something equally descriptive. As the `suricata` index does NOT exist on our system, change the value of `index` to `main`.



```
[monitor:///var/log/suricata/eve.json]
host = ips-vm
sourcetype = suricata
index = suricata
```

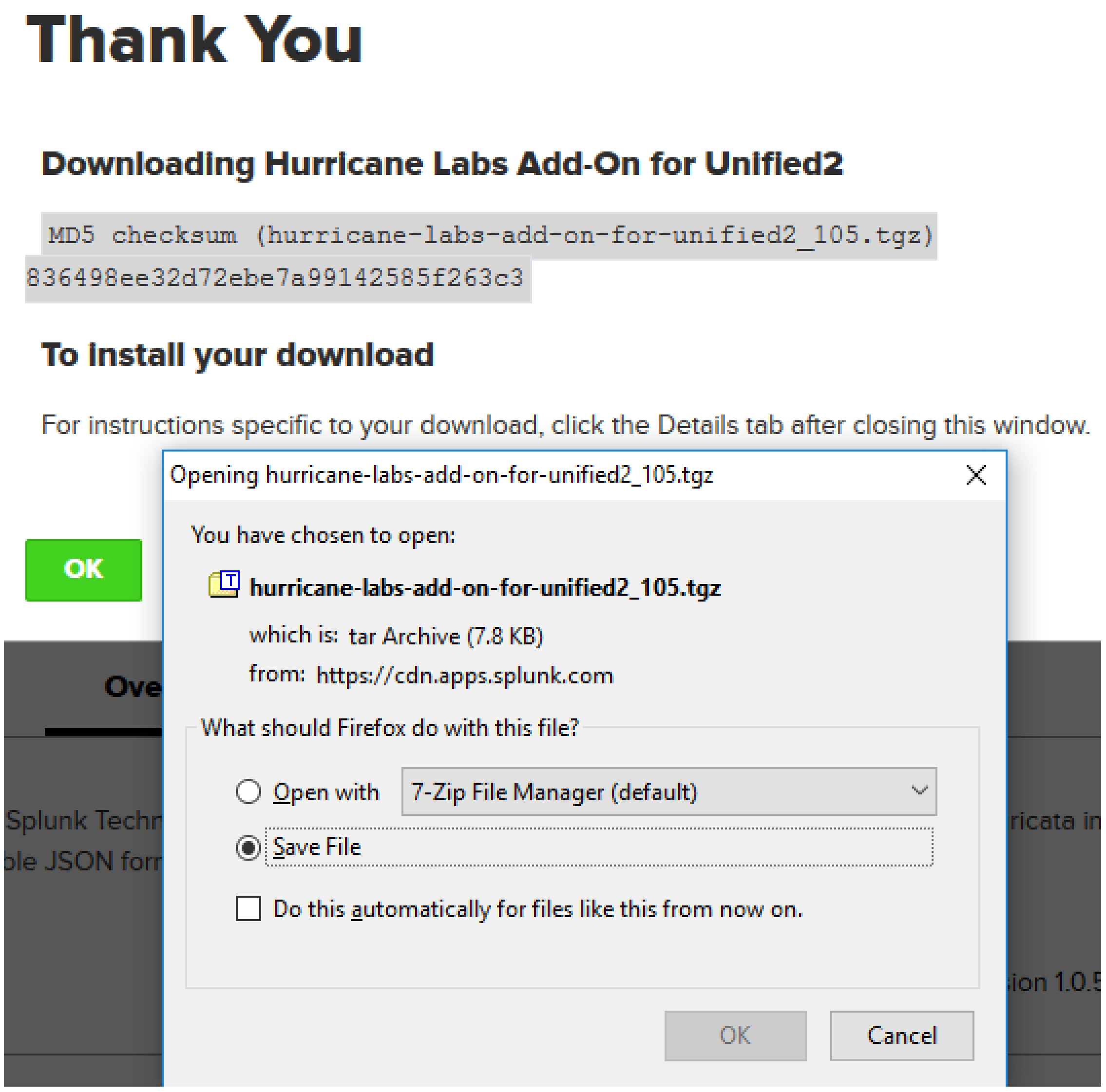
Note: A Splunk administrator asked me to point out that, while in our case using the default index of `main` is fine, since we don't have a ton of different types of data that we're throwing into our Splunk instance, in a non-lab environment (e.g. an enterprise network), however, where you will likely be dealing with IDS/IPS events from multiple sensors, setting up/running a separate index for your Suricata sensors makes much more sense.

When you are done, save the modified `inputs.conf` file, quit the text editor, and jump to the Starting The Forwarder + Persistence section for your next steps.

Hurricane Labs Add-On for Unified2

Splunk features what are known as technical applications or add-ons (also known as “TAs”) for Universal Forwarders. These TAs enable the universal forwarder to parse a variety of different log types easily. The Hurricane Labs Add-On for Unified2 is such a TA, not unlike the Splunk TA for Suricata.

Splunk's various TAs and add-ons are available at splunkbase.splunk.com. Use your favorite browser to go there, log in using your splunk.com website username and password, and navigate to splunkbase.splunk.com/app/1858/. As of writing, the current version of the Hurricane Labs Add-On for Unified2 app is 1.0.5. Click the *Download* button, accept the license terms, and save the file (which in my case was `hurricane-labs-add-on-for-unified2_105.tgz`) to your workstation. Click OK to close the Thank You message box in your web browser, then log out of SplunkBase again.



Thank You

Downloading Hurricane Labs Add-On for Unified2

```
MD5 checksum (hurricane-labs-add-on-for-unified2_105.tgz)
836498ee32d72ebe7a99142585f263c3
```

To install your download

For instructions specific to your download, click the Details tab after closing this window.

Opening hurricane-labs-add-on-for-unified2_105.tgz

You have chosen to open:

hurricane-labs-add-on-for-unified2_105.tgz

which is: tar Archive (7.8 KB)

from: https://cdn.apps.splunk.com

What should Firefox do with this file?

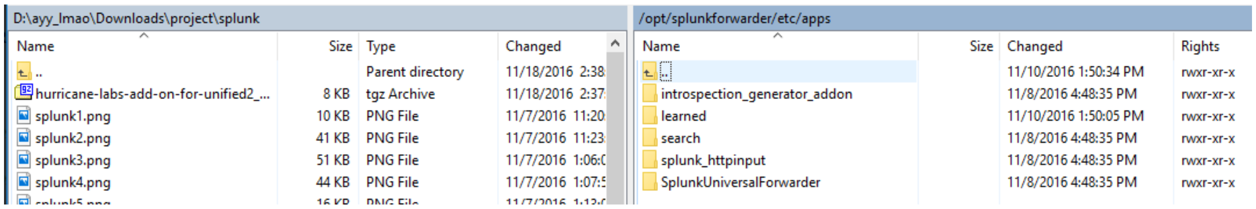
Open with 7-Zip File Manager (default)

[✓ Save File]

Do this automatically for files like this from now on.

To continue, you will be required to use SCP to copy the file to the IPS VM. If you haven't already, I recommend setting up key-based authentication for your virtual machines, and allowing key-based authentication as the root user (in addition to enabling key-based authentication for WinSCP if you are using a Windows workstation, or hypervisor host).

If you are using Windows, open WinSCP, connect to the IPS VM as the `root` user, using key-based authentication, and copy the locally stored file `hurricane-labs-add-on-for-unified2_105.tgz` to the remote directory `/opt/splunkforwarder/etc/apps`.



D:\ayy_lmao\Downloads\project\splunk

Name	Size	Type	Changed
..		Parent directory	11/18/2016 2:38
hurricane-labs-add-on-for-unified2_...	8 KB	tgz Archive	11/18/2016 2:37
splunk1.png	10 KB	PNG File	11/7/2016 11:20
splunk2.png	41 KB	PNG File	11/7/2016 11:23
splunk3.png	51 KB	PNG File	11/7/2016 1:06:
splunk4.png	44 KB	PNG File	11/7/2016 1:07:

/opt/splunkforwarder/etc/apps

Name	Size	Changed	Rights
..		11/10/2016 1:50:34 PM	rwxr-xr-x
introspection_generator_addon		11/8/2016 4:48:35 PM	rwxr-xr-x
learned		11/10/2016 1:50:05 PM	rwxr-xr-x
search		11/8/2016 4:48:35 PM	rwxr-xr-x
splunk_httpinput		11/8/2016 4:48:35 PM	rwxr-xr-x
SplunkUniversalForwarder		11/8/2016 4:48:35 PM	rwxr-xr-x

If you are on Linux/OS X, open your favorite terminal app, change directory (`cd`) into the folder you downloaded the `.tgz` package to (which normally is `~/Downloads`, but may differ for you), and run the command below (please note that this command should be entered on a single line, that it displays on two lines here due to formatting):

```
scp hurricane-labs-add-on-for-unified2_105.tgz
root@172.16.1.4:/opt/splunkforwarder/etc/apps
```

Assuming that key-based authentication has been configured for the root user on the IPS VM, the `scp` command should be able to utilize your SSH key to copy the file.

Note: the instructions above assume that you have enabled SSH access to the IPS VM, and that you have enabled SSH access to the IPS VM as the root user with key-based authentication. If you are not comfortable doing this, copy the `splunk-ta-for-suricata_233.tgz` file to the home directory of the user you created when you first installed Ubuntu on the IPS VM. For example, in my case, my default user's home directory would be `/home/ayy`. If you are using winSCP, simply log open an SCP session as that user. If you are using Linux, OS X or BSD, your scp command will differ slightly from the directory that contains the .tgz file we want to transfer (usually `~/Downloads`) run:

```
scp splunk-ta-for-suricata_233.tgz ayy@172.16.1.4:~/
```

This will place the file in that user's home directory. After copying the file to the IPS VM, either via a console session to the IPS VM, or an SSH session, run the following commands:

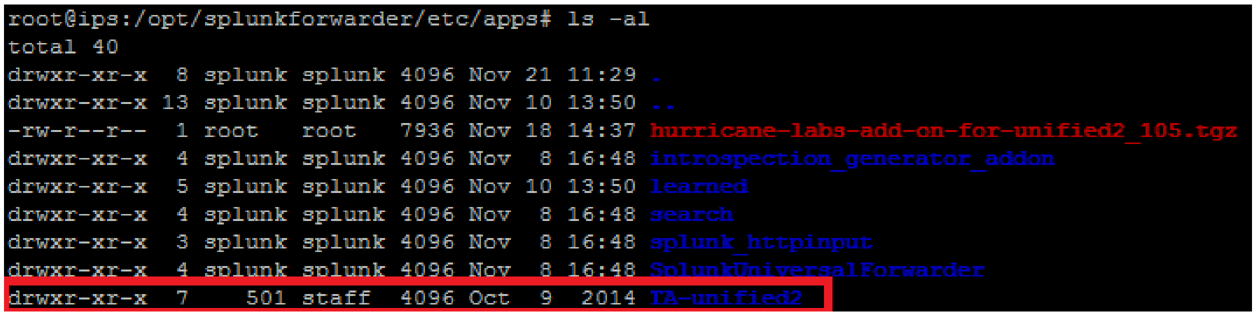
```
sudo su -
chown root:root /home/ayy/splunk-ta-for-suricata_233.tgz
mv /home/ayy/splunk-ta-for-suricata_233.tgz
/opt/splunkforwarder/etc/apps/splunk-ta-for-suricata_233.tgz
```

These commands have you become the root user (you will need root permissions for the remainder of this exercise), change the file permissions for `splunk-ta-for-suricata_233.tgz` so that root owns the file, then moves the file from the user's home directory, to `/opt/splunkfowarder/etc/apps` which is necessary for the next section. Please be aware that the `mv` command should be all on one line, but was broken into two lines here due to formatting.

After having copied the `.tgz` file to `/opt/splunkforwarder/etc/apps`, return to the terminal with the active console/SSH session connected to the IPS virtual machine. If you are not already the `root` user, run `sudo su -` to login as root, then run:

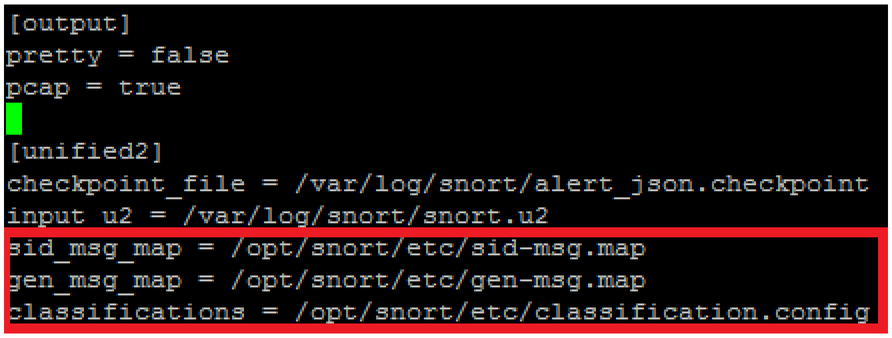
```
tar -xzvf hurricane-labs-add-on-for-unified2_105.tgz
```

This will decompress the app and effectively "install" it to the `TA-unified2` directory.



```
root@ips:/opt/splunkforwarder/etc/apps# ls -al
total 40
drwxr-xr-x  8 splunk splunk 4096 Nov 21 11:29 .
drwxr-xr-x 13 splunk splunk 4096 Nov 10 13:50 ..
-rw-r--r--  1 root   root   7936 Nov 18 14:37 hurricane-labs-add-on-for-unified2_105.tgz
drwxr-xr-x  4 splunk splunk 4096 Nov  8 16:48 introspection_generator_addon
drwxr-xr-x  5 splunk splunk 4096 Nov 10 13:50 learned
drwxr-xr-x  4 splunk splunk 4096 Nov  8 16:48 search
drwxr-xr-x  3 splunk splunk 4096 Nov  8 16:48 splunk_httpinput
drwxr-xr-x  4 splunk splunk 4096 Nov  8 16:48 SplunkUniversalForwarder
drwxr-xr-x  7    501 staff  4096 Oct  9  2014 TA-unified2
```

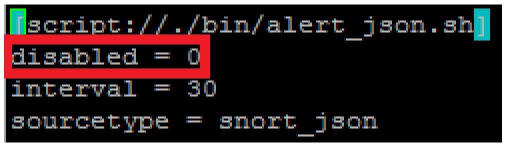
Change directory into `/opt/splunkforwarder/etc/apps/TA-unified2/default`, and open the `unified2.conf` file for editing, using your favorite text editor. In this file, there are 3 lines that need to be edited: `sid_msg_map`, `gen_msg_map`, and `classifications`. By default, these lines assume that they are placed in `/etc/snort`. As shown in the image below, change the respective values to `/opt/snort/etc` instead (e.g. `/opt/snort/etc/sid-msg.map`, `/opt/snort/etc/gen-msg.map`, and `/opt/snort/etc/classification.config`).



```
[output]
pretty = false
pcap = true

[unified2]
checkpoint_file = /var/log/snort/alert_json.checkpoint
input_u2 = /var/log/snort/snort.u2
sid_msg_map = /opt/snort/etc/sid-msg.map
gen_msg_map = /opt/snort/etc/gen-msg.map
classifications = /opt/snort/etc/classification.config
```

Note: You may notice the line "`pretty = false`" in the `unified2.conf` file. If you change the value of `pretty` to `true`, this enables more verbose output from our sensor to our Splunk instance.

After modifying these lines, save and close `unified2.conf`, then open the `input.conf` file for editing. Change the line `disabled = 1` to `disabled = 0`. When you are done, save the modified `inputs.conf` file, and quit the text editor.



```
[script://./bin/alert_json.sh]
disabled = 0
interval = 30
sourcetype = snort_json
```

Before moving on, we need to install python for the TA-unified2 app to work properly. Still in the `root` shell on the IPS VM, run the command

```
apt-get -y install python
```

Note: By default, this TA dumps Snort logs to the “snort_json” sourcetype, under the “main” index. We have to remember this important detail when querying the IDS logs for data to verify everything is working, which we will do in a few moments.

With the universal forwarder set up, proceed to the Starting The Forwarder + Persistence section below.

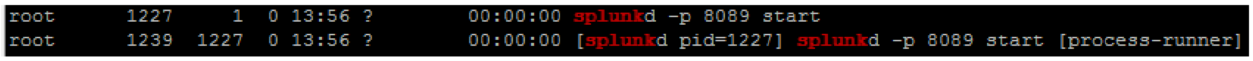
Starting The Forwarder + Persistence

After finishing its configuration,, we need to start the universal forwarder in order to send the log data to the Splunk system on the SIEM VM. To do this, enter and run the following lines, one at a time:

```
/opt/splunkforwarder/bin/splunk start --accept-license
/opt/splunkforwarder/bin/splunk stop
/opt/splunkforwarder/bin/splunk add forward-server 172.16.1.3:9997
/opt/splunkforwarder/bin/splunk enable boot-start
init 6
```

The first command performs all of the initial setup tasks to get the universal forwarder up and running, with the respective option set to accept the usage license..The second command stops the server again, while the third command configures the universal forwarder to send its data to 172.16.1.3 on TCP port 9997. When executed, the second to last line enables the universal forwarder on system boot, while the last command triggers an immediate reboot of the system. After the reboot is complete, log back in to the IPS VM and run

```
ps -ef | grep splunk
```



```
root      1227     1  0 13:56 ?        00:00:00 splunkd -p 8089 start
root      1239  1227  0 13:56 ?        00:00:00 [splunkd pid=1227] splunkd -p 8089 start [process-runner]
```

If your output looks similar to the one displayed in the illustration above, then the Splunk universal forwarder was able to start successfully. If your results look different , run

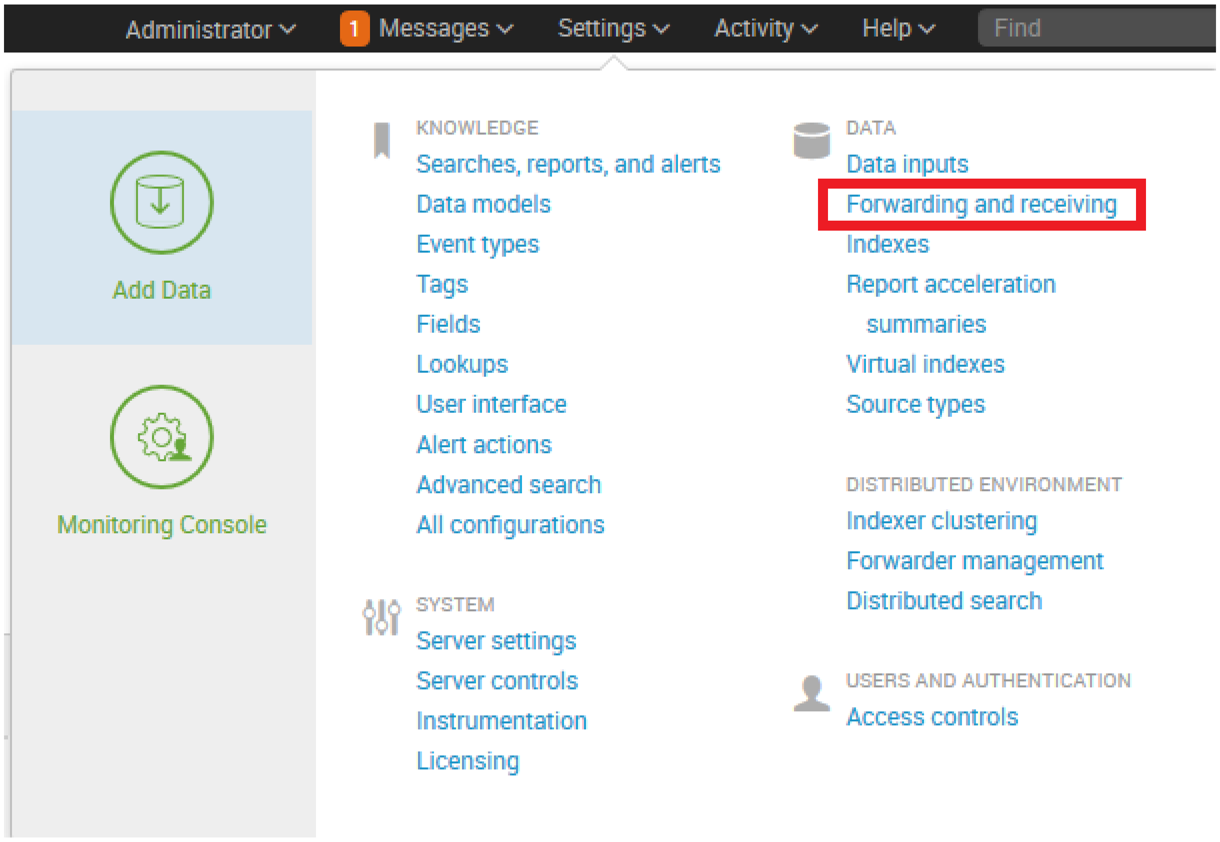
```
service splunk status
```

to try and determine the issue. Additionally, consider reviewing the universal forwarder log files (which can be found at `/opt/splunkforwarder/var/log/splunk/*.log`) for clues that may indicate why the Splunk universal forwarder might be misbehaving. Please note that if you were to check `splunkd.log` at this point, you might notice the following log entries:

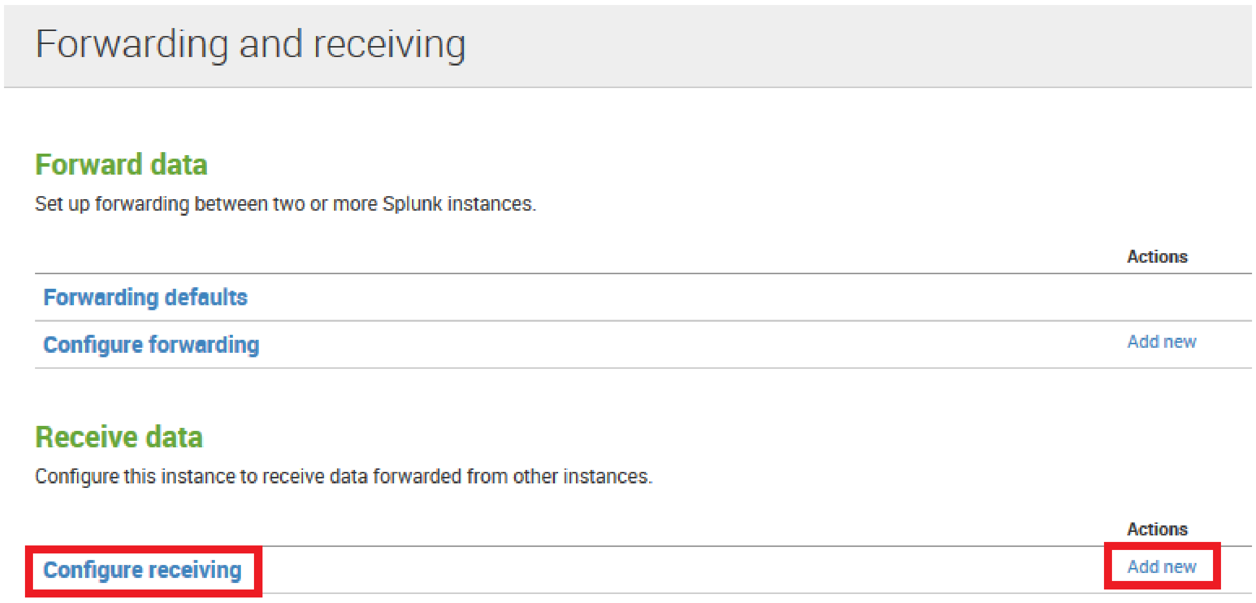
```
11-10-2016 14:12:19.080 -0500 WARN  TcpOutputFd - Connect to 172.16.1.3:9997 failed. Connection refused
11-10-2016 14:12:19.081 -0500 ERROR TcpOutputFd - Connection to host=172.16.1.3:9997 failed
```

As we haven’t finished our setup yet, those messages being logged are normal and expected. We’re going to fix this momentarily.

Using your favorite web browser, log in to your Splunk instance on the SIEM VM (via https://172.16.1.3:8000), and navigate to *Settings* > *Forwarding and receiving*.



On the page that appears, entitled *Forwarding and receiving*, under the section titled *Receiving data*, find the line that starts with the label *Configure receiving*, and click the *Add new* link on the right.



Forwarding and receiving

Forward data

Set up forwarding between two or more Splunk instances.

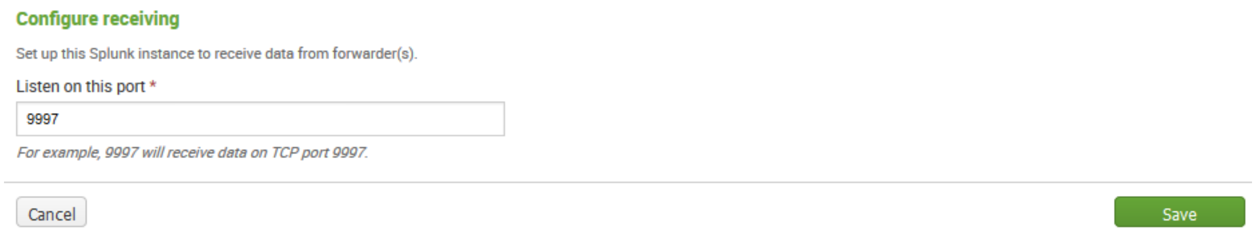
	Actions
Forwarding defaults	
Configure forwarding	Add new

Receive data

Configure this instance to receive data forwarded from other instances.

	Actions
Configure receiving	Add new

You are brought to the*Configure receiving* page. Enter “9997” in the input box labeled *Listen on this port*, then click the green *Save* button.



Configure receiving

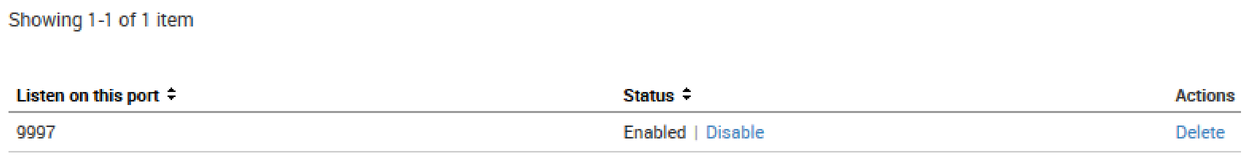
Set up this Splunk instance to receive data from forwarder(s).

Listen on this port *

9997

For example, 9997 will receive data on TCP port 9997.

The confirmation view appears, showing that a new TCP listener on port 9997 has been enabled and is waiting for data from our universal forwarder. Return to the Splunk home page.



Showing 1-1 of 1 item

Listen on this port	Status	Actions
9997	Enabled | Disable	Delete

Testing Splunk and the Universal Forwarder

The quickest way to test an IDS is to generate what I like to call “The Full Broadside Battery”. Recall that the virtual machine running Metasploitable 2 is incredibly vulnerable. You may also recall that I referred to our Kali VM as our obnoxiously loud attacker. You’re about to find out why in a minute.

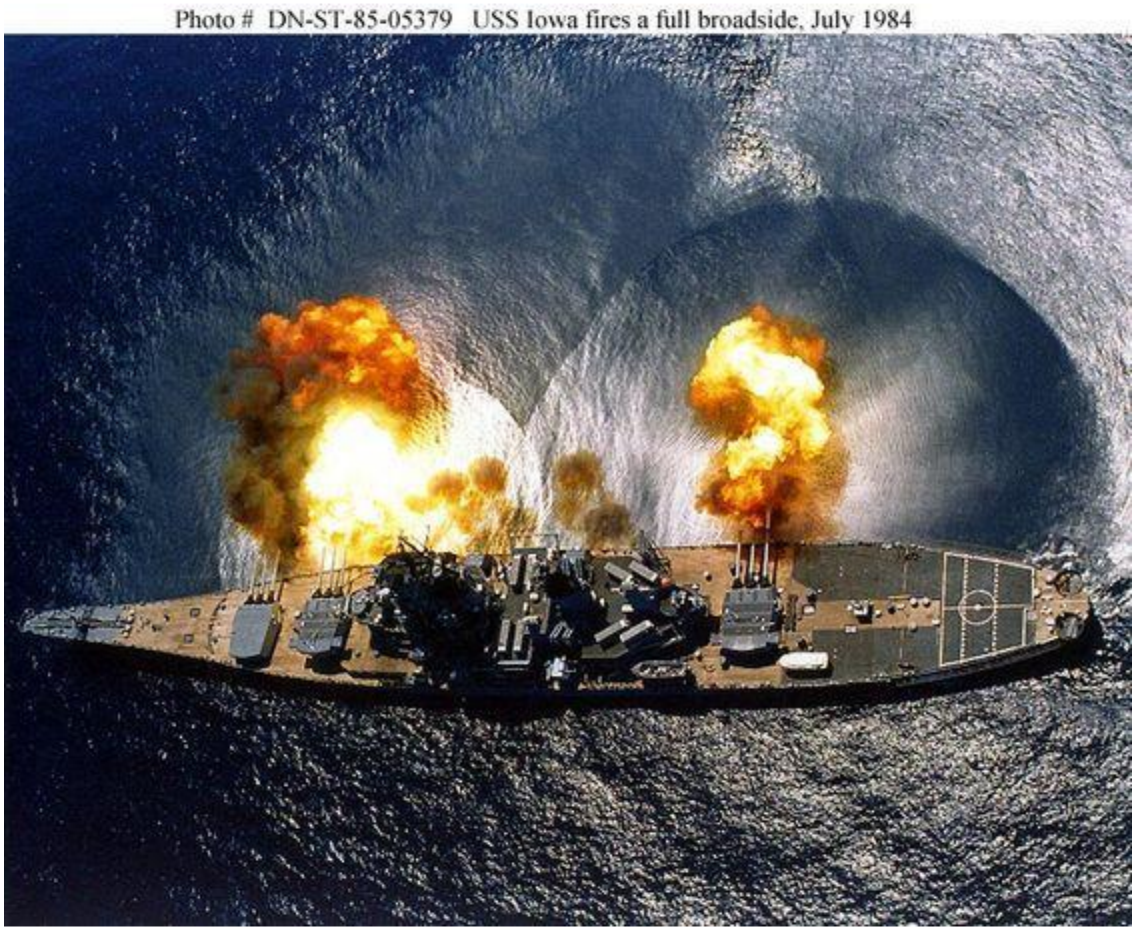


Photo # DN-ST-85-05379 USS Iowa fires a full broadside, July 1984

It’s kinda like that.

Generating The Test Battery

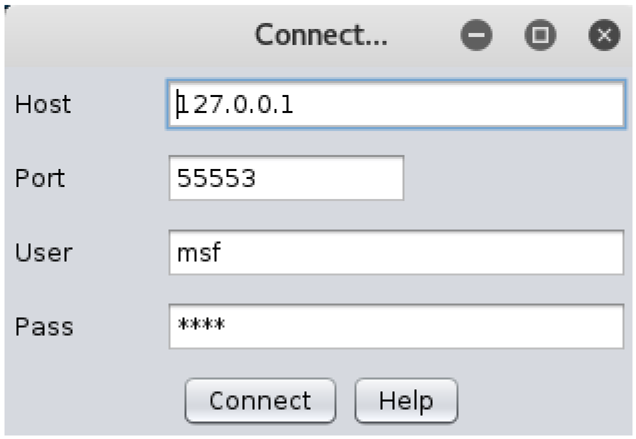
Log in to the console of your Kali VM, and open the terminal application. Execute the following commands:

```
msfdb init
service postgresql start
```

These commands are responsible for initializing and setting up the postgres database backend that the Metasploit framework relies on for storing information during standard operation. The above commands may take a moment or two to finish, depending on the speed of the drive the VM is stored on, and the amount of resources available to the Kali VM. Upon completion of the previous commands, run the command

```
armitage
```

This starts the Armitage GUI for the Metasploit framework. You are greeted by a dialogue box entitled *Connect…* . The default setting of 127.0.0.1 on port 55553 with the default username and password should be fine; simply click the *Connect* button to continue.



Connect...

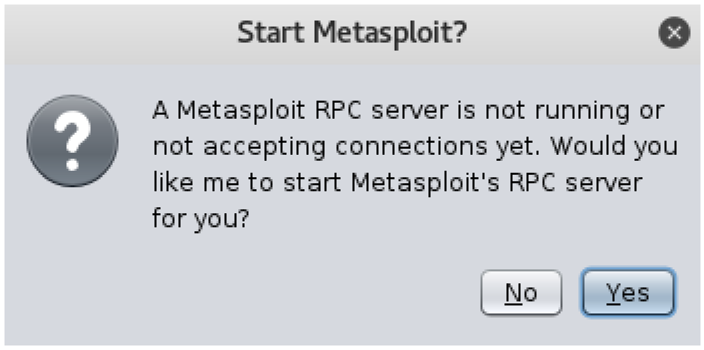
Host: 127.0.0.1

Port: 55553

User: msf

Pass: ****

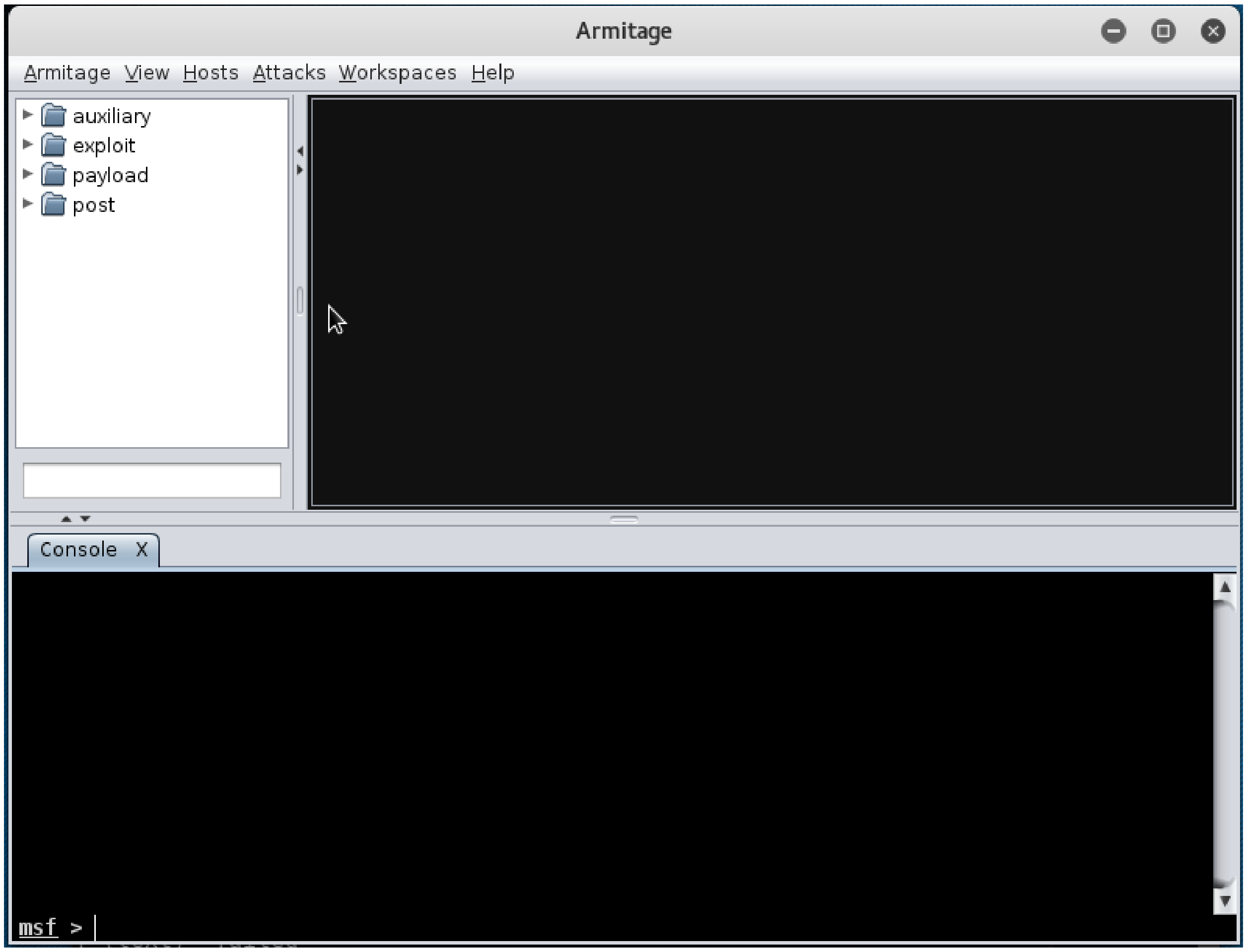
While Armitage is starting, it may ask you if you'd like it to start metasploit/msfrpcd.



Start Metasploit?

A Metasploit RPC server is not running or not accepting connections yet. Would you like me to start Metasploit's RPC server for you?

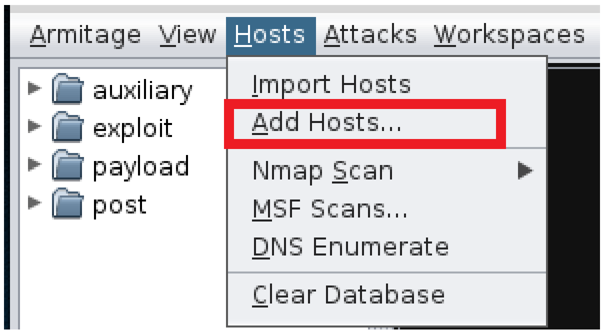
Click the *Yes* button, and allow the Armitage GUI to load.



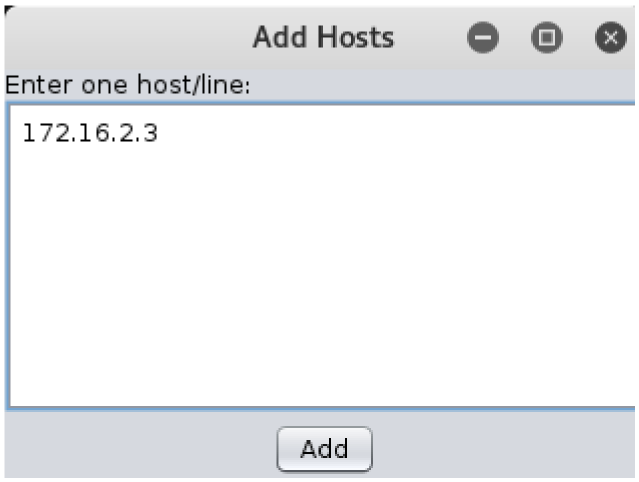
```
msf >
```

Armitage is best described as a wrapper for the Metapsloit framework. Metasploit in and of itself is a tremendously useful tool for offensive security professionals, mainly because it provides a massive, centralized database of exploits and payloads, but also because of its ease of use. Armitage even improves the usability of the Metasploit framework by making it way simpler to perform certain functions, while making it possible for die-hard Metasploit command line enthusiasts to keep using the CLI framework to their hearts' content.

On the menu bar along the top of the Armitage window, click on *Hosts*, and select *Add Hosts…* from the drop-down list.

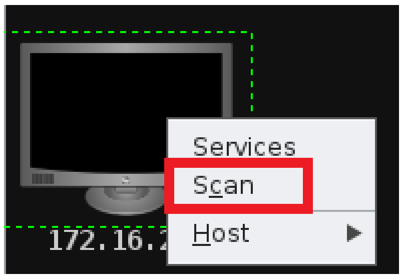


An input box will pop up, asking you to enter IP addresses of targets you want to add. Insert 172.16.2.3 on a single line, then click *Add*.

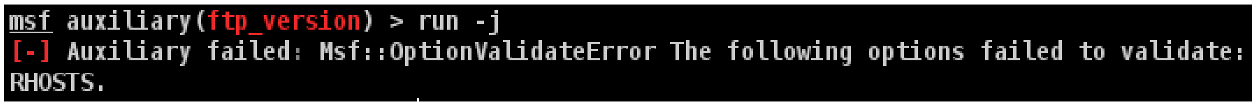


```
172.16.2.3
```

An icon showing a computer screen will pop up in the main window. If you right click on it, a small selection of options you have available right now appears. From that list, select the *Scan* item.



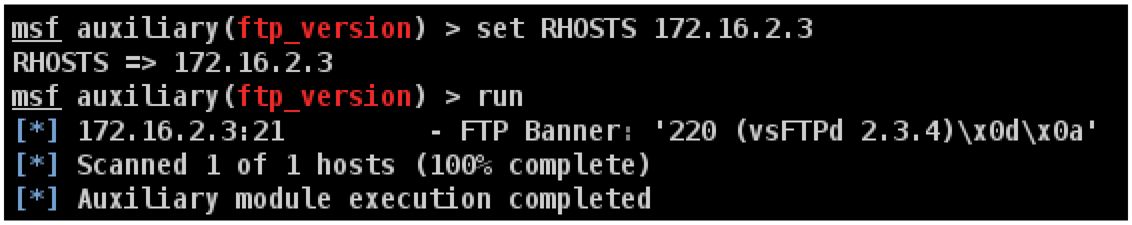
Scan runs several probes against the specified system to identify open ports and running services. However, there is an issue with some of the probes that are ran, where it won't properly pass the IP address of the target host from Armitage to the Metasploit framework correctly. You may see error messages like the one shown in the image below pop up in the *scan* tab.



```
msf auxiliary(ftp_version) > run -j
[-] Auxiliary failed: Msf::OptionValidateError The following options failed to validate:
RHOSTS.
```

In that case, you need to run the following commands in the *scan* tab of the Armitage window for each service/banner grab that the tool attempts to perform:

```
set RHOSTS 172.16.2.3
run
```

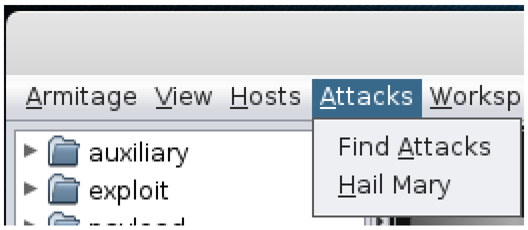


```
msf auxiliary(ftp_version) > set RHOSTS 172.16.2.3
RHOSTS => 172.16.2.3
msf auxiliary(ftp_version) > run
[*] 172.16.2.3:21         - FTP Banner: '220 (vsFTPd 2.3.4)\x0d\x0a'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

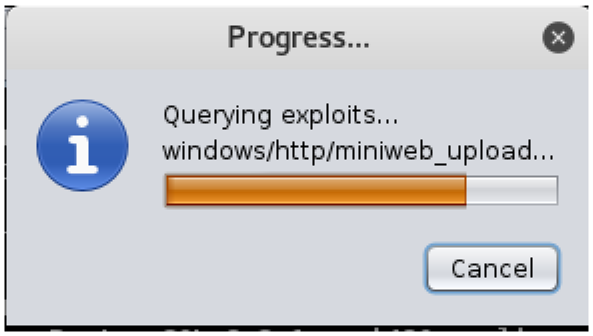
Note: Some pentesters may immediately say `setg` is supposed to be used in order to set RHOSTS globally, but in my experience, this doesn't resolve the issue.

Armitage runs 7 of Metasploit's auxiliary banner grabs/service scans, namely FTP(21/tcp), SSH(22/tcp), Telnet(23/tcp), SMTP(25/tcp), HTTP(80/tcp), SMB(445/tcp), java rmi(1099/tcp), mysql(3306/tcp), postgres(5432/tcp).

While it's annoying to have to reset the RHOSTS variable for each of those scans, running them is important for the next phase of our test. As soon as the scanning is finished, click "*Attacks*" in the Armitage menu, then select *Find Attacks* from the drop-down list.

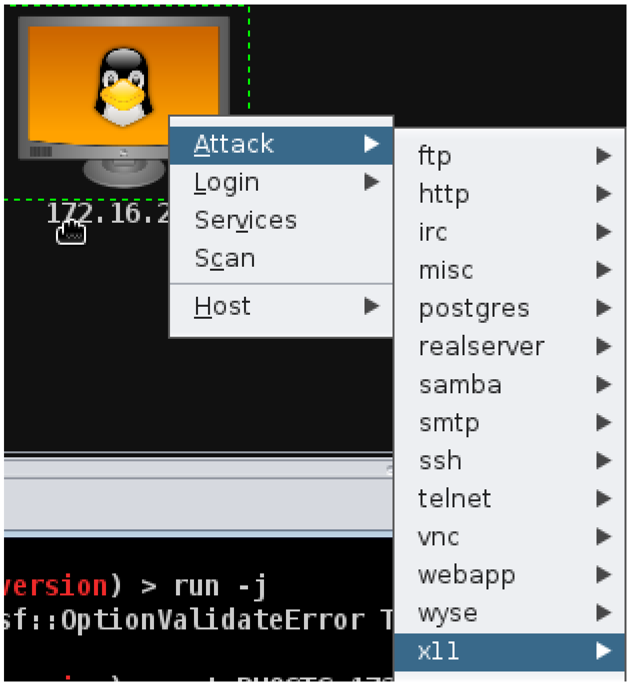


Thanks to the port scans and service detection we did, Armitage is able to make a bunch of attack/exploit recommendations.



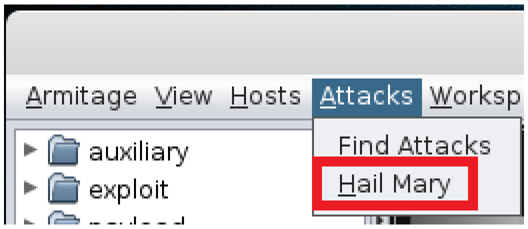
Querying exploits...
windows/http/miniweb_upload...

This takes a moment or two to complete, but when it's done, it'll notify you. If you right click on the computer (that now has a "Tux" penguin in the center, denoting it as a Linux system), you'll have an attack menu available. These are *possible* attacks you can use against this system. The exploits mapped are NOT guaranteed to work. Don't ever rely on *Find Attacks* to be accurate or provide you with 100% correct exploits to a given target host. Ever.

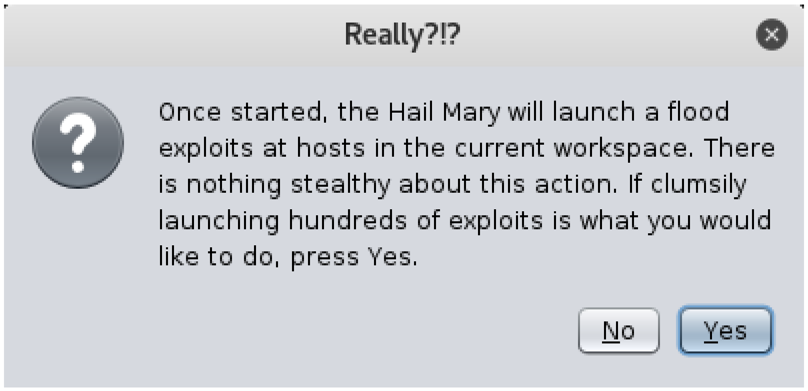


```
ersion) > run -j
sf::OptionValidateError T
```

Now, we could try running each one of these exploits individually, taking our sweet time, or, considering that the goal of this exercise is to generate as much noise as possible, fire as many of them as we can at once. There was a feature in the metasploit framework called db_autopwn. In Armitage, this feature has been re-created as *Hail Mary*. This feature can be found under the Attacks menu, underneath the *Find Attacks* option.

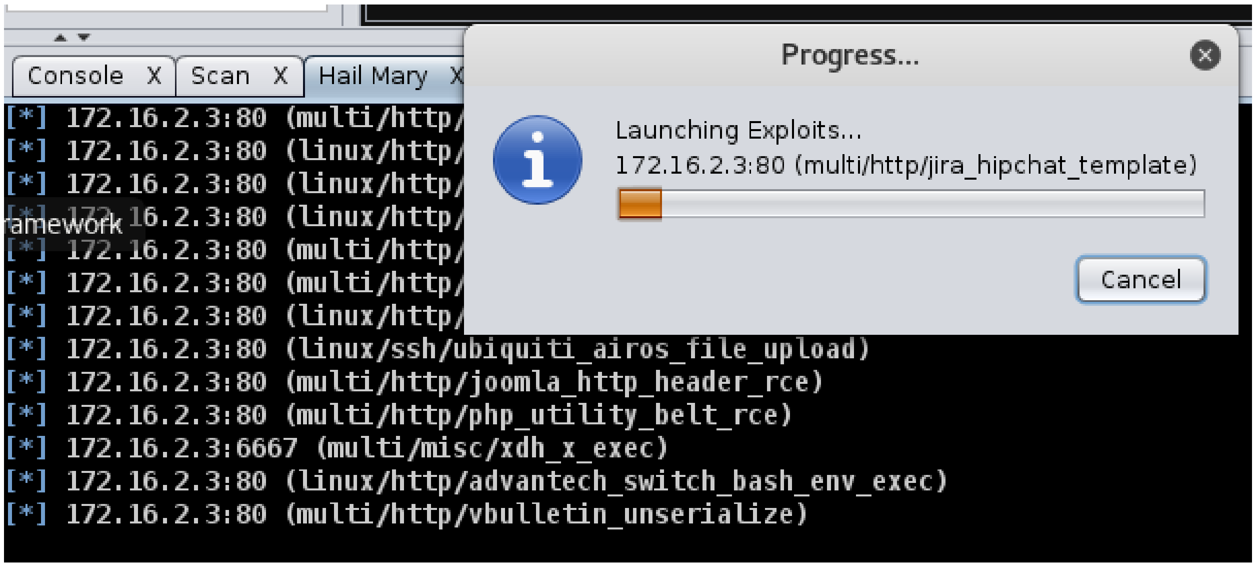


When you run the Hail Mary, you are throwing everything you can at this host with absolutely no regard for stealth or tradecraft; Armitage even warns you when you try to run it, and asks if you are absolutely sure you want to do this.



Once started, the Hail Mary will launch a flood exploits at hosts in the current workspace. There is nothing stealthy about this action. If clumsily launching hundreds of exploits is what you would like to do, press Yes.

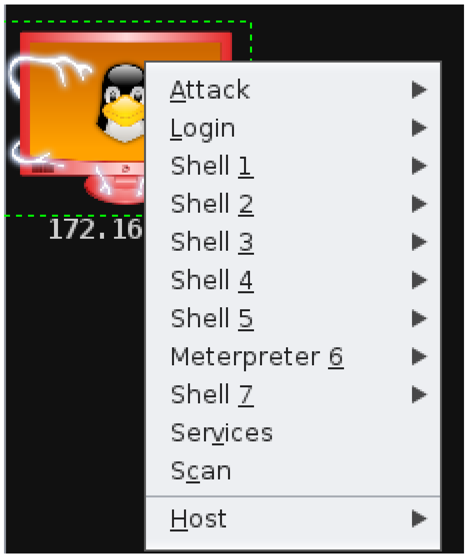
This is what we're looking for; to generate as much noise as we can, with minimum effort, so say *Yes*. This will take some time, and you will see a progress bar go by as Armitage keeps "throwing" all the exploits it can map to our metasploitable host.



Launching Exploits...
172.16.2.3:80 (multi/http/jira_hipchat_template)

```
[*] 172.16.2.3:80 (multi/http/
[*] 172.16.2.3:80 (linux/http/
[*] 172.16.2.3:80 (linux/http/
[*] 172.16.2.3:80 (linux/http/
[*] 172.16.2.3:80 (multi/http/
[*] 172.16.2.3:80 (multi/http/
[*] 172.16.2.3:80 (linux/http/
[*] 172.16.2.3:80 (linux/ssh/ubiquiti_airos_file_upload)
[*] 172.16.2.3:80 (multi/http/joomla_http_header_rce)
[*] 172.16.2.3:80 (multi/http/php_utility_belt_rce)
[*] 172.16.2.3:6667 (multi/misc/xdh_x_exec)
[*] 172.16.2.3:80 (linux/http/advantech_switch_bash_env_exec)
[*] 172.16.2.3:80 (multi/http/vbulletin_unserialize)
```

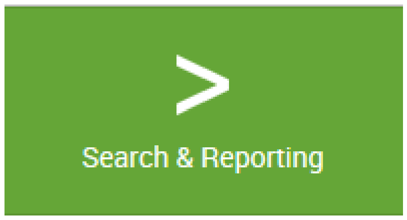
The aftermath is pretty bloody. You'll end up with at least half a dozen shells on the Metasploitable 2 host that you can interact with. If you right click on our target system, you'll see several prompts for `shell #` and/or `meterpreter #` for each successful exploit that resulted in remote code execution on the host.



At this point, this should have been more than enough noise to cause the IDS to flag some alerts. Log in to the splunk search head/index at https://172.16.1.3:8000. At this point, depending on whether you have Snort or Suricata installed, you will be jumping to Verifying Results with Snort or Verifying Results with Suricata.

Verifying Results with Snort

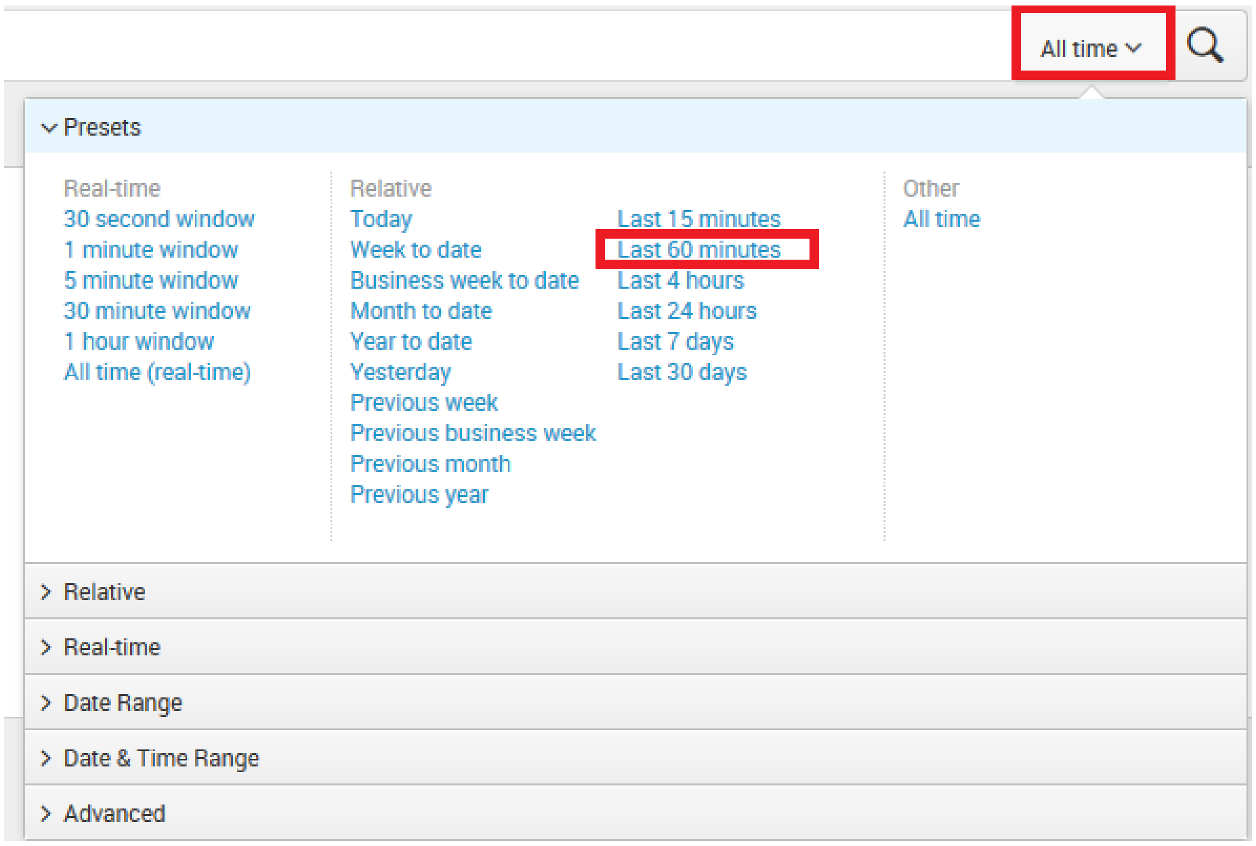
Once you are logged in on the home page, click the “search & reporting” app on the left side of the browser window.



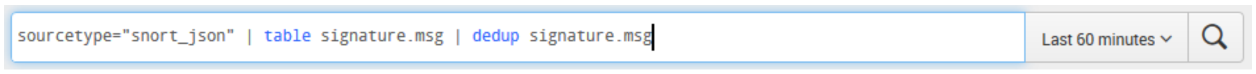
On the next page, you will see a search bar in which you can enter a query for information. Input the following query

```
index=main sourcetype=snort_json | table signature.msg | dedup signature.msg
```

Do not click the spyglass just yet to begin the search, because now we're going to adjust the timeframe. To the right if the input box for search, there is a small drop down labeled *All time* (by default). You can click the drop-down and choose a time range for Splunk to search for certain events. Assuming you just completed the previous section, and launched the *Hail Mary* attack, this activity should have been less than 60 minutes ago, so you may want to modify the time range, and set it to *Last 60 minutes*. Failing that, adjust the time/date range to suit when you actually launched your *Hail Mary* attack against the Metasploitable 2 VM.

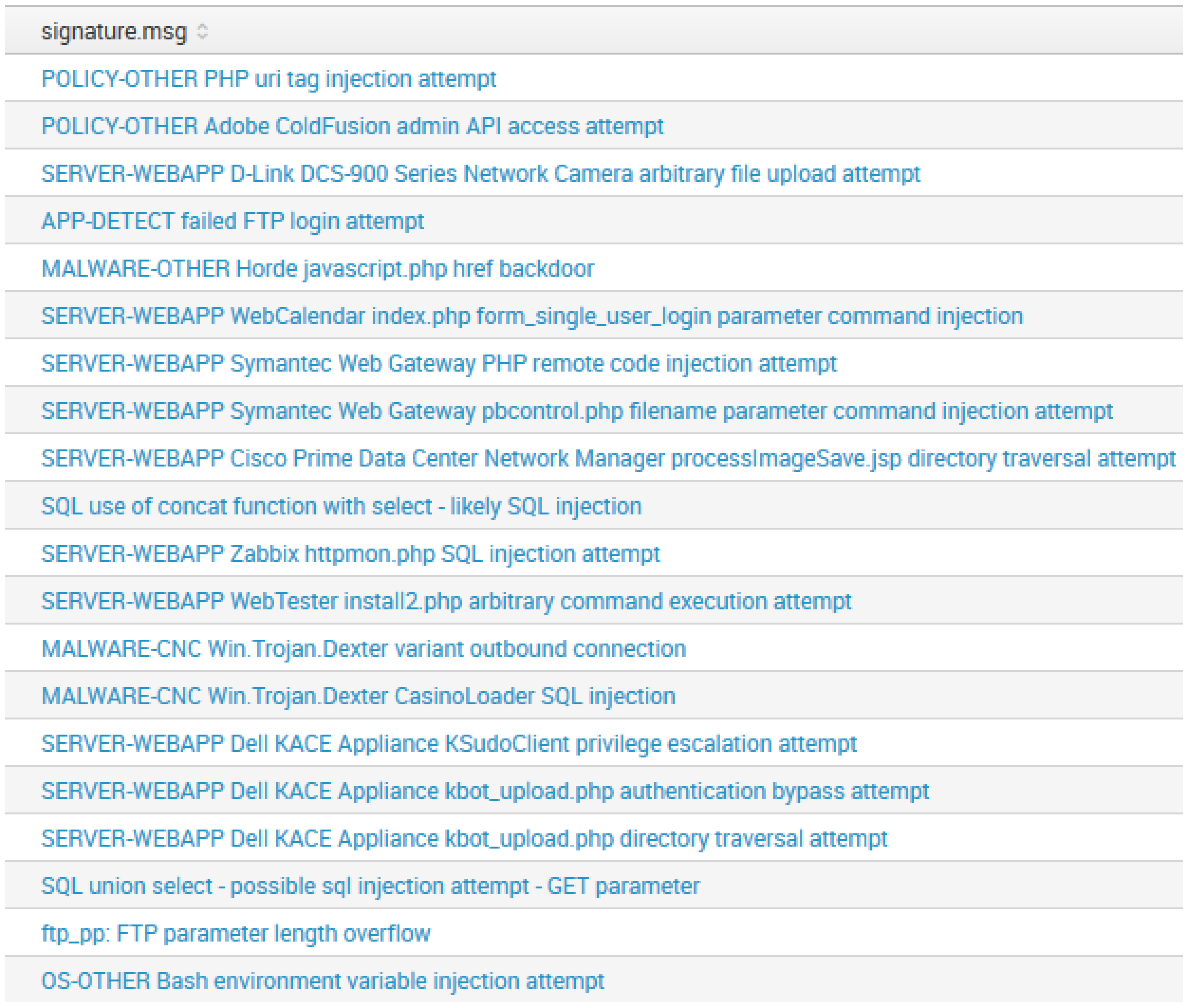


Your query should look like this when you are finished:



```
sourcetype="snort_json" | table signature.msg | dedup signature.msg
```

Allow Splunk some time to find the data and display it. What we're doing is looking in the `main` index, under the `snort_json` sourcetype being fed from our IPS sensor. Then we're looking specifically for the contents of the "signature.msg" field. Then we're saying show me only the data in the `signature.msg` field. Then we're taking the `signature.msg` field, and removing any duplicate entries. The entirety of this query only applies to data collected within the last 60 minutes.

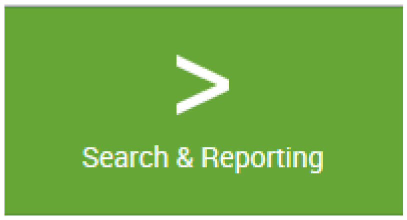


signature.msg
POLICY-OTHER PHP uri tag injection attempt
POLICY-OTHER Adobe ColdFusion admin API access attempt
SERVER-WEBAPP D-Link DCS-900 Series Network Camera arbitrary file upload attempt
APP-DETECT failed FTP login attempt
MALWARE-OTHER Horde javascript.php href backdoor
SERVER-WEBAPP WebCalendar index.php form_single_user_login parameter command injection
SERVER-WEBAPP Symantec Web Gateway PHP remote code injection attempt
SERVER-WEBAPP Symantec Web Gateway pbcontrol.php filename parameter command injection attempt
SERVER-WEBAPP Cisco Prime Data Center Network Manager processImageSave.jsp directory traversal attempt
SQL use of concat function with select - likely SQL injection
SERVER-WEBAPP Zabbix httpmon.php SQL injection attempt
SERVER-WEBAPP WebTester install2.php arbitrary command execution attempt
MALWARE-CNC Win.Trojan.Dexter variant outbound connection
MALWARE-CNC Win.Trojan.Dexter CasinoLoader SQL injection
SERVER-WEBAPP Dell KACE Appliance KSudoClient privilege escalation attempt
SERVER-WEBAPP Dell KACE Appliance kbot_upload.php authentication bypass attempt
SERVER-WEBAPP Dell KACE Appliance kbot_upload.php directory traversal attempt
SQL union select - possible sql injection attempt - GET parameter
ftp_pp: FTP parameter length overflow
OS-OTHER Bash environment variable injection attempt

The output, while it might not have captured every single attack, and may not be 100% accurate, confirms that Snort is configured properly, and that IPS VM is sending its data to Splunk via the Hurricane Labs Unified 2 Add-on, and universal forwarder. Tuning your IDS and modifying the ruleset is beyond the scope of this guide.

Verifying Results with Suricata

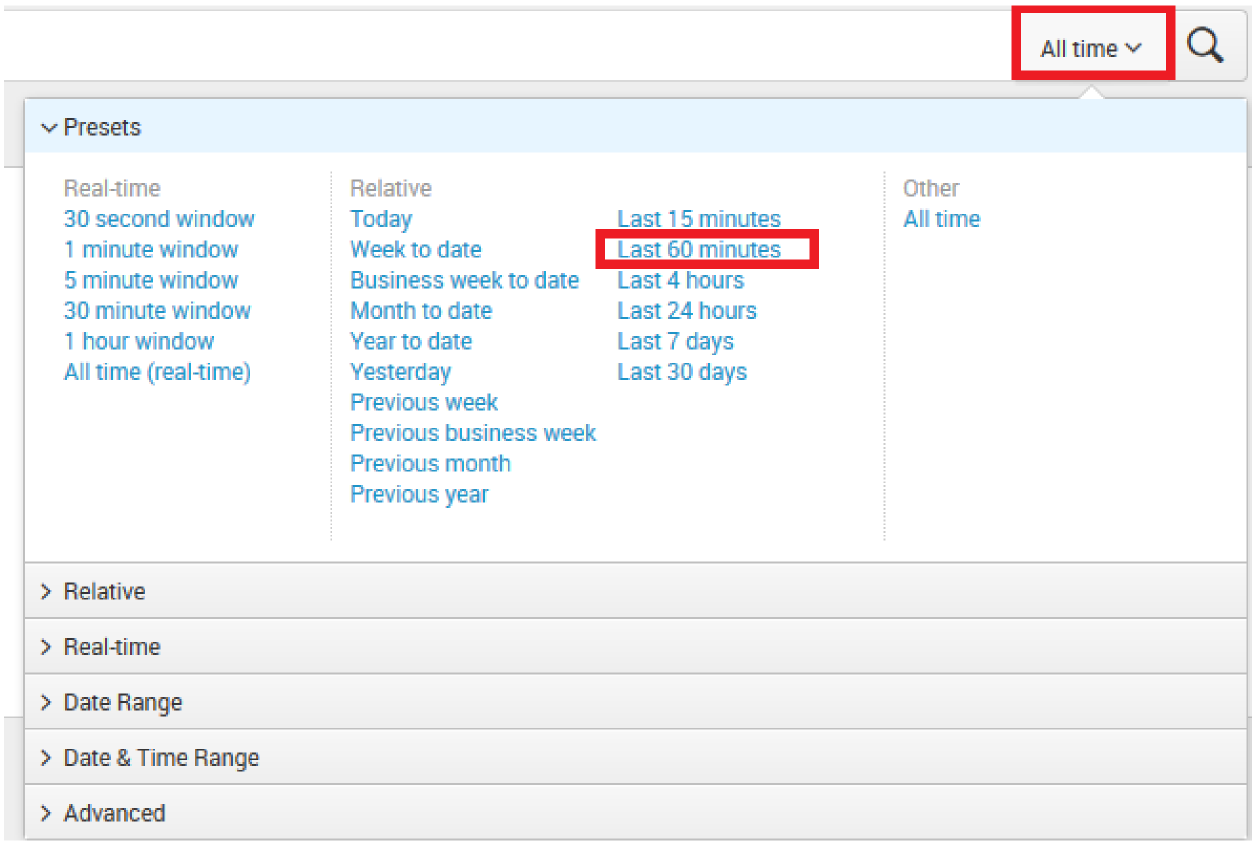
Once you are logged in on the home page, click the *Search & Reporting* app on the left side of the browser window.



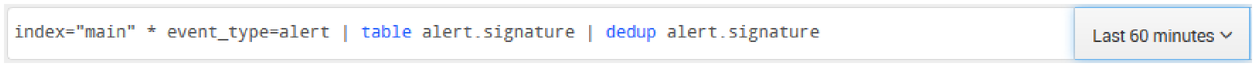
On the next page, you will see a search bar in which you can enter a query for information. Input the following query:

```
index=main * event_type=alert | table alert.signature | dedup alert.signature
```

Do not click the spyglass just yet to begin the search, because now we're going to adjust the timeframe. To the right if the input box for search, there is a small drop down labeled *All time* (by default). You can click the drop-down and choose a time range for Splunk to search for certain events. Assuming you just completed the previous section, and launched the *Hail Mary* attack, this activity should have been less than 60 minutes ago, so you may want to modify the time range, and set it to *Last 60 minutes*. Failing that, adjust the time/date range to suit when you actually launched your *Hail Mary* attack against the Metasploitable 2 VM.

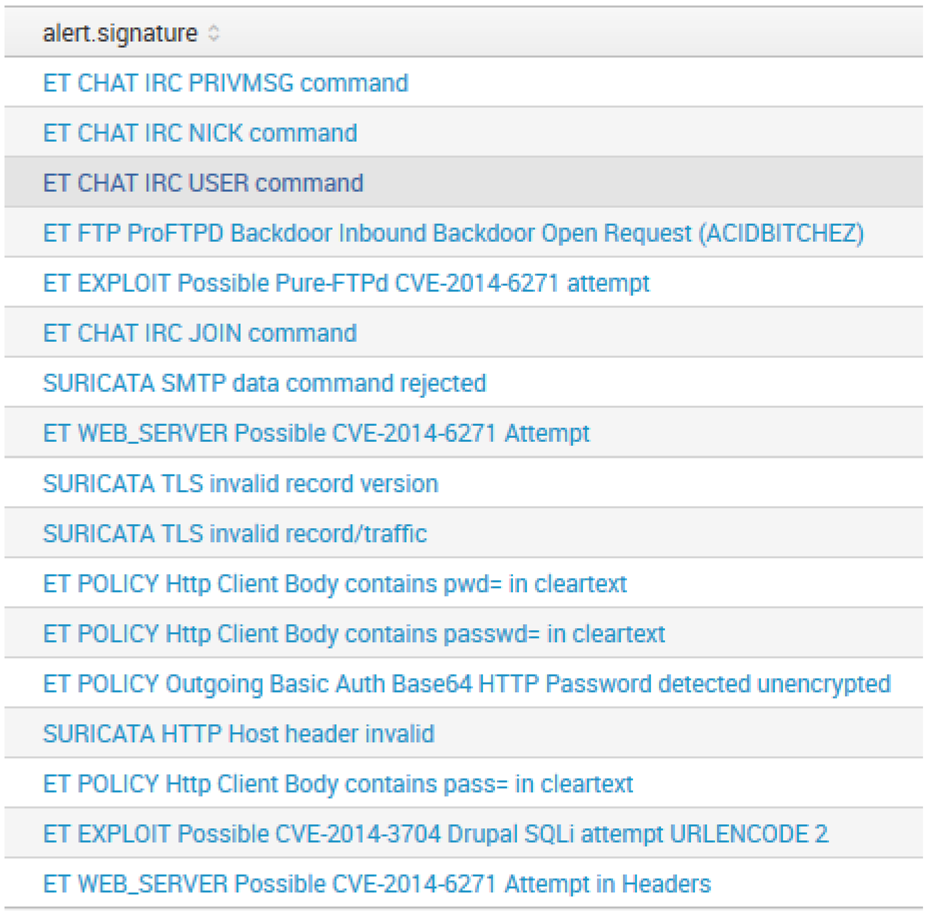


My query looked like this when I was finished:



```
index="main" * event_type=alert | table alert.signature | dedup alert.signature
```

Allow Splunk some time to find the data and display it. What we're doing is looking for data in the `main` index. Then we're looking specifically looking for any records that have the `event_type` field set to `alert` (denoting that it is a Suricata IDS alert). Then we're saying show me only the data in the `alert.signature` field. Then we're taking the `alert.signature` field, and removing any duplicate entries. The entirety of this query only applies to data collected within the last 60 minutes.



alert.signature
ET CHAT IRC PRIVMSG command
ET CHAT IRC NICK command
ET CHAT IRC USER command
ET FTP ProFTPD Backdoor Inbound Backdoor Open Request (ACIDBITCHEZ)
ET EXPLOIT Possible Pure-FTPd CVE-2014-6271 attempt
ET CHAT IRC JOIN command
SURICATA SMTP data command rejected
ET WEB_SERVER Possible CVE-2014-6271 Attempt
SURICATA TLS invalid record version
SURICATA TLS invalid record/traffic
ET POLICY Http Client Body contains pwd= in cleartext
ET POLICY Http Client Body contains passwd= in cleartext
ET POLICY Outgoing Basic Auth Base64 HTTP Password detected unencrypted
SURICATA HTTP Host header invalid
ET POLICY Http Client Body contains pass= in cleartext
ET EXPLOIT Possible CVE-2014-3704 Drupal SQLi attempt URLENCODE 2
ET WEB_SERVER Possible CVE-2014-6271 Attempt in Headers

The output, while it might not have captured every single attack, and may not be 100% accurate, confirms that Suricata is configured properly, to send its data to Splunk via the Suricata TA, and universal forwarder. Tuning your IDS and modifying the ruleset is beyond the scope of this guide.

In Your Own Image

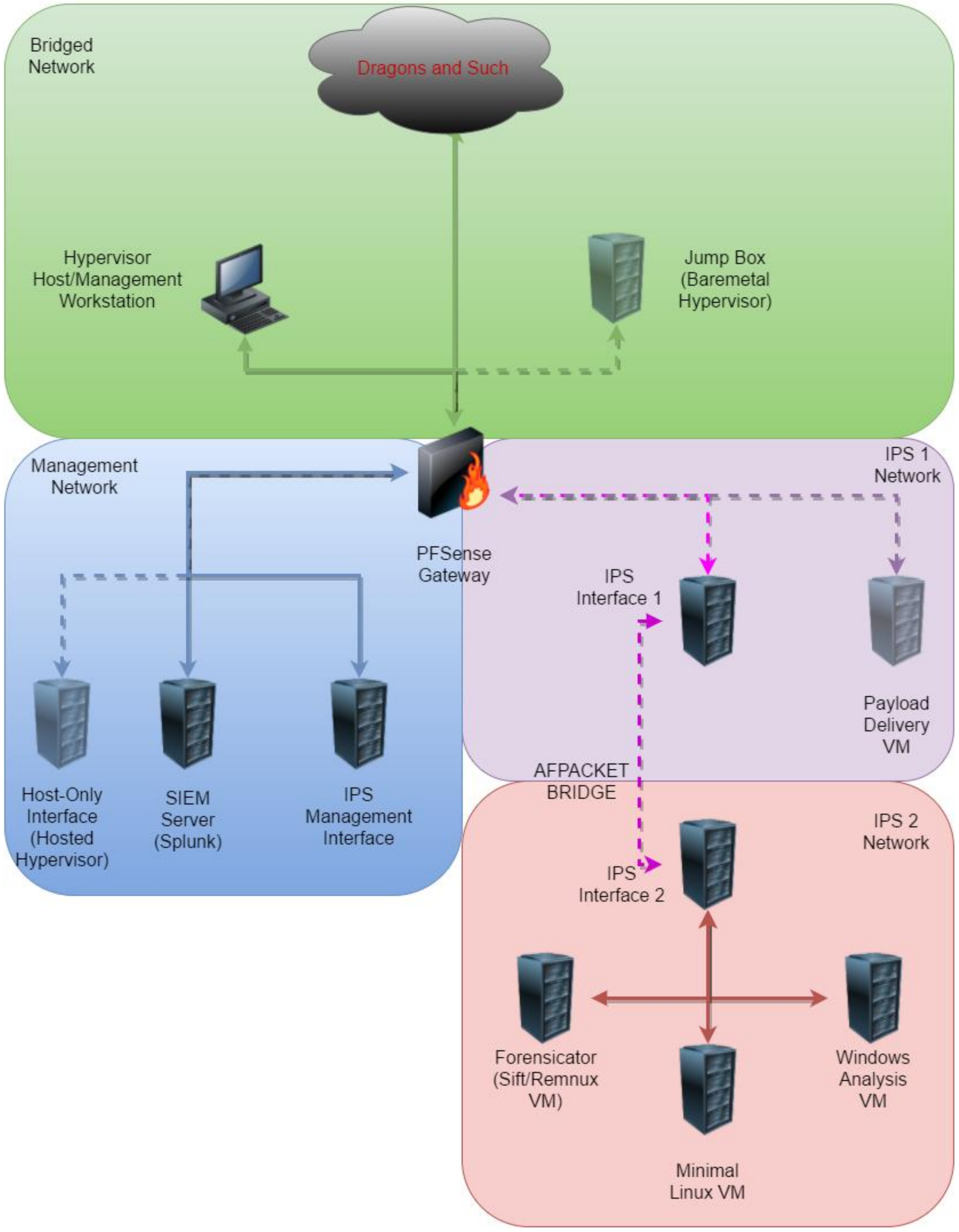
Congratulations. With the creation of your VM lab, the world of information security or information technology study and research is well within your grasp. Where you take your lab environment from here is entirely up to you, and depends on what purpose you want your lab to serve. The purpose of this lab design I taught you how to build is for it to be multi-purpose; capable of filling multiple roles for IT and information security professionals looking for a secure, self-contained environment in order to practice their trade.

What if you don’t like Splunk, and would like to use an ELK (Elasticsearch, Logstash, Kibana) for logging/SIEM instead? What if you don’t want to run your services on Ubuntu, but instead would rather use Redhat or another Linux distro? What if you don’t want to use pfSense, but would rather use OPNSense, Vyatta, or Untangle as your firewall distribution instead? These choices are yours, and yours alone to make. I chose Ubuntu as our Linux distribution of choice for building the lab baseline, because the LTS releases of Ubuntu Server (currently at 16.04) have a long support cycle of five years. I chose pfSense because it has a long development history and a rich featureset. If you would rather use other distributions in building your lab network to better serve your needs or preferences, that is totally up to you.

The goal of this exercise is to teach you how to create, install, and configure VMs and virtual networks across a variety of common and popular hypervisors used all over the world today. Once you have achieved that knowledge, there is no limit (aside from hardware, of course) to how you can configure your VM lab; you can use whatever operating systems you want and make the lab as complex or as minimal as you like. This chapter is dedicated to giving you a few ideas towards modifying your lab to better suit your needs.

Visions of What Might Be

It's all well and good to tell you that the possibilities are endless, but it's another thing altogether to actually show you and give you ideas. Below are a couple of network diagrams that you could use as a springboard to develop/design your own lab network.


Bridged Network
Dragons and Such
Hypervisor Host/Management Workstation
Jump Box (Baremetal Hypervisor)
Management Network
PFSense Gateway
IPS 1 Network
IPS Interface 1
Payload Delivery VM
Host-Only Interface (Hosted Hypervisor)
SIEM Server (Splunk)
IPS Management Interface
AFPACKET BRIDGE
IPS 2 Network
IPS Interface 2
Forensicator (Sift/Remnux VM)
Windows Analysis VM
Minimal Linux VM

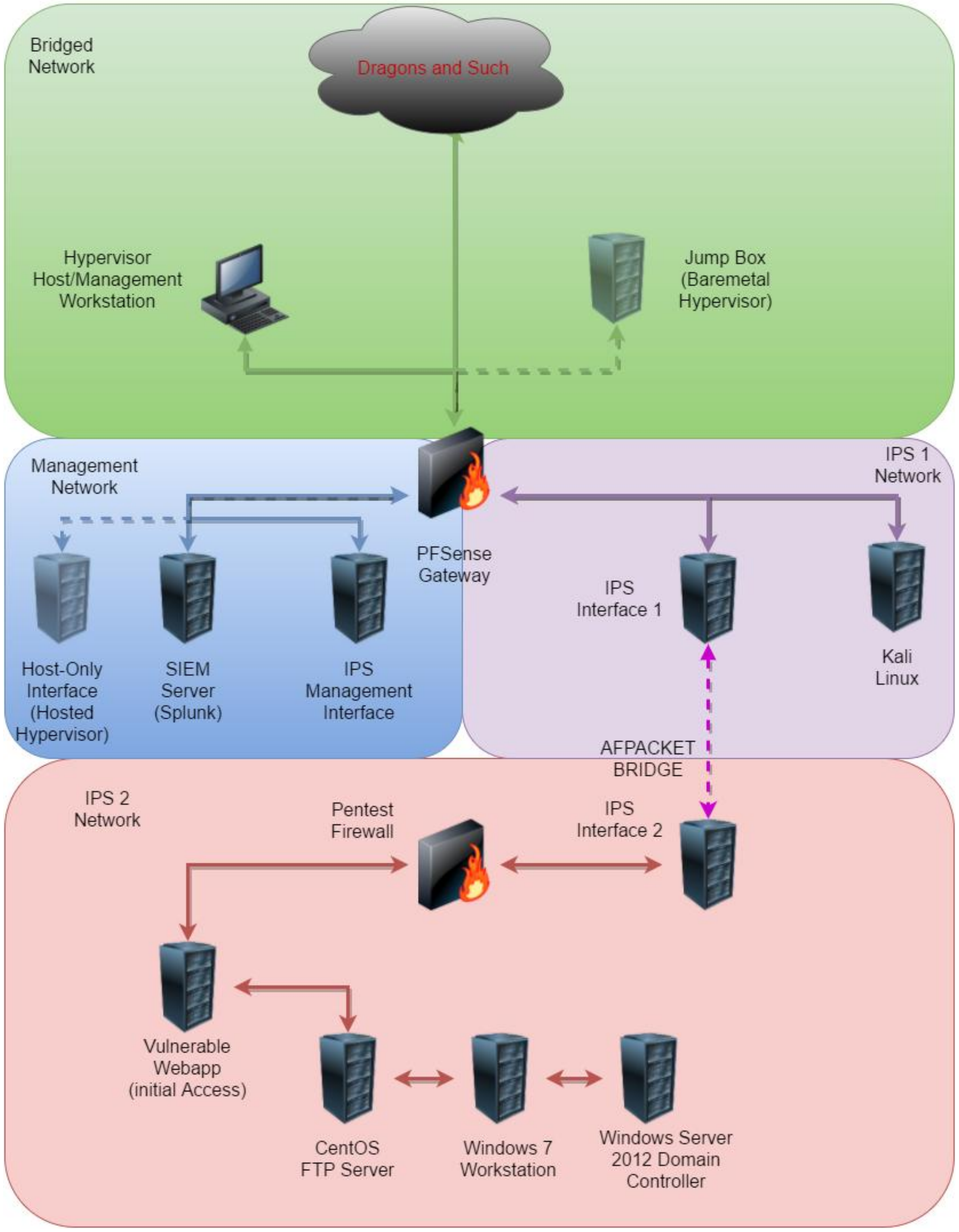
Fortunately for you, If you wanted to build a malware analysis lab out of the baseline configuration we set up, there isn't much you have to change. You can remove the Metasploitable 2 and Kali Linux VMs entirely, since they won't be particularly helpful for malware analysis. In the diagram above on the *IPS 1* network, I have a *Payload Delivery VM* that I specify as being an optional component (transparent). The idea behind this extra VM is that if you have a malware sample you downloaded from a physical system or otherwise need to transfer to the analysis network, you can use this VM as a diode, or halfway point for delivering it to the malware network. Barring that, the VM could be used to download samples via wget, etc. This VM would ideally be either a Linux or BSD VM that is as minimal as possible -- very similar to how a jump box or bastion host would be set up.

The *IPS 2* network is where most of the action will be taking place, and where your malware analysis VMs should live. The VM labeled "Forensicator" is a specialty VM consisting of packages from a pair of special Linux distributions provided by the SANS organization. One of these distros is "SIFT", and is loaded with digital forensics and incident response tools, while the other distro is "Remnux" and is loaded with malware analysis and reverse engineering tools. To go along with our Linux analysis VM, *IPS 2* would also host a Windows analysis VM for dynamic analysis (sometimes known as "sandboxing"), or running specialty malware analysis tools that are simply Windows only.

Finally, the *IPS 2* network also hosts a minimal Linux VM. On Linux, there are several tools you can use to trace system activity as it is going on. Obviously, the more applications and services installed on a system, the more system activity you will have to filter out. Having a VM available with a minimal number of services installed allows you to monitor system activity with little to no overhead to worry about filtering out. This could come in handy for trying to perform dynamic analysis of Linux malware, to determine what changes to the system are being made when you execute a given malicious payload. Alternatively, you could potentially have this system perform double duty as your malware payload delivery VM, and do away with the payload delivery VM in the *IPS 1* network altogether.

Other possibilities for your analysis lab would be to possibly allocate more resources to your IPS VM and install a second network monitoring package on the system, such as Bro IDS, or perhaps run a full packet capture tool such as Moloch or even just tcpdump to gather all of the network traffic coming from the *IPS 2* network to try and capture command and control traffic, beaconing, or other network artifacts beneficial to malware analysis.

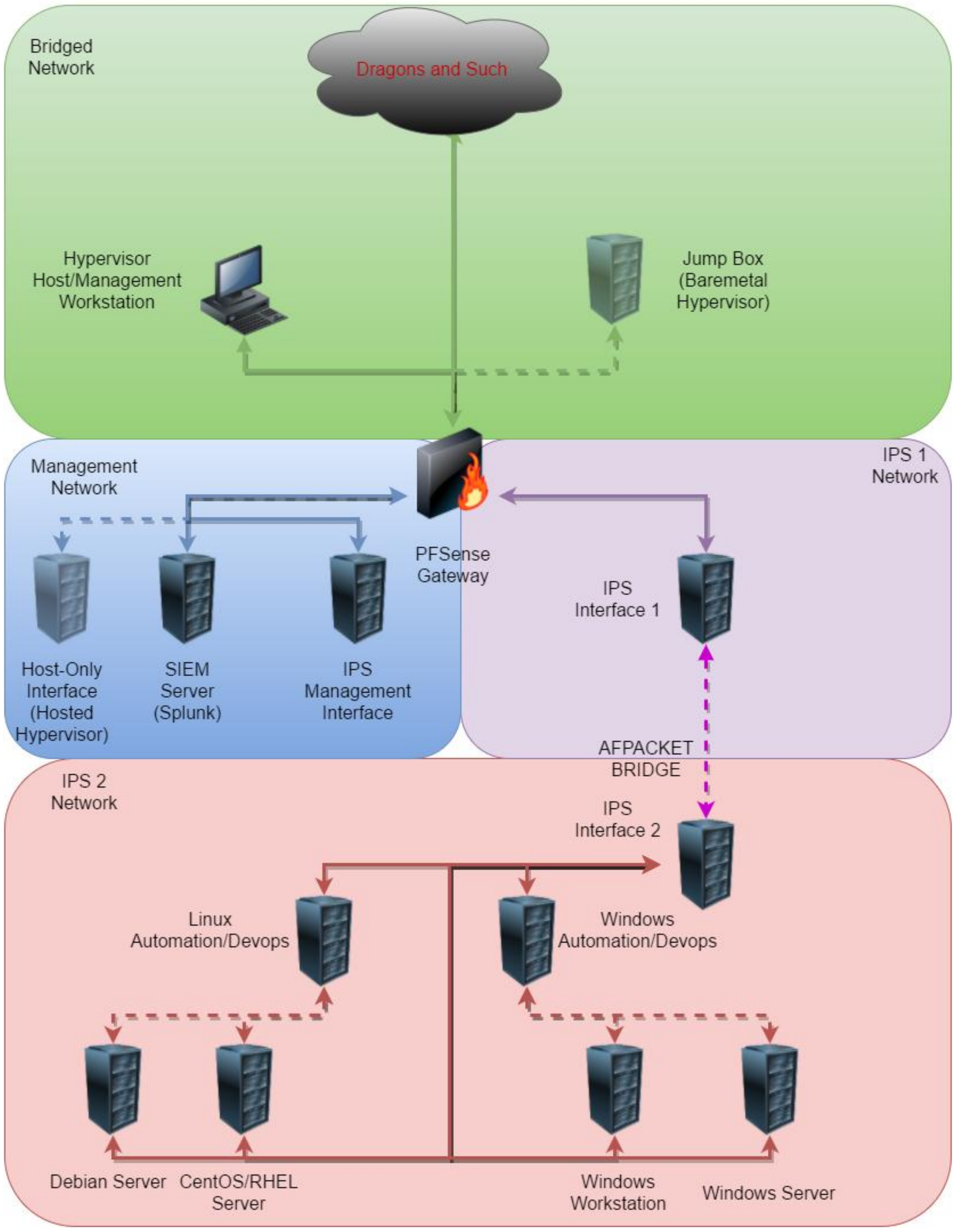
Depending on the number of VMs deployed or resources reallocated, the full lab might require probably around 18-22GB of RAM, somewhere around 650GB of disk, and a quad core CPU (or better) to fully support. Again, this depends on how you allocate resources and how many additional VMs you create. You may be able to get away with fewer resources, provided you have some of the VMs turned off when they are not in use (e.g. turn of the Windows Analysis VM when looking at Linux malware, and vice-versa).


Bridged Network
Dragons and Such
Hypervisor Host/Management Workstation
Jump Box (Baremetal Hypervisor)
Management Network
PFSense Gateway
IPS 1 Network
Host-Only Interface (Hosted Hypervisor)
SIEM Server (Splunk)
IPS Management Interface
IPS Interface 1
Kali Linux
AFPACKET BRIDGE
IPS 2 Network
Pentest Firewall
IPS Interface 2
Vulnerable Webapp (initial Access)
CentOS FTP Server
Windows 7 Workstation
Windows Server 2012 Domain Controller

They told me red team was sexy, and I bought into it. So now I have to build a pentesting lab to learn to be leet. Depending on your skill level, technically, you could download the Metasploitable 2 exploitation guide (https://community.rapid7.com/docs/DOC-1875), and get started immediately. Our baseline VM lab includes Kali Linux and Metasploitable 2, which is practically everything you need to get started. Even though Metasploitable 2 is considerably old (and Metasploitable 3 is now available), Metasploitable 2 is still a perfectly acceptable starting point to learning about exploitation and post-exploitation operations once you have successfully gained control of a system. Not only is this a perfectly good starting point to learn the basics of exploitation, you also have an IPS configured to generate alerts on the attacks you perform against the Metasploitable 2 VM. As a red teamer, it is important to understand what the blue team sees, or what possible alerts may be generated by your actions in order gain better tradecraft and improve your skills as a penetration tester. After all, offense informs defense, but the opposite is also true. Consider implementing more network security tools and technology in order to measure the impact of your attacks and operations as you progress in skill.

The lab design I concocted above however, is considerably more complicated than the base lab network we worked on together. The network diagram above consists of another pfSense VM on the IPS 2 network acting as a firewall and DHCP server for the internal IPS 2 network. The firewall would would only allow ports 80 and 443/tcp in and outbound. The only system that would be accessible from the Kali Linux VM on the IPS 1 network would be a system running a vulnerable web application that the attacker would have to figure out how to compromise in order to drop a webshell. From there, the attacker would need to discover how to pivot from this initial system to a Linux FTP server. From there, the attacker would have to discover how to pivot to Windows workstation that the administrator uses, and finally from there, discover how to pivot to the domain controller (usually a major objective in most pentesting engagements due to domain administrator accounts having access to essentially everything in the network). The idea behind this pentesting lab would be to discover which machines (if any) could talk to the internet, which machines the attacker would need to pivot off of, and learning various methods of acquiring elevated privileges in a target network until complete control is achieved. Sniffing network traffic for plaintext credentials, cracking weak passwords, passing the hash, beaconing implants, pivoting, etc.

To fully implement this pentesting lab, you would need somewhere in the neighborhood of around 17-22GB of RAM, and probably around 600+GB of disk space to store the VM files and and snapshots, plus at least a quad core CPU. You might be able to get away with less RAM, but it would be a very tight fit. Bear in mind that this lab network is merely a suggestion. You could choose to implement it in full, part, or go do your own thing entirely; not unlike the malware analysis lab, the possibilities are endless. Consider bookmarking vulnhub.com for ideas, and VMs to add to your pentesting lab in order to test your leetness.



To represent the IT folks and sysadmins in training, I also made a network diagram on how the lab network might look if it were reconfigured to support experimentation with IT Automation, Monitoring and/or Devops tools. For instance, this might be an environment where you'd want to try out spiceworks, nagios, icinga, making your own software repository (e.g. APT mirror, or WSUS, etc.) as well as devops technologies such as containers (e.g. lxc, docker, etc.) and other automation tools (e.g. puppet, chef, ansible, etc.).

Not unlike the pentesting lab, the resources required to run this lab are likely to be extensive. Depending on RAM and disk space allocations, you would likely require around 20-24GB of RAM, and around 750GB to fully support this lab environment, plus the CPU requirements (again, minimum of a quad core CPU). You could probably get away with less resources, so long as you're not running any of the systems particularly hard, or if perhaps you had some VMs powered off while you were experimenting with others.

Summary

In this chapter, we're going to take the time to review all the ground we've covered.

What Have We Learned Today?

Let's review what we've accomplished here so far:

- General hardware requirements have been provided as a starting point for building your own VM lab
- A network diagram illustrating the end result of our work has been provided to grant a better understanding of the goal we worked towards
- Instructions have been provided on how to create and configure these VMs, and virtual networking on five different hypervisors:
    - Oracle VirtualBox
    - VMWare Workstation
    - VMWare Fusion
    - VMWare vSphere Hypervisor (ESXi)
    - Microsoft Client Hyper-V
- The VMs we have created and configured are as follows:
    - A pfSense VM with a set of core network services configured/provided (NTP, DHCP, DNS and Squid proxy services) and strict firewall rules for three separate network zones (Outbound/Bridged, IPS/test, Management/Secure)
    - An Ubuntu VM running either Snort or Suricata in inline mode via afpacket, serving as a network bridge between one segment of our IPS/test network, and the other half of the IPS/test network, as well as a Splunk universal forwarder installation, configured for pushing the IDS alerts from this system generates to our splunk instance
    - An Ubuntu VM running a Splunk indexer/search head where we can query our IDS alerts, and possibly expand our logging system to host additional logs for our lab environment
    - A Kali Linux VM we used to verify the IDS bridge works properly, and that the IDS properly sends IDS alerts to our Splunk instance
    - A metasploitable2 VM we used to serve as a test system in our IPS/test network, as well as to verify that the bridge our IPS VM provides is working properly
- Each hypervisor has 4 Virtual Network segments/Virtual Switches defined:
    - The Bridged/External network, connecting your lab environment to a physical network (by way of the pfSense VM) for internet access, as your lab network

    requires (e.g. software updates, or external access from a management workstation/jump box for bare-metal hypervisors)
    - The Management network, where the primary network interfaces for both the IPS and the SIEM VMs are located
    - The IPS 1 network, one half of the IPS/test network. Hosts the Kali Linux VM
    - The IPS 2 network, the second half of the IPS/test network. Hosts the Metasploitable 2 VM
- The instructions for each hypervisor provide as much isolation between the hypervisor host and the VMs as possible through disabling excess features (e.g. guest extensions, copy and paste, drag and drop, etc.) and removing excess virtual hardware
- The hypervisor virtual networking, as well as the network configuration of the ips VM itself, should be able to support bridging the IPS 1 and IPS 2 network segments via the AFPACKET bridging functionality that Snort or Suricata can provide
- Additional instructions have been provided to secure Windows workstations, and Windows hosted hypervisor systems against possible compromise from lab VMs
- Instructions have been provided to enable remote management (SCP, SSH, key-based authentication, adding static network routes for hosted hypervisors, etc.) for Windows, Linux, OS X, and BSD systems
- Instructions have been provided on how to enable access to VMs directly from a management workstation to VMs hosted on a bare-metal hypervisor
- Instructions have been provided on how to enable access to VMs hosted on a bare-metal hypervisor via a jump box (bastion host) system
- A simple script to automate system updates on Linux VMs has been provided, as well as instructions on how to implement it
- Instructions and scripts to automate the installation of Snort or Suricata have been provided
- Instructions have been provided on how to install a splunk indexer and search head on the SIEM VM, acquire dev licensing for a personal lab (optional), and finally, install a universal forwarder and necessary TAs to support parsing and forwarding ids logs for Snort or Suricata as necessary.
- Instructions have been provided on how to use Armitage and its “Hail Mary” function as a test battery to verify that the IDS is functioning and that data is being logged to the Splunk instance.
- All VMs should have a baseline snapshot to revert each VM to a known-good working state
- You have been given a couple of ideas in the form of general network diagrams on how the baseline lab network could be reconfigured to support red team, blue team, or IT monitoring for the express purpose of learning

Epilogue: We Need You (Now More than Ever)

If you’re still reading at this point, then gods help you, I have no idea why. When I first started writing this guide, which then became known as project AVATAR (based off the plot device of my current favorite video game, XCOM 2), I didn’t intend for it to be a primer on virtual networking or a crash-course on virtualization, but here we are, approximately 10 months later, over 550 pages, 95,000 words, and well over 500 screen captures and illustrations. It’s been a long, strange journey.

I did this primarily because we need guidance for new blood to information technology in general, and information security disciplines, in particular. It doesn’t matter who you are and where you come from, we’re short-staffed, under-resourced, and burnt out. We need you, now more than ever. But, new blood needs guidance, and frankly I’m tired of the blank looks when I tell newbies to “Go build a lab!” and them not having any idea what I’m talking about. Now, there is a somewhat comprehensive starting point that I can jerk a thumb at and say, “Get to work. Come back to me if you have questions.”

Could this book have been made better? Probably. There are several hypervisors I didn’t create guides for, simply because I have never used them or never heard of them. Could I have covered automation and/or devops things? No idea; I have Grampa Simpson syndrome where devops is new and scary to me and I don’t trust it yet. After all, these automated build tools, containers, and crap have a fatal flaw: if the container is broken, or the build process fails catastrophically, often times, there's no effective error-catching where the build process will detect that a step has failed and attempt to retry so many times before giving up. No, sometimes these tools just plow forward without a care in the world until you’re left with a broken mess in an unknown state. Then the developers of said image or container simply shrug, say ‘works on my machine’, and go about their business. This is why I didn’t go into devops and automation in this book. It is my fervent belief that as an IT or security professional, you need to know how to crawl before you walk, and walk before you run. This entails learning how to do things the hard way before being coddled by automation tools doing all of that stuff for you, or having some stranger provide a pre-configured container for you.

At some point, you’re going to be the ones having to fight the good fight and continue to solve IT and Security problems long after I’m in the retirement home, killing ayys while playing XCOM 5. So welcome to IT/Infosec, have fun, and don’t get burnt out. The fight will still be there the next day.

Tony “da_667” Robinson