INSTANT

Burp Suite Starter

Get up and running with Burp Suite using this hands-on practical guide

Luca Carettoni

Instant Burp Suite Starter

Get up and running with Burp Suite using this hands-on practical guide

Luca Carettoni

[PACKT] PUBLISHING

BIRMINGHAM - MUMBAI

Instant Burp Suite Starter

Credits

Author

Luca Carettoni

Reviewers

Claudio Criscione

Luca De Fulgentis

Acquisition Editor

Martin Bell

Commissioning Editor

Harsha Bharwani

Technical Editor

Dominic Pereira

Project Coordinator

Amigya Khurana

Proofreaders

Maria Gould

Mario Cecere

Production Coordinator

Aparna Bhagat

Cover Work

Aparna Bhagat

Cover Image

Sheetal Aute

About the Author

Luca Carettoni is a security researcher with over eight years of experience in the application security field. His professional expertise includes black box testing, web application security, vulnerability research, and source code analysis. He is the Director of Information Security at Addepar, a company that is re-inventing the infrastructure which powers global wealth management.

Prior to Addepar, Luca worked at Matasano Security as a senior security consultant, performing vulnerability research activities on a wide range of systems: from web applications to stand-alone software and mobile applications. In the past years, he has been an active participant in the security community and a member of the **Open Web Application Security Project** (**OWASP**). He holds a Masters Degree in Computer Engineering from the Politecnico di Milano university.

About the Reviewers

Claudio Criscione is a cat tamer and a security expert, even though he often fails to see the difference. He graduated in Milan and worked as a security consultant for large enterprises, focusing on web and virtualization security. He is a believer in full (yet responsible) disclosure and still appreciates the challenges in security, even though he's currently busy scaling security testing on large organizations. He had the chance to be a speaker around the world, yet he now lives in Switzerland.

Luca De Fulgentis is an application security engineer with experience in application penetration testing and source code reviewing. He holds a Master's degree in Computer Engineering from Politecnico di Milano from where he graduated with a thesis on evolutionary fuzzing. He works for Secure Network as a security consultant, where he is involved in penetration testing and security research on web application exploiting techniques and Microsoft Windows Phone.

www.packtpub.com

Support files, eBooks, discount offers and more

You might want to visit `www.packtpub.com` for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.packtpub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

PacktLib.packtpub.com

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

✦ Fully searchable across every book published by Packt

✦ Copy and paste, print and bookmark content

✦ On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at `www.packtpub.com`, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

PACKTLiB

Table of Contents

Instant Burp Suite Starter

Welcome to the Instant Burp Suite Starter. This book has been especially created to provide you with all the information that you need to get set up with Burp Suite. You will learn the basics of Burp Suite, get started with testing your first application, and discover some tips and tricks for using Burp Suite.

This document contains the following sections:

So, what is Burp Suite? – find out what Burp Suite actually is, what you can do with it, and why it's so great.

Installation – learn how to download and set up Burp Suite so that you can use it as soon as possible.

Quick start – this section will show you how to perform one of the core tasks of Burp Suite; intercept HTTP/S requests and perform tampering. Follow the steps to intercept, inspect, and modify HTTP/S traffic between the client and the server.

Top 8 features you need to know about – here, you will learn how to perform eight tasks with the most important features of Burp Suite. By the end of this section you will be able to use the target site map functionality, crawl a web application, launch a scan to detect security vulnerabilities, automate customized attacks, manipulate and iterate web requests, analyze the randomness of application data, decode and encode data in multiple format, and compare site maps in order to detect authorization bugs.

People and places you should get to know – every security project is centered around a community. This section provides you with many useful links to the project page and forums, as well as a number of helpful articles, tutorials, and blogs on Burp Suite.
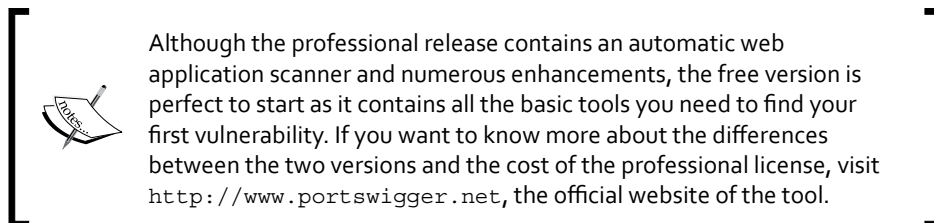
So, what is Burp Suite?

Burp Suite is an easy-to-use integrated platform for web application security. Burp includes multiple tools that are seamlessly integrated and allow you to test every component and aspect of modern web applications. Whether you need to verify the robustness of your authentication mechanism, the predictability of your session tokens, or the input validation checkpoints present in your application, Burp is the Swiss-army knife for security practitioners. Not only does it allow in-depth manual assessments, but it also combines automated techniques to enumerate and analyze web application resources.

Burp has been developed by PortSwigger Ltd. and it is distributed in two editions:

✦ Burp Free
✦ Burp Professional

> Although the professional release contains an automatic web application scanner and numerous enhancements, the free version is perfect to start as it contains all the basic tools you need to find your first vulnerability. If you want to know more about the differences between the two versions and the cost of the professional license, visit `http://www.portswigger.net`, the official website of the tool.

In its essence, Burp is a local web proxy that allows to intercept, inspect, and modify HTTP/S requests and responses between the user's browser and the target website. While the user navigates through the web application, the tool acquires details on all visited pages, scripts, parameters, and other components. The traffic between the browser and the server can be eventually visualized, analyzed, modified, and repeated multiple times. The different tools included in Burp Suite can be easily distinguished by the upper tabs:

✦ **Target**: This tool allows to aggregate all web application resources, thus guiding the user throughout the security test.

✦ **Proxy**: It is the core component of the tool, which allows to intercept and modify all web traffic.

✦ **Spider**: An automatic crawler that can be used to discover new pages and parameters.

✦ **Scanner**: A complete web application security scanner, available in the Professional version only.

✦ **Intruder**: Burp Intruder allows to customize and automate web requests. Repeating multiple times the same request with different content allows to perform **fuzzing**. Web fuzzing typically consists of sending unexpected inputs to the target application. This process may help to identify security flaws.

✦ **Repeater**: A simple yet powerful tool that can be used to manually modify and re-issue web requests.

✦ **Sequencer**: Burp Sequencer is the perfect tool for verifying the randomness and predictability of security tokens, cookies, and more.

✦ **Decoder**: It allows to encode and decode data using multiple encoding schemes (for example, URLencode) or common hash functions (for example, MD5)

✦ **Comparer**: A visual *diff* tool that can be used to detect changes between web pages.

Burp's main window is shown in the following screenshot:
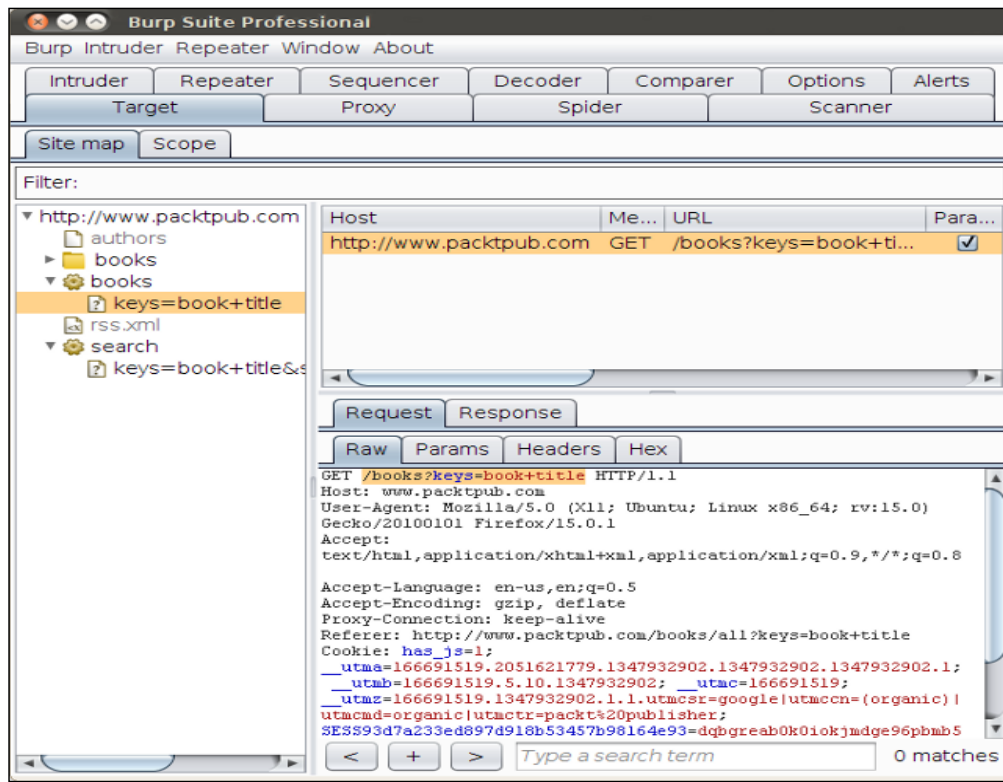
Installation

In a few easy steps, you can set up Burp Suite and your browser.

Step 1 – What do I need?

Before starting, you will need to check the system requirements, as listed here:

- ✦ Disk space: At least 100 MB free. Disk space is required for temporary files, saving the configuration, and your Burp's state files.
- ✦ Memory: At least 2 GB. This amount of memory is usually sufficient. You may need more if you are testing a large application.
- ✦ Operating system: Burp Suite works on Windows, Mac OS X, and Linux
- ✦ Software components: An updated Oracle Java Runtime Environment (v1.6 or later) is required to run Burp Suite. Alternatively, OpenJDK can also be used, although it is not officially supported. Also, make sure to install the latest version of a modern browser (Firefox, Internet Explorer, Chrome, Safari or Opera). The author suggests Mozilla Firefox.

Step 2 – Downloading Burp Suite

Burp Suite can be downloaded as a compressed package from `http://www.portswigger.net/burp/download.html`.

I suggest that you download the free edition and start evaluating the basic functionalities of the software. Eventually, you may decide to purchase a license for the professional version with all advanced features.

After downloading and unpacking the archive, you will be left with a folder containing a Java Archive (JAR) file.

Step 3 – Launching Burp Suite

In the latest free version, available at the time of writing, Burp Suite's JAR is named `burpsuite_v1.4.01.jar`. This executable Java archive can be launched with the following commands.

Windows

1. Click on **Start**.
2. Click **Run**.
3. Type `cmd` and press *Enter*.

4. Move to the folder containing the Burp's JAR using the command `cd` (for example, `cd burpsuite_v1.4.01`).

5. Launch Burp using `java -Xmx2g  -jar burpsuite_v1.4.01.jar.`

Linux and Mac OS X

1. Open a terminal.

2. Move to the folder containing the Burp's JAR using the command `cd` (for example, `cd burpsuite_v1.4.01`).

3. Launch Burp using `java -Xmx2g  -jar burpsuite_v1.4.01.jar.`

The option `-Xmx2g` is used to increase up to 2 gigabyte, the maximum memory allocated for Java.

Please note that on some platforms (for example, Windows), Burp Suite can be launched by simply double-clicking the JAR file. However, executing Burp in this way does not allow you to customize the maximum memory available for the tool.

After a few seconds, the main Burp Suite window will appear in the center of the screen. If not, we suggest double checking all commands and carefully reading any error message in the command line. Common errors include wrong permissions or incorrect paths.

Step 4 – Verify Burp Proxy configuration

Burp Proxy acts as an intermediary for requests from the browser to the target web application. As a result, it's not immediately apparent even if it's correctly configured until you connect your browser.
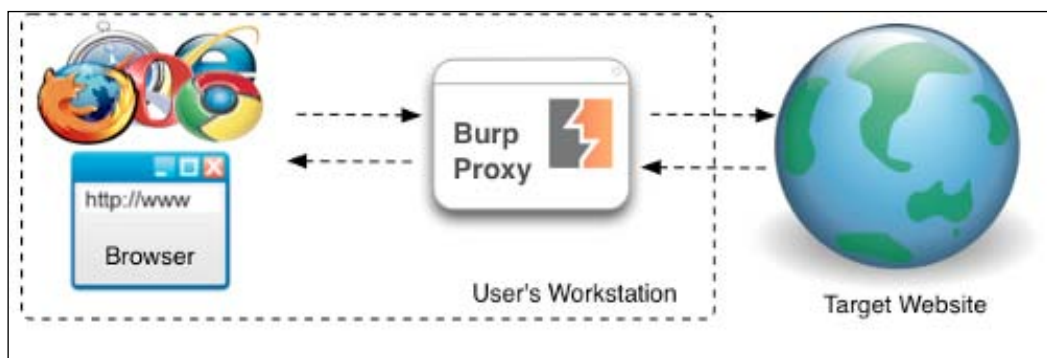
How Burp Proxy interacts with the browser and the target website

By default, Burp Proxy is configured to listen on port **8080/tcp**. To verify that no other software on the computer is interfering with it (for example, using the same TCP port), you can check the proxy listener in the **Proxy | Options**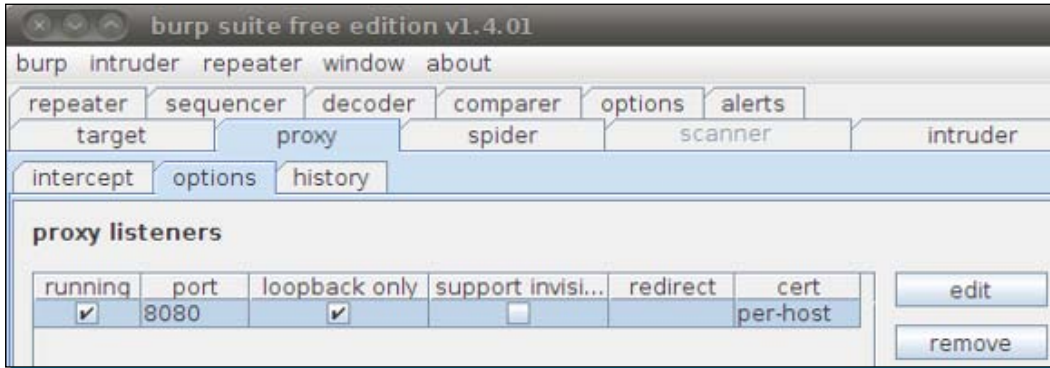 tab. If the **running** checkbox is marked, Burp Proxy is ready to receive requests from the browser. In case of errors, you will notice the presence of exceptions in the **alerts** tab. In some cases, it may be required to change the port and restart the listener, by simply clicking the **running** checkbox.

Burp Proxy configuration

The configuration can be modified by selecting the proxy item and clicking on **edit**. For example, you can change the port by typing a new port number in the **local listener port** textbox and then by clicking on **update**. Finally, click again on the **running** checkbox to start the listener if it is not already selected.

If the **loopback only** checkbox is selected, Burp Proxy will allow connections from the local machine only. Otherwise, it is possible to deselect this option in case other computers need to access Burp remotely.

> Please note that exposing the Burp Proxy listener to other hosts in the network is highly discouraged due to security implications.

If you are testing a standard web application, you can skip to *Step 5 – Configuring the browser*. In most cases, these are the only configurations required for Burp Proxy.

In particular situations, for example while testing standalone clients or mobile applications communicating over HTTP/S, you may need to select the **support invisible proxying for non-proxy-aware clients** checkbox as well as manually enter the target host and port in the appropriate fields. In this way, Burp will take care of all non-proxy style requests allowing you to redirect all traffic to the target host.

Step 5 – Configuring the browser

At this stage, it is just necessary to configure your favorite browser in order to redirect all HTTP/S requests through Burp Proxy, instead of the actual target website. If you haven't changed the default configuration in Burp during the previous step, you would need to set the proxy host address to `127.0.0.1` and the proxy port to `8080`, for both HTTP and HTTPS.

A step-by-step configuration is provided for two of the most common browsers: Mozilla Firefox and Internet Explorer. For other browsers, such as Safari, Chrome, and Opera, please refer to the official browser's documentation. I suggest that you use Mozilla Firefox, because of its versatility. Also, at the time of writing, Mozilla Firefox does not include any embedded **Anti-Cross-Site Scripting** (**XSS**) filter that may interfere with your testing.

Mozilla Firefox

1. Go to the Firefox menu and click on **Preferences**.
2. In the **Advanced** options, under the **Network** tab, click on connection **Settings**.
3. Select **Manual proxy configuration**.
4. Enter the proxy host address (for example, **127.0.0.1**) and the proxy port (for example, **8080**), as configured during *Step 4 – Verify Burp Proxy configuration*.
5. Select **Use this proxy server for all protocols**.
6. Make sure to remove all exceptions from the **No Proxy for** field.
7. Click **OK** and close all windows.
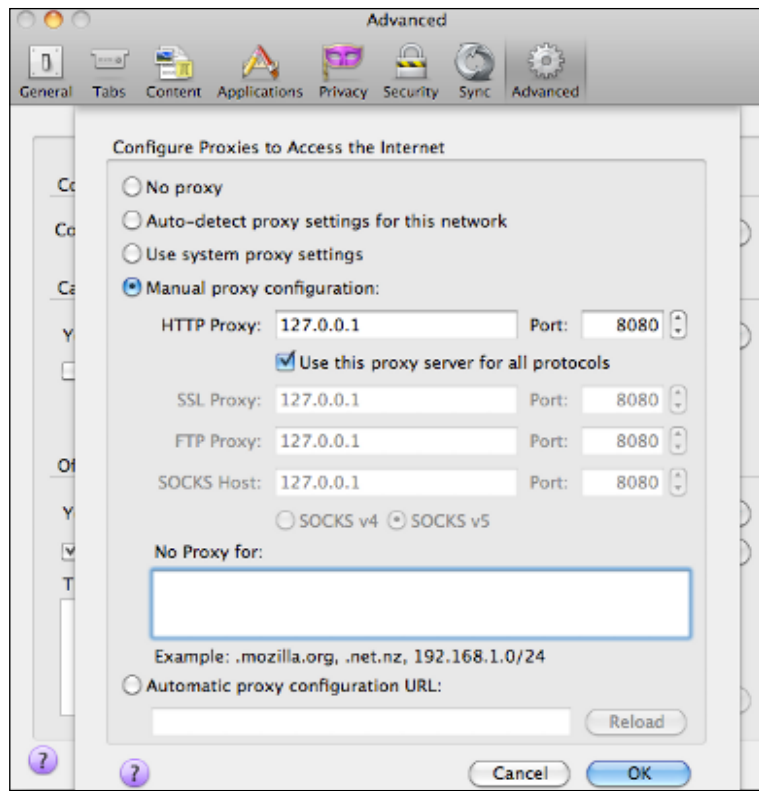
Proxy configuration in Mozilla Firefox

Microsoft Internet Explorer

1. Go to the **Tools** menu and click on **Internet Options**.

2. Under the **Connections** tab, click on **LAN Settings**.

3. Select **Use a proxy for your LAN**.

4. Click on **Advanced**.

5. In the **HTTP** field, enter the proxy host address (for example, `127.0.0.1`) and the proxy port (for example, **8080**).

6. Select **Use the same proxy server for all protocols**.

7. Make sure to remove all exceptions from the **Do not user proxy server for addresses beginning with** field.

8. Click **OK** and close all windows.
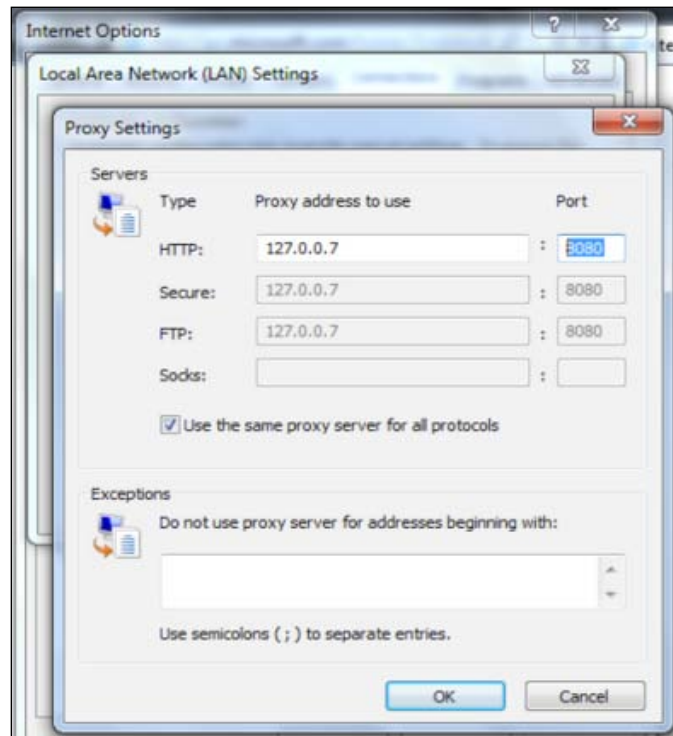
Proxy configuration in Internet Explorer

Independently from the specific browser used, make sure to disable all extensions and additional features that may interfere with Burp Suite. These include third-party proxy add-ons and security enhancements (for example, Anti-XSS filters, NoScript, and so on). If possible, create a dedicated profile for testing purposes only.

And that's it!!

By this point, you should have a working installation of Burp Suite and your browser should be properly configured to intercept all requests.

Go to the browser, enter `http://www.packtpub.com/` in the address bar and press *Enter*. If everything is properly configured, Burp Proxy should intercept your request. In Burp Suite, go to the **Proxy | Intercept** tab and verify that the web request is waiting for your approval.

The **intercept on** button should be highlighted; click on it and allow the request to transit through Burp. Back in the browser, you should see the Packt Publishing page displayed as usual. Also, you can observe that, under the tab **Target | Site Map**, a tree of resources is being populated.

> Make sure to verify the Burp Proxy intercept configuration from time to time. If you are browsing an application and your browser does not load the page, Burp is probably stopping the request or the response.
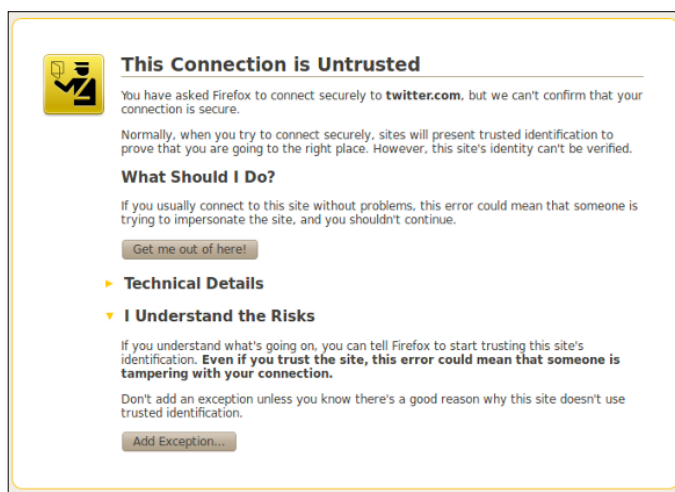
If your browser behaves in a different way or things are not working as described, repeat the previous steps making sure that Burp Proxy is properly listening and the browser is configured in order to connect to Burp.

Assuming that you have successfully set up Burp Suite, it's time to start testing your application.

One more thing...

As you already know, Burp Suite supports HTTP and HTTPS. The latter is a widely used protocol for secure communications between the browser and the server. Nowadays, HTTPS is the standard for protecting online shopping, Internet banking, and other sensitive operations. Using this protocol, HTTP is encapsulated over an SSL/TLS layer. HTTPS protects the transit of data against network sniffing and the so-called **Man-in-the-Middle** (**MitM**) attacks.

In our setup, Burp Suite is exactly configured as a Man-in-the-Middle, since it is supposed to eavesdrop all requests and responses. As a side effect, by visiting HTTPS web pages (for example, `https://www.twitter.com`), you will notice that the browser initially displays a security warning. For example, in Firefox, you will see a **This Connection is Untrusted** page. In these situations, you are required to manually approve the connection by clicking on **I Understand The Risks**, then **Add Exceptions...** and again **Confirm Security Exception**. To make sure that Burp Proxy is actually causing the warning, you may want to click on the certificate status **View...** and verify that the certificate belongs to PortSwigger CA.

This Connection is Untrusted

You have asked Firefox to connect securely to **twitter.com**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

[Get me out of here!]

▶ **Technical Details**

▼ **I Understand the Risks**

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

[Add Exception...]

Invalid certificate warning in Mozilla Firefox

Similarly, using Internet Explorer or other browsers, it is possible to bypass the security warning and continue the navigation. For example, in Google Chrome, you can simply click on **proceed** within the warning page.

> Please note that we are consciously tweaking the browser to allow traffic eavesdropping. As a result, it is discouraged to perform online shopping, check your e-mail, visit your banking portal, or other sensitive activities while using Burp Suite. In fact, it is highly suggested to use Burp on a browser specifically configured for testing purposes only.

Quick start – Using Burp Proxy

Burp Proxy is a crucial component of the entire Burp Suite. This tool allows you to intercept the web traffic between the browser (client) and the target application (server). Thanks to the setup described in the previous section, we are now able to look under the hood and discover how web applications work.

At the top of Burp Proxy, you will notice the following three tabs:

- ✦ **intercept**: HTTP requests and responses that are in transit can be inspected and modified from this window
- ✦ **options**: Proxy configurations and advanced preferences can be tuned from this window
- ✦ **history**: All intercepted traffic can be quickly analyzed from this window

> If you are not familiar with the HTTP protocol or you want to refresh your knowledge, *HTTP Made Really Easy, A Practical Guide to Writing Clients and Servers*, found at `http://www.jmarshall.com/easy/http/`, represents a compact reference.

Step 1 – Intercepting web requests

After firing up Burp and configuring the browser, let's intercept our first HTTP request. During this exercise, we will intercept a simple request to the publisher's website:

1. In the **intercept** tab, make sure that Burp Proxy is properly stopping all requests in transit by checking the **intercept** button. This should be marked as **intercept is on**.

2. In the browser, type `http://www.packtpub.com/` in the URL bar and press *Enter*.

   Back in Burp Proxy, you should be able to see the HTTP request made by the browser. At this stage, the request is temporarily stopped in Burp Proxy waiting for the user to either forward or stop it.

   For instance, press forward and return to the browser. You should see the home page of Packt Publishing as you would normally interact with the website.

3. Again, type `http://www.packtpub.com/` in the URL bar and press *Enter*.

4. Let's press **drop** this time.

   Back in the browser, the page will contain the warning **Burp proxy error: message was dropped by user**. We have dropped the request, thus Burp Proxy did not forward the request to the server. As a result, the browser received a temporary HTML page with the warning message generated by Burp, instead of the original HTML content.

5. Let's try one more time. Type `http://www.packtpub.com/` in the URL bar of the browser and press *Enter*.

    Once the request is properly captured by Burp Proxy, the **action** button becomes active. Click on it to display the contextual menu. This is an important functionality as it allows you to import the current web request in any of the other Burp tools.

You can already imagine the potentialities of having a set of integrated tools that allow you to manipulate and analyze web requests so easily. For example, if we want to decode the request, we can simply click on **send to decoder**.
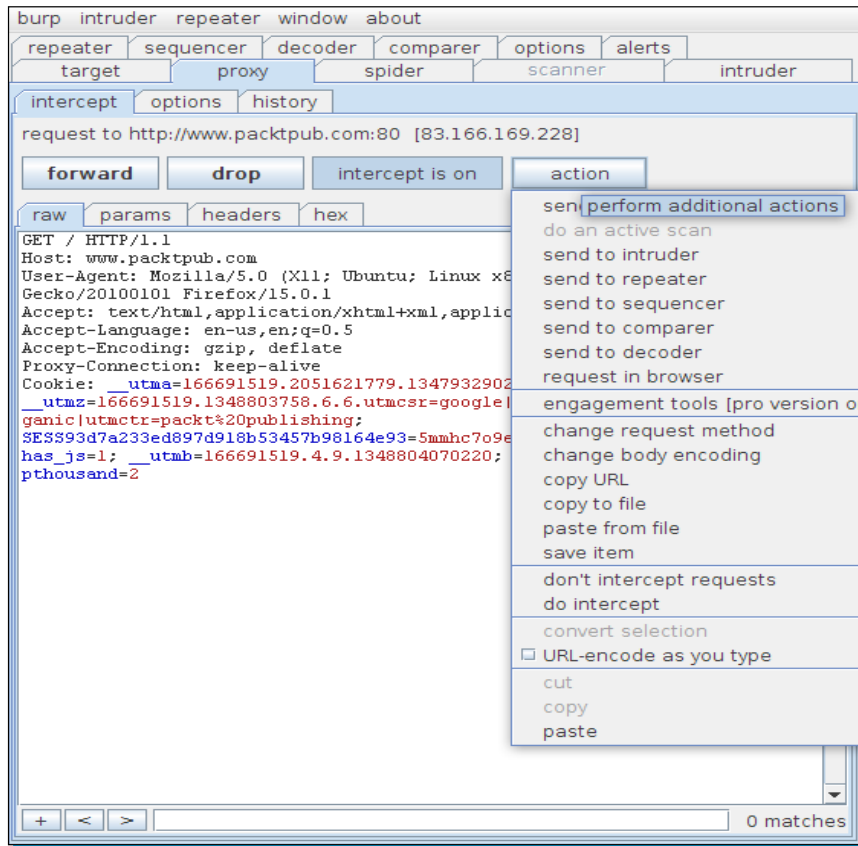
Burp Proxy

In Burp Proxy, we can also decide to automatically forward all requests without waiting for the user to either forward or drop the communication. By clicking on the **intercept** button, it is possible to switch from **intercept is on** to **intercept is off**. Nevertheless, the proxy will record all requests in transit.

Also, Burp Proxy allows you to automatically intercept all responses matching specific characteristics. Take a look at the numerous options available in the **intercept server response** section from within the Burp Proxy **options** tab. For example, it is possible to intercept the server's response only if the client's request was intercepted. This is extremely helpful while testing input validation vulnerabilities as we are generally interested in evaluating the server's responses for all tampered requests. Or else, you may only want to intercept and inspect responses having a specific return code (for example, 200 OK).

Step 2 – Inspecting web requests

Once a request is properly intercepted, it is possible to inspect the entire content, headers, and parameters, using one of the four Burp Proxy message analysis tabs:

- ✦ **raw**: This view allows you to display the web request in raw format within a simple text editor. This is a very handy visualization as it enables maximum flexibility for further changing the content.

- ✦ **params**: In this view, the focus is on user-supplied parameters (GET/POST parameters, cookies). This is particularly important in case of complex requests as it allows to consider all entry points for potential vulnerabilities. Whenever applicable, Burp Proxy will also automatically perform URL decoding. In addition, Burp Proxy will attempt to parse commonly used formats, including JSON.

- ✦ **headers**: Similarly, this view displays the HTTP header names and values in tabular form.

- ✦ **hex**: In case of binary content, it is useful to inspect the hexadecimal representation of the resource. This view allows to display a request as in a traditional hex editor.

The **history** tab enables you to analyze all web requests transited through the proxy:

1. Click on the **history** tab. At the top, Burp Proxy shows all the requests in the bundle. At the bottom, it displays the content of the request and response corresponding to the specific selection. If you have previously modified the request, Burp Proxy **history** will also display the modified version.

Displaying HTTP requests and responses intercepted by Burp Proxy

2. By double-clicking on one of the requests, Burp will automatically open a new window with the specific content. From this window, it is possible to browse all the captured communication using the **previous** and **next** buttons.

3. Back in the **history** tab, Burp Proxy displays several details for each item including the request method, URL, response's code, and length. Each request is uniquely identified by a number, visible in the left-hand side column.

4. Click on the request identifier. Burp Proxy allows you to set a color for that specific item. This is extremely helpful to highlight important requests or responses. For example, during the initial application enumeration, you may notice an interesting request; you can mark it and get back later for further testing. Burp Proxy **history** is also useful when you have to evaluate a sequence of requests in order to reproduce a specific application behavior.

5. Click on the display filter, at the top of the history list to hide irrelevant content. If you want to analyze all HTTP requests containing at least one parameter, select the **show only parameterised** checkbox. If you want to display requests having a specific response, just select the appropriate response code in the **filter by status code** selection. At this point, you may have already understood the potentialities of the tool to filter and reveal interesting traffic.

> In addition, when using Burp Suite Professional, you can also use the **filter by search term** option. This feature is particularly important when you need to analyze hundreds of requests or responses as you can filter relevant traffic only by using regular expressions or simply matching particular strings. Using this feature, you may also be able to discover sensitive information (for example, credentials) embedded in the intercepted pages.

Step 3 – Tampering web requests

As part of a typical security assessment, you will need to modify HTTP requests and analyze the web application responses. For example, to identify SQL injection vulnerabilities, it is important to inject common attack vectors (for example, a single quote) in all user-supplied input, including HTTP headers, cookies, and GET/POST parameters.

> If you want to refresh your knowledge on common web application vulnerabilities, the *OWASP Top Ten Project* article at `https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project` is a good starting point.

Tampering web requests with Burp is as easy as editing strings in a text editor:

1. Intercept a request containing at least one HTTP parameter. For example, you can point your browser to `http://www.packtpub.com/books/all?keys=ASP`.

2. Go to **Burp Proxy | Intercept**. At this point, you should see the corresponding HTTP request.

3. From the **raw** view, you can simply edit any aspect of the web request in transit. For example, you can change the value of the the the `GET` parameter's `keys` value from `ASP` to `PHP`. Edit the request to look like the following:

```
GET /books/all?keys=PHP HTTP/1.1
Host: www.packtpub.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:15.0)
Gecko/20100101 Firefox/15.0.1
Accept: text/html,application/xhtml+xml,application/
xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Proxy-Connection: keep-alive
```

4. Click on **forward** and get back to the browser. This should result in a search query performed with the string `PHP`. You can verify it by simply checking the results in the HTML page.

Although we have used the **raw** view to change the previous HTTP request, it is actually possible to use any of the Burp Proxy view. For example, in the **params** view, it is possible to add a new parameter by following these steps:

1. Clicking on **new** (right side), from the Burp Proxy **params** view.

2. Selecting the proper parameter type (**URL, body,** or **cookie**). **URL** should be used for GET parameters, whereas **body** denotes POST parameters.

3. Typing the name and the value of the newly created parameter.

Advanced features

After practicing with the basic features provided by Burp Proxy, you are almost ready to experiment with more advanced configurations.

Match and replace

Let's imagine that you are testing an application designed for mobile devices using a standard browser from your computer. In most cases, the web server examines the user-agent provided by the browser to identify the specific platform and respond with customized resources that better fit mobile phones and tablets. Under these circumstances, you will particularly find the **match and replace** function, provided by Burp Proxy, very useful. Let's configure Burp Proxy in order to tamper the user-agent HTTP header field:

1. In the **options** tab of Burp Proxy, scroll down to the **match and replace** section.

2. Under the **match and replace** table, a drop-down list and two text fields allow to create a customized rule. Select **request header** from the drop-down list since we want to create a match condition pertaining to HTTP requests.

3. Type `^User-Agent.*$` in the first text field. This field represents the match within the HTTP request. Burp Proxy's **match and replace** feature allows you to use simple strings as well as complex regular expressions.

> 📝 If you are not familiar with regular expressions, have a look at `http://www.regular-expressions.info/quickstart.html`.

4. In the second text field, type `Mozilla/5.0 (iPhone; U; CPU like Mac OS X; en) AppleWebKit/4h20+ (KHTML, like Gecko) Version/3.0 Mobile/1C25 Safari/419.3` or any other fake user-agent that you want to impersonate.

5. Click **add** and verify that the new match has been added to the list; this button is shown here:
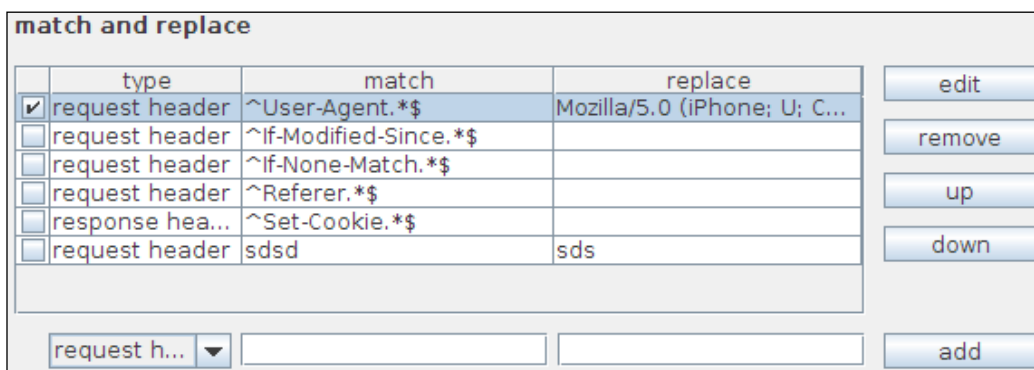
Burp Proxy match and replace list

6. Intercept a request, leave it to pass through the proxy, and verify that it has been automatically modified by the tool.
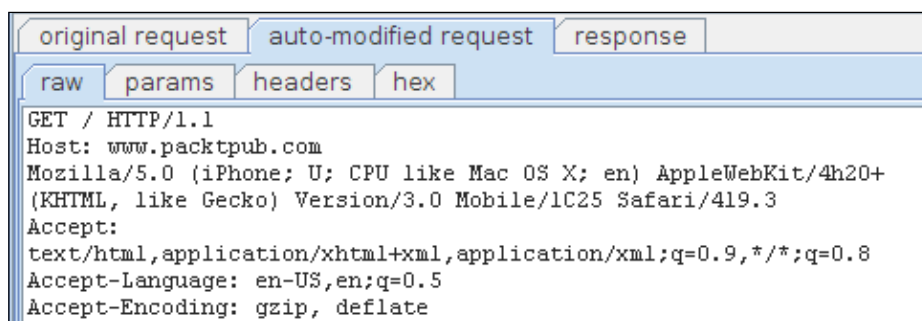
Automatically modified HTTP header in Burp Proxy

HTML modification

Another interesting feature of Burp Proxy is the automatic HTML modification, that can be activated and configured in the appropriate section within **Burp Proxy | options**. By using this function, you can automatically remove JavaScript or modify HTML forms of all received HTTP responses.

Some applications deploy client-side validation in the form of disabled HTML form fields or JavaScript code. If you want to verify the presence of server-side controls that enforce specific data formats, you would need to tamper the request with invalid data. In these situations, you can either manually tamper the request in the proxy or enable HTML modification to remove any client-side validation and use the browser in order to submit invalid data. This function can be also used to display hidden form fields.
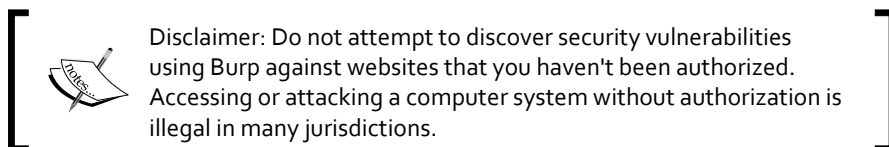
Let's see in practice how you can activate this feature:

1. In Burp Proxy, go to **options**, scroll down to the **HTML modification** section.

2. Numerous options are available in this section: **unhide hidden form fields** to display hidden HTML form fields, **enable disabled form fields** to submit all input forms present inside the HTML page, **remove input field length limits** to allow extra-long strings in the text fields, **remove JavaScript form validation** to make Burp Proxy all `onsubmit` handler JavaScript functions from HTML forms, **remove all JavaScript** to completely remove all JS scripts and **remove object tags** to remove embedded objects within the HTML document.

3. Select the desired checkboxes to activate automatic HTML modification.

Using this feature, you will be able to understand whether the web application enforces server-side validation. For instance, some insecure applications use client-side validation only (for example, via JavaScript functions). You can activate the automatic HTML modification feature by selecting the **remove JavaScript form validation** checkbox in order to perform input validation testing directly from your browser.

Top 8 features you need to know about

As you start to use Burp Suite, you will soon realize that there is a wide variety of things that you can do with it. This section will teach you all about the most commonly performed tasks and most useful features in the Burp tools

> Disclaimer: Do not attempt to discover security vulnerabilities using Burp against websites that you haven't been authorized. Accessing or attacking a computer system without authorization is illegal in many jurisdictions.

1 – Using the target site map functionality

One of the first activities during a web application security assessment consists of exploring the application in order to enumerate resources and endpoints. Using Burp Suite, you can simply browse the application and exercise all functionalities as you would normally do with your browser. Burp Suite keeps track of all HTTP requests and responses and displays all data using the target site map functionality.
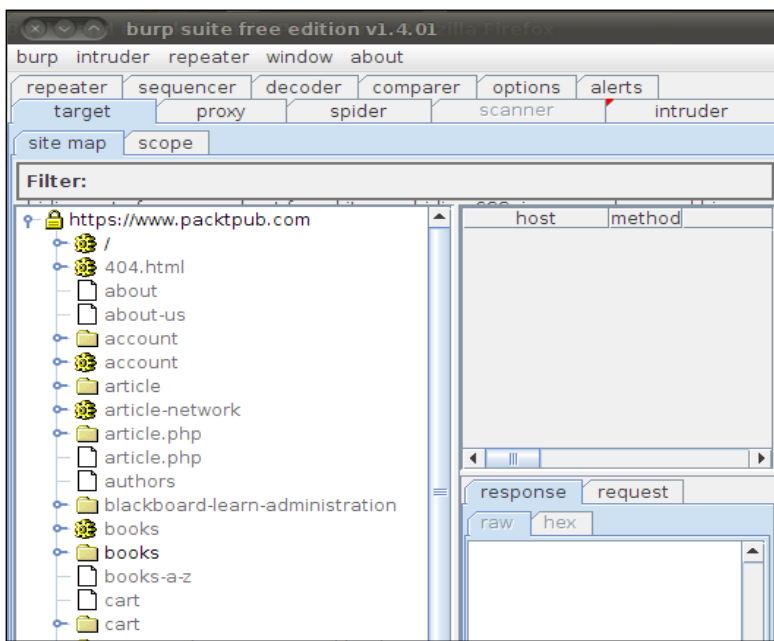
Burp target site map

The Burp **Target** tab shows all endpoints and parameters in a convenient hierarchical representation. This view is normally referred to as a site map. The process of mapping all application resources is crucial and Burp site map allows you to quickly analyze the application's attack surface.

From the site map tree it is possible to select the target of our assessment and to reduce the scope for all built-in tools. This is an important feature as it allows Burp's user to focus on relevant resources and prevent any interaction with third-party websites Let us see how this can be done:

1. In the **site map** tab of Burp Target, select your application by clicking on the root node containing the domain name (for example, `http://www.packtpub.com/`).

2. Right-click and click **add item to scope** to reduce the in-scope target for the entire Burp Suite. By default, Burp in-scope items list is empty and all domains are considered part of the audit

3. Moreover, it is possible to filter out resources related to other domains by clicking on the Burp **Target | site map | Filter** area. Then select show **only in-scope items** from the **filter by request type** section

4. At this point, your site map should only display resources belonging to the selected domain. It is also possible to verify this setting by checking the **include in scope** table within Burp **Target | scope** tab.
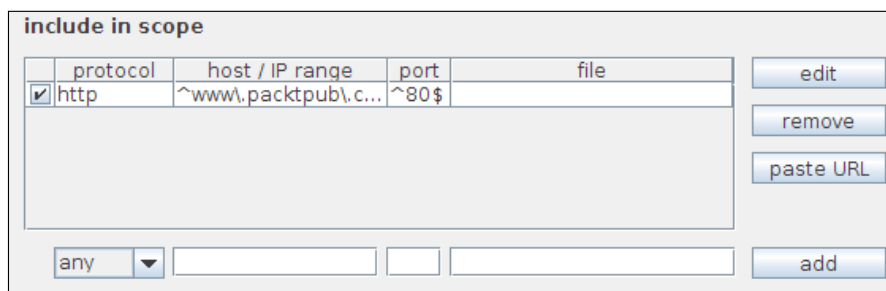
Burp Target in-scope items

From this form, it is also possible to manually add, edit, or remove in-scope items. For instance, if you want to include a new domain, you can proceed in the following way:

1. Under the **include in scope** table, choose the correspondent protocol (**any, http,** or **https**)

2. In the first text field from the left, specify a regular expression that identifies the domain or the sub-domain to add to the scope (for example, **^www\.authors. packtpub\.com$**), as illustrated in the previous image. The regular expression syntax in Burp is most similar to that found in Perl.

3. In the central text field, specify a regular expression that identifies the appropriate port number (for example, usually `^80$` for **http** and `^443$` for **https**).

4. Optionally, it is possible to specify a regular expression for files and folders in the third text field. If you intend to analyze and test the entire application domain, leave this field blank.

5. Finally, click on **add** and verify that the checkbox in the **include in scope** table has been automatically selected.

> Important! Please note that reducing the scope of the tool to the application under analysis is crucial. This setting will prevent any involuntary request and attack to third-party applications.

In the same way, by using the **exclude from scope** table, it is possible to define resources that should not be examined by the tool. This feature allows to define a black list for off-limit endpoints, which is particularly useful to make Burp avoiding logout functions, reset buttons, or other destructive operations.

In the site map, from any domain and any item (endpoints or parameters), it is possible to invoke a contextual menu by right-clicking on the specific item.
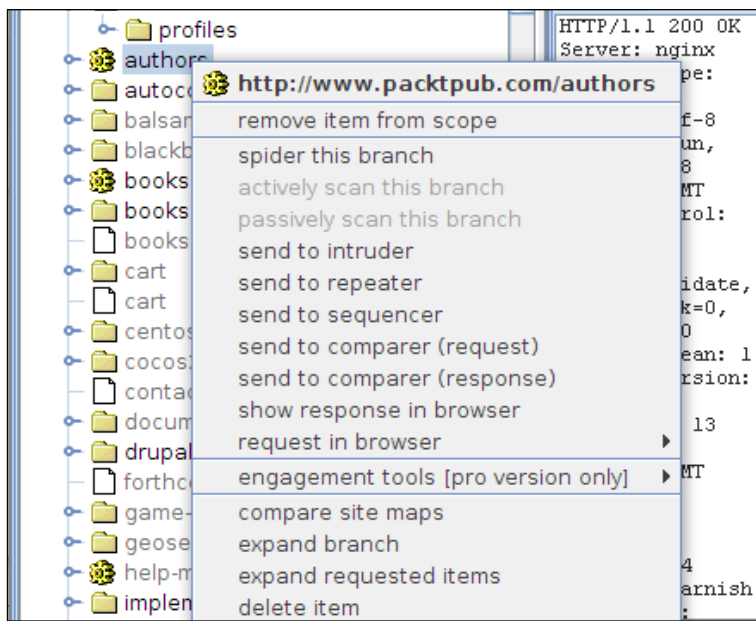
Burp Target contextual menu

This mechanism allows you to quickly import requests/responses in all Burp Suite tools by selecting one the following items:

✦ **Spider this branch** to activate Burp Spider

✦ **Actively/Passively scan this branch** to start an automatic scan with Burp Scanner (available in the professional version only)

✦ **Send to intruder** to launch customized attacks

✦ **Send to repeater** to modify and re-iterate the same request over and over

✦ **Send to sequencer** to analyze application data predictability

✦ **Send to comparer (request/response)** to visually compare multiple requests or response

These functionalities will be described in the following sections of this chapter.

In addition, the contextual menu allows to reproduce HTTP requests and responses in the browser. This is particularly useful to verify the behavior of a specific browser during the analysis of client-side attacks (for example, Cross-Site Scripting, UI redressing, and so on).

1. Select a request from the site map tree.

2. Right-click and select **request in browser**.

3. Choose to either use the **current browser session** or the **original session** option, which makes Burp—using the session token —available in the saved request (if applicable).

4. A pop-up window will display a virtual URL (for example. `http://burp/repeat/0`). Click on **copy**.

5. In the browser, paste the URL by pressing *Ctrl + V* or using the corresponding command from the toolbar menu.

6. Finally, press *Enter* to emulate the request within the browser

During the course of your security testing, consult the site map to verify that you have analyzed all application entry points. Burp tools such as Burp Spider will help you to automatically populate the site map. Resources that have been already requested by the tool are marked in black, whereas endpoints that are linked by other resources, but haven't been retrieved by Burp, are marked in gray.

2 – Crawling a web application with Burp Spider

Burp Spider allows to automatically crawl web applications and retrieve visible and hidden resources. The tool uses a combination of techniques to maximize the result, including following links discovered in previously saved HTTP responses and automatically submitting web forms.

The first step requires setting up the spider by using the **options** tab in Burp Spider. Although in most cases, the default options are sufficient to achieve good results, you may want to customize some of the spider's preferences

✦ For large websites, it may be necessary to modify **maximum link depth**, which represents the maximum number of redirections to follow for a resource.

✦ In case of fragile hosts with limited system resources, you may need to reduce the number of threads by changing the number in the **thread count** textbox within the **spider engine** section. Also, you can increase the number of retries in case of network failure and the pause time before each trial.

✦ If you want to have Burp Spider automatically submit credentials, you can define username and password in the **application login** section.

Another interesting feature of Burp Spider is the possibility to define **name** and **value** fields used by the tool to automatically submit HTML forms. During crawling, the spider may encounter web forms that have to be filed with semantically valid content. For example, let's imagine a registration form with an **e-mail** field; in this case, the spider must be able to recognize the specific field and submit a valid e-mail. Burp Spider allows you to define custom regular expressions to match field names:

1. In Burp Spider, go to **options | forms**.

2. Assuming that you want Burp Spider to submit forms, select the **automatically submit using the following rules to assign parameter values** option.

3. The table, shown in the following screenshot, includes all name-value pairs used by the tool to populate fields. Like in other Burp tools, it is possible to add, remove, or modify entries. For example, if you want to create a rule for submitting a user ID specified by the keyword `PacktUserID`, you can first select **regex** from the drop-down list.

4. Then, type `PacktUserID` in the first text box from the left. This is the field name for the custom rule and it is interpreted by the tool as the regular expression `^PacktUserID$`.

5. Insert the correspondent **field value**. This is the actual value that we want to assign for that specific field.

6. Finally, click on **add** and make sure that the checkboxes are properly selected. All these steps can be better understood by observing the following screenshot:

Automatic forms submit configuration in Burp Spider

At this point, the tool is ready to be activated. Burp Spider can be either activated by the contextual menu in Burp Target or by marking the **spider running** checkbox in Burp Spider's **control** tab.

It is suggested to select one node from the site map tree, right-click and choose spider from here. Burp Spider will start crawling from the selected resource under the specific branch in that domain. By default, Burp Spider uses the scope defined in the **Target** tab, this behavior ensures that the tool will not invoke resources on domains outside the target.

From the **control** tab in Burp Spider, it is also possible to verify the progress of the tool thanks to the displayed information. Details include the total number of HTTP requests sent by the spider and the remaining number of resources yet to be invoked. All results from the discovery are automatically added to **Target | site map**.
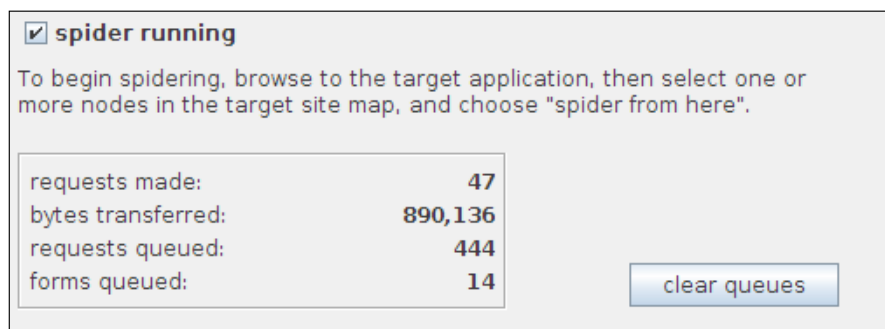
Burp Spider in action

In addition to automatic crawling with Burp Spider, it is very important to properly map all application resources by manually browsing the website. Enumerating all resources is crucial before scanning the application or manually testing endpoints.

3 – Launching an automatic scan with Burp Scanner

Burp Scanner is a dynamic web application scanner, included in the Professional edition of the Burp Suite. The tool allows you to automatically scan websites and detect common security flaws, including but not limited to SQL Injection, Cross-Site Scripting, XML Injection, missing cookie flags (for example, `HttpOnly` and `Secure`), and so on.

The tool allows two scanning modes:

✦ **Active Scanning**: In this mode, the detection of vulnerabilities is performed by sending HTTP requests containing common attack patterns and analyzing responses with pattern-matching heuristics

✦ **Passive Scanning**: Using this mode, Burp Scanner uses stored requests and responses to identify flaws that can be analyzed offline and do not require active probing

For testing this functionality, I suggest you to use **Google Gruyere** (`http://google-gruyere.appspot.com/`), a deliberately-insecure web application. Gruyere can be accessed online:

1. Visit `http://google-gruyere.appspot.com/part1` and carefully read the instructions and the disclaimer.

2. Go to `http://google-gruyere.appspot.com/start`.

3. Click on **Agree&Start**.

4. Click on **Sign Up** and create a testing account.

5. Click on **Sign in** and log in using the previously created account. At this point, configure Burp to intercept all requests. Your browser should look like:


Google Gruyere home page screen

Burp Scanner can automatically scan resources on-the-fly while browsing the website or it can be activated from the contextual menu in Burp **Target site map**.

By default, Burp Scanner is configured to perform passive scanning on all domains, while active scanning is disabled. In Burp's **Scanner** tab, select **Live scanning**, then select **Use suite scope** in both **Active Scanning** and **Passive Scanning** sections to automatically scan all resources of the application under analysis that are passing through Burp Proxy. This modality is often referred to as on-the-fly scanning.

Alternatively, you can select a specific branch of the target in the **site map** tab and click on **actively scan this host** or **passively scan this host**, depending on the desired mode.
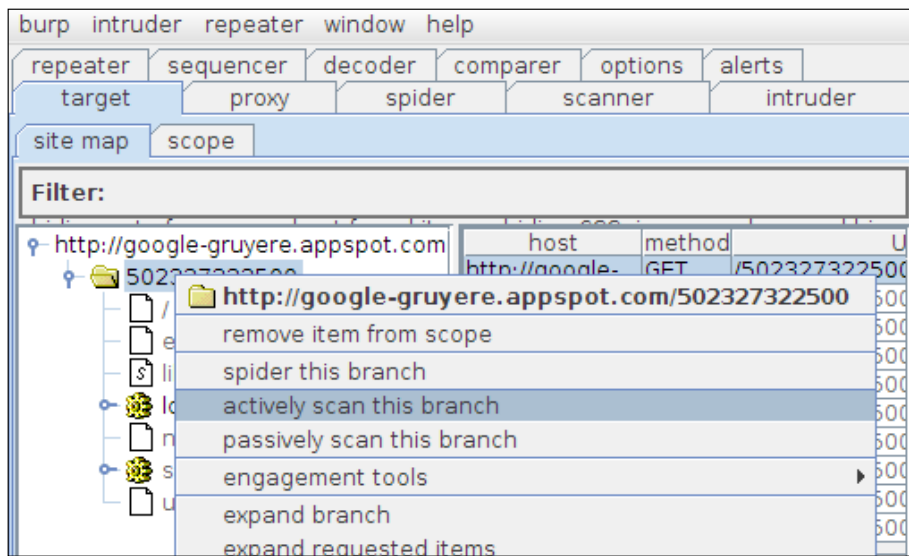
Launching Burp Scanner from the Burp Target site map

If you choose to start an active scan, Burp Suite will display a new window named **Active scanning wizard**, an easy configuration tool for Burp Scanner:

1. The first step in the configuration process allows you to remove specific types of resources, including images, JavaScript, or stylesheets. In most cases, the default setup is adequate and it is just necessary to click on **next**.

2. In the second step, the tool will display a table containing the entire list of endpoints and parameters that Burp Scanner is going to include during the scanning. It is very important to carefully review this list and remove endpoints that are either not relevant or may cause malfunctions (for example, delete users, reset application functionalities, and so on). At the bottom of the endpoints' table, the tool also displays the total number of items. Once you have finalized your selection, click on **Ok** to start scanning.
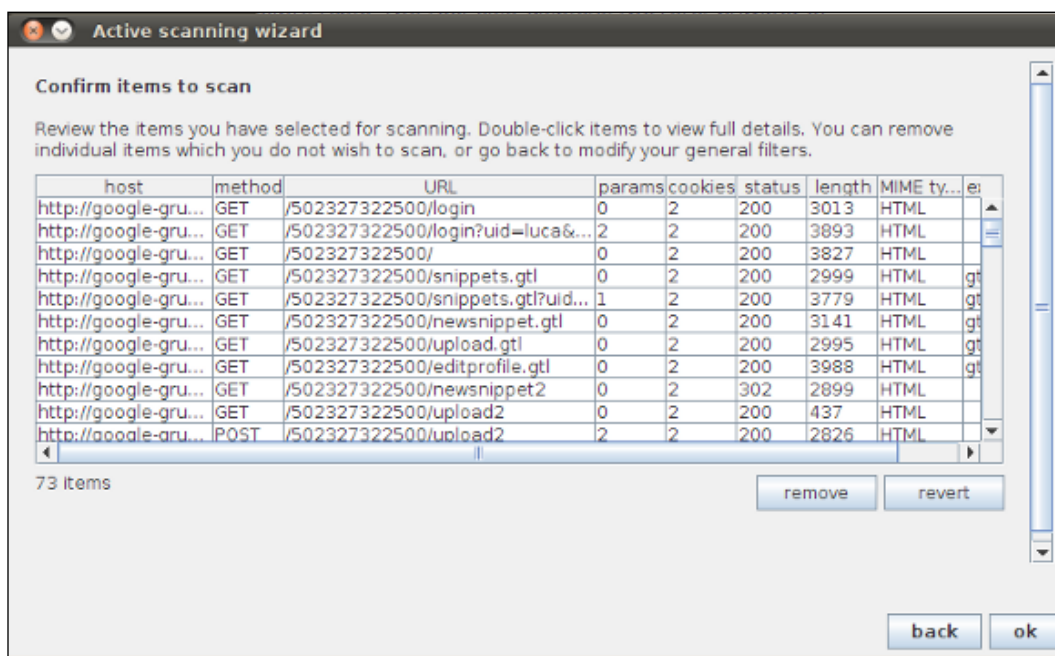
Burp Scanner Active scanning wizard

The default configuration of Burp Scanner fits most of the use cases. However, if you want to further tune the scanner, you can customize all configurations by going to the **options** tab in Burp Scanner. This tab contains numerous options, including the possibility to enable/disable insertion points:

✦ **URL parameter values**: To perform tampering in all HTTP GET parameters

✦ **Body parameter values**: To perform tampering in all HTTP POST parameters

✦ **Cookie parameter values**: To consider all session tokens as possible entry points

✦ **Parameter name**: To consider the name of HTTP GET/POST parameters as possible entry point

- ✦ **HTTP headers**: To perform tampering in all request headers, including standard and custom headers

- ✦ **AMF string parameters**: In case of applications developed using Adobe Flex, Burp Scanner will parse the Action Message Format binary protocol and perform tampering in all string parameters

- ✦ **REST-style URL parameters**: In case of applications implementing a REST interface, Burp Scanner will tamper with part of the URL, which is normally used to identify operations and arguments
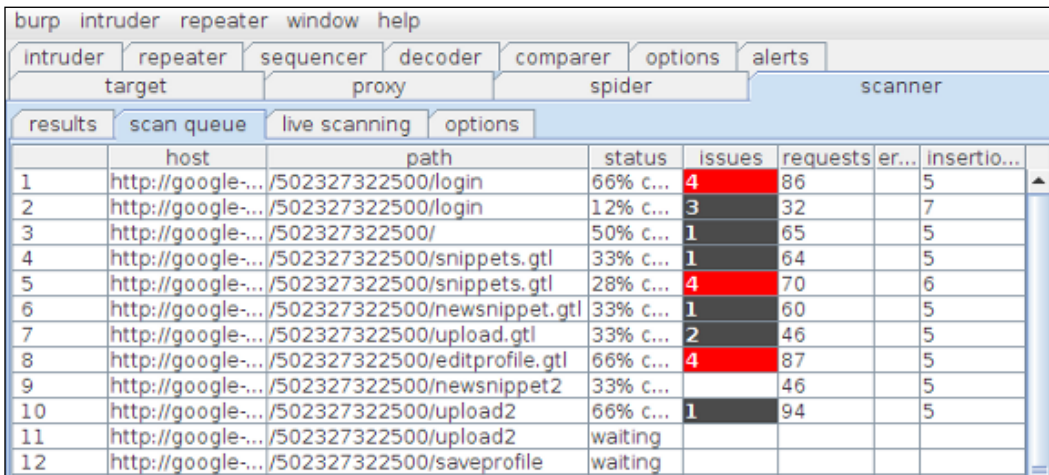
If you are looking for specific categories of vulnerabilities, you can enable/disable each single check performed by the tool by selecting/deselecting the corresponding checkbox in the **Active Scanning Area**s section and in the **Passive Scanning Areas** section. For instance, if you are testing an application with no access to LDAP subsystems, you can optimize your scan by deselecting the **LDAP injection** checkbox.

Also, the **Options** tab in Burp Scanner allows to limit the number of threads used by the tool or increase the time between consecutive requests. Depending on the system resources available on the server, you may decide to speed up or slow down your scan by tuning all options in the **Active Scanning Engine** section.

> Important! Do not attempt to find security vulnerabilities using Burp Scanner or other Burp tools in websites that you haven't been authorized. Although all Burp Scanner checks are designed to avoid malfunctions, they may cause severe failures and even irreversible damage. Even if you are authorized, consider performing a backup of your system and your data before testing. Be always careful and constantly monitor the scan's progress and the server's status.

Once you have launched your scan, you can monitor the progress by checking the **scan queue** tab in Burp Scanner. This table provides information on the scan requests completed and in progress. Also, it provides an overview of the results by displaying the number of issues discovered for each endpoint. From this table, you can also remove items by selecting those resources, right-clicking on the table, and clicking on **Delete items**. Moreover, you can pause and restart the entire scanner by right-clicking on the table and selecting **Pause Scanner** or **Resume Scanner**.

Burp Scanner scan queue

Scanning an entire web application may require several minutes, sometimes even hours. Nevertheless, you can analyze the results at any time by checking the findings tree in the **results** tab of Burp Scanner. Like in the site map, this visualization groups vulnerabilities per endpoints and categories with a convenient tree representation.
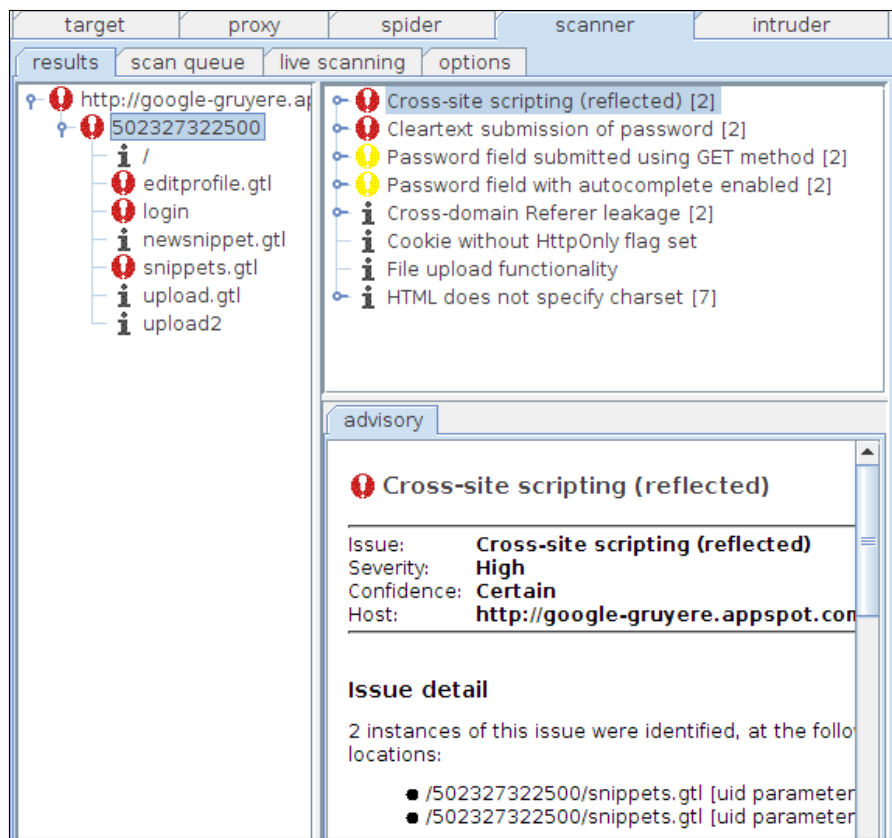
Burp Scanner results

If you click on a specific item, the advisory for the selected security vulnerability is shown. A precise description of the bug is displayed, including the following details:

✦ **Issue**: The vulnerability category (for example, Cross-Site Scripting).

✦ **Severity**: An estimate of the impact on the affected system. Departure from best-practices are normally categorized as **Information** or **Low**, whereas vulnerabilities that may facilitate system compromise are marked as **High**.

✦ **Confidence**: An estimation of the tool's confidence (**Certain**, **Firm**, and **Tentative**). For some classes of vulnerabilities, manual intervention is required to validate results and confirm the presence of a security flaw. In other cases, the tool is able to detect and confirm the vulnerability with no margins of error.

✦ **Host**: The system affected by the security vulnerability.

✦ **Path**: The specific endpoint affected by the security vulnerability.

A contextual menu from the results window allows to remove issues (**Delete selected issues**), or assign a different level of severity (**Set severity**) and confidence (**Set confidence**).

Once all resources have been analyzed and the scan is complete, you can export the results. Burp Scanner allows you to create basic HTML or XML reports that can be used to keep track of the discovered bugs. In addition, other security tools (for example, Rapid7 Metasploit) allow you to import those results to perform further tasks. The following steps will explain how you can export the results:

1. In the **results** tab of Burp Scanner, select all the items that you want to export. In the findings tree, you can also select the root node to export all findings.

2. Right-click to display the contextual menu. Select **Report selected issues**.

3. A new window, titled **Burp Scanner reporting wizard**, will guide you through the format of the report. The first step consists of selecting the report type; namely, **screen-friendly HTML**, **printer-friendly HTML**, or an **XML** report.

4. Next, you can personalize the level of details to be included in the report. For instance, you can decide to have the maximum level of details and verbosity by selecting all the checkboxes. Then, click on **Next**.

5. As it is sometimes useful to provide snapshots of the affected HTTP requests and responses, you can also decide to include relevant extracts in the final report. Select the appropriate checkbox and click on **Next**.

6. In the next step, **Burp Scanner report wizard** allows you to select/deselect categories of issues to export. Make your decision and select the appropriate checkboxes. Then, click on **Next**.

7. Finally, in the last step, you are required to specify the filename of the report. Click on **Select file...** and browse your filesystem to find a folder where you want to save the report. Type the filename, including the file extension. For instance, if you are exporting the results in a HTML report, type `BurpScannerReport.html`. Click on **Save**.

8. Furthermore, you can personalize the layout of the document by changing the order of the content, selecting the **Issue organization** and the **Table of contents levels** scroll-down lists. Also, you can specify the report title and subtitle by filling the **Report title** and **Report subtitle** text fields. Click on **Next**.

9. At the end of the wizard, a progress bar will provide you a feedback on the report generation. Once done, click on **Ok** to close this window.

A report should be available in the folder previously selected. If you saved it as HTML, you can use your browser to open it and analyze your findings.

4 – Automating customized attacks with Burp Intruder

Although Burp Scanner is a useful tool to automatically detect vulnerabilities, it does not allow you to customize attack vectors for each specific request. As soon as you understand the basics of web application security, you will feel the need of having full control over your testing. Web application scanners are not the silver bullet for web security, thus it is also suggested to perform manual testing.

At high level, a web application security assessment consists of testing all entry points (GET/POST parameters, cookies, headers, and so on) with common attack patterns and evaluate the server's responses to identify security flaws. For instance, if you suspect that an endpoint is vulnerable to SQL injection, you may want to iterate the same request over and over again by supplying different attack vectors (for example, a single quote, a single quote and a parenthesis, and so on) for each parameter. This is a very time-consuming task that requires constant supervision. Luckily, Burp Intruder can significantly speed up the process by setting up an attack in few seconds, sending all the requests, and collecting all the responses.

The first step in using Burp Intruder consists of importing a web request in the tool. From the entire Burp Suite, you can send requests to Burp Intruder by using the standard contextual menu. For instance, if you are browsing requests in the Burp Proxy **history** tab, right-click on a specific item and select **send to intruder**. Burp's **intruder** tab should immediately blink red. At this stage, a four-step configuration is required before launching the attack.
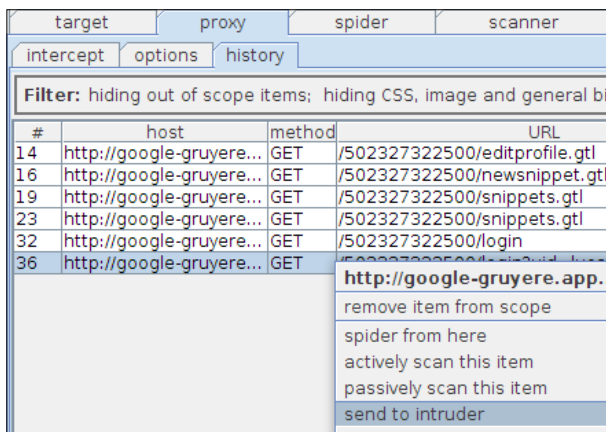
Import a web request in Burp Intruder

Configuring the target

Go to the **target** tab in Burp Intruder. From this tab, it is possible to specify the target host and port. In most cases, it is not required to change anything, thus you can just double-check the pre-filled content and move to the next tab.

Configuring the attack type and positions

In the **positions** tab of Burp Intruder, you need to select the payload positions thereby defining a request template for the attack. By default, Burp Intruder will automatically mark all GET/POST parameters and cookies' values. However, you are encouraged to personalize the attack by adding or removing positions. For example, let's see how to select the first parameter in a HTTP GET request:

1. Import a web request (for example, from Burp Proxy) containing at least one GET parameter, by right-clicking the item and selecting **send to intruder**. For instance, when using Google Gruyere, you can use the login request.

2. In the **positions** tab of Burp Intruder, click on **clear §** to remove all markers.

3. Position the mouse pointer before the first character of the first parameter value in the URL (for example, in `GET /<Your Gruyere Instance ID>/ login?uid=luca&pw=aa`, the mouse pointer should be placed between the character `=` and the letter `l`).

4. Add the first marker by clicking on **add §**.

5. Move the mouse pointer after the last character of the same parameter value and finalize the selection by clicking again on **add §**.

The resulting request template should look like the one shown in the following screenshot:
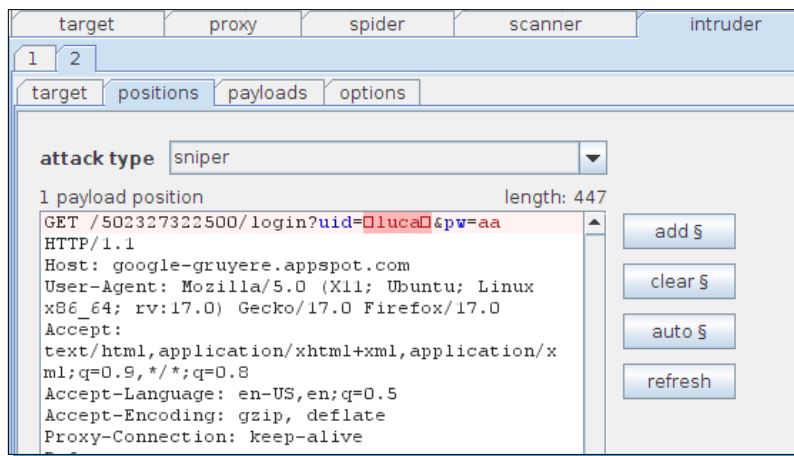
Selection of the first position in Burp Scanner

At this stage, we have successfully marked our first entry point. You can proceed further and create more positions. By clicking on **auto §**, it is possible to revert to the initial setup where all GET/POST parameters and cookies' values are selected.

In the **positions** tab of Burp Intruder, it is also necessary to define a specific attack type using the drop-down list. This setting defines the heuristic used by the tool to replace the previously-marked positions with attack payloads.

The **attack type** drop-down menu consists of the following four modalities:

✦ **sniper**: By using this type, Burp will replace all positions with strings from a single payload list. In particular, it will iterate through all payloads, one by one, for all positions. This allows you to permute all combinations of attack payloads and original values in the template request.

Request	Position	Payload
#1	1	Item_1_List_1
#2	1	Item_2_List_1
#3	2	Item_1_List_1
#4	2	Item_2_List_1

✦ **battering ram:** Similar to the sniper attack, this heuristic uses a single payload list. In this case, all positions are simultaneously replaced with the same attack payload.

Request	Position	Payload
#1	1, 2	Item_1_List_1
#2	1, 2	Item_2_List_1

✦ **pitchfork**: In this attack type, Burp Intruder will use two or more payload lists, depending on the number of marked positions. During the first iteration, Burp will replace the marked positions with the corresponding first-attack payload of each list. In other words, it will use the first word of the first list for the first position and so on.

Request	Position	Payload
#1	1, 2	Item_1_List_1, Item_1_List_2
#2	1, 2	Item_2_List_1, Item_2_List_2

✦ **cluster bomb**: Similar to the **pitchfork** attack, multiple lists are used in this heuristic. However, in this case, Burp Intruder will iterate through all possible combinations.

Request	Position	Payload
#1	1, 2	Item_1_List_1, Item_1_List_2
#2	1, 2	Item_2_List_1, Item_1_List_2
#3	1, 2	Item_1_List_1, Item_2_List_2
#4	1, 2	Item_2_List_1, Item_2_List_2

Configuring payloads

After having selected all positions and the attack type, it is necessary to define the actual payloads. In the **payloads** tab of Burp Intruder, it is possible to define custom lists. With the term "attack payloads", the author of Burp refers to a list of common attack patterns or, in other words, a list of strings that if injected in vulnerable parameters may allow to detect security vulnerabilities.

As mentioned, some attacks require more than one payload list. By selecting a different list number from the **payload set** drop-down list, you can configure all payload sets, one by one. The second drop-down list allows you to define the payloads type.

Numerous types exist, although the most common are:

✦ **preset list**: By using a preset list, the user can load a list of attack vectors (wordlist) from external text files. Alternatively, it is possible to manually insert new words. Burp Professional users can also benefit from pre-loaded lists.

✦ **numbers**: By using a numbers list, Burp Intruder will automatically generate numbers based on the specific configuration. The user is required to define the starting and ending number, in addition to the number of steps.

✦ **dates**: Burp Intruder allows to specify a date format and automatically generate date from and to a specific day.

✦ **bruteforce**: By using this selection, Burp Intruder will generate all possible strings permutations given a characters set and the min/max length of the resulting string.

Let's say that we want to create an attack vector list from scratch to detect SQL Injection vulnerabilities:

1. In the **payloads** tab of Burp Intruder, select **1** in the **payload set** drop-down and make sure that it is marked as **preset list**.

2. Below, type a new vector in the text field and click on **add**. For example, you can start with a single quote (**'**)—a very common string to trigger SQL exceptions in database-driven applications.

3. Keep adding new strings. If you want to remove a string, select the item in the table and click on **delete**. Otherwise, if you want to remove all strings click on **clear**. Please refer to the following screenshot:
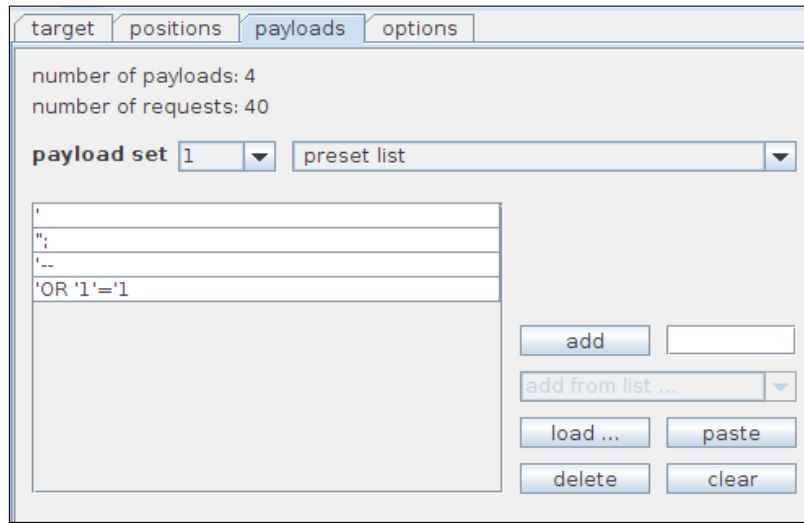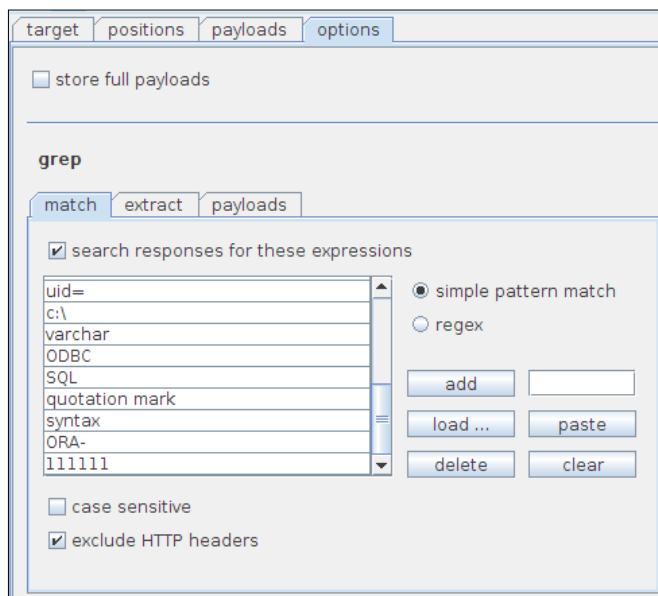

Defining payloads in Burp Intruder

At the bottom of the **payloads** tab, please note the payload encoding within requests section. By default, Burp Intruder will URL-encode all characters specified in this text field. For instance, the character **'** will be replaced with the corresponding `%27` URL encoding. If you don't want to encode a specific character, remove it from the list.

Additional Burp Intruder options

Burp Intruder is a very versatile tool, thus it has an extensive set of configuration options. You are encouraged to try and experiment yourself in order to understand the impact of each configuration option.

For example, in the **options** tab of Burp Intruder, you can configure the number of threads used by the tool and the timing between retries. Also, it is possible to specify a fixed or variable throttle time between requests.

Another interesting setting is represented by the **grep** section. The tool allows to specify strings or regular expressions to be searched in the web responses. This is extremely useful to detect exceptions and common error strings that can highlight underlying vulnerabilities.

Defining search expressions in Burp Intruder responses

Burp Intruder's **grep** functionality can be used to identify application behaviors by searching for specific keywords. Let's say that you are trying to access unauthorized resources, search for *error*, *invalid*, *unauthorized*, *incorrect*, and so on. If you notice requests without these keywords, it may be possible that you have successfully guessed a valid and accessible resource.

Launching an attack

At this point, everything is configured and we are ready to launch our attack. From the top menu in Burp, go to **intruder | start attack**. Burp will first verify the configuration and alert the user with a pop-up alert box in case of problems. Then, it will open a results window and start the actual attack.

> Burp Intruder, in the free edition, does not include all the advanced configuration options available in the professional edition. Most importantly, it does limit the speed of the attack by exponentially increasing the time after each request.

During the course of the attack, you can observe the results in the **results table** window. Depending on the configuration, Burp will display different columns, including the request ID, the used payload, the HTTP status code of the web response, the response time and, if you have enabled the **grep** functionality, the presence of specified strings. All these columns can be reordered and sorted. Moreover, Burp Intruder allows to export the results in a flat file.

> Discovering security vulnerabilities is mostly about dedication, patience, and a lot of motivation. As soon as you start to perform security assessments, you will discover how minimal changes in the responses help to identify flaws. Pay always attention to the different length and different HTTP status code of all web responses. Also, use the **grep** functionality to identify bugs that result in error messages and application exceptions.

# 5 – Manipulating and iterating web requests with Burp Repeater

In the previous section, we have seen how to automate and programmatically generate multiple requests with different payloads.

If you have already discovered a security vulnerability or you want to make sure that a particular endpoint is secure, it is sometimes necessary to manually repeat requests and carefully tune the attack vector. This is a trial-and-error approach that requires patience and experience.

Burp Repeater allows to modify each aspect of an HTTP request and to send it multiple times. Start by importing a web request, from any of the Burp Suite tool, using the traditional contextual menu:

1. From any tool (for example, Burp Proxy **history**) select a specific web request and right-click on it. Then, select **send to repeater**.

2. Go to Burp Repeater. You should see the entire content of the selected request. In addition, Burp Repeater will automatically fill the host and port numbers.

3. At this point, you can modify every aspect of this web request. Let's start by transforming a GET request to POST or vice-versa. In the **request** window, right-click and choose **change request method**.
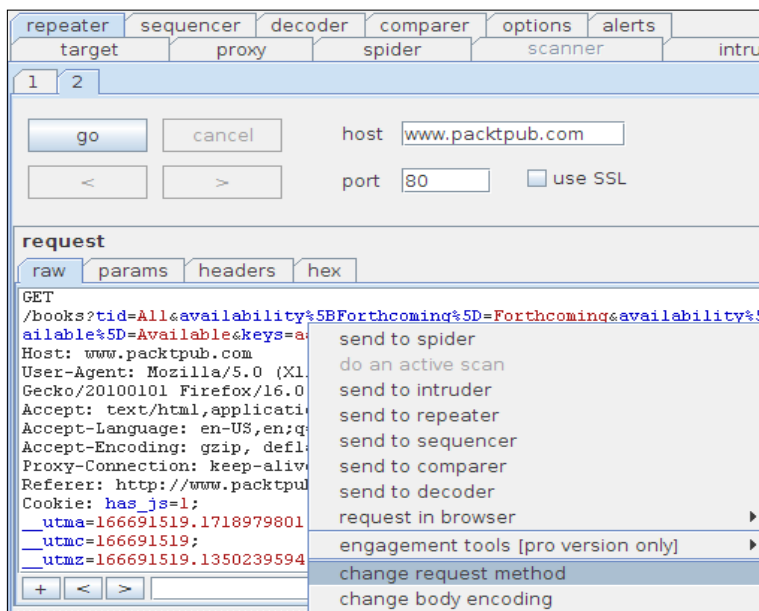
Change request method with Burp Repeater

4. Also, let's add a fake parameter by adding the string `&debug=true` at the end of the URL.

5. Finally, click on **go** to send the request. After a few seconds, Burp Repeater should be able to display the response. As usual, you can visualize the raw response, isolate parameters or headers, audit the HTML code, or even render the page.

Although Burp Repeater seems to be a very simple tool, it is actually very useful and incorporates some advanced functionalities. If you click on the **repeater** menu, as shown in the following screenshot, you can see a list of these features:
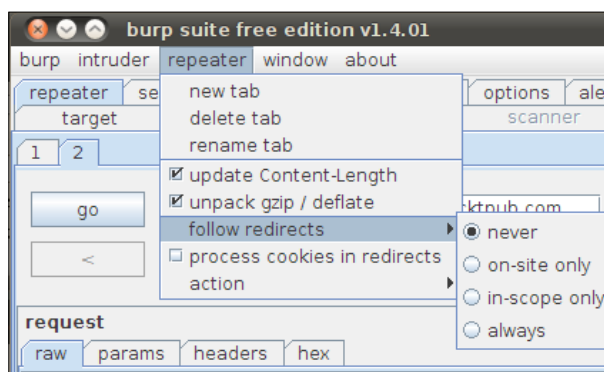
Burp Repeater options

The **update Content-Length** checkbox allows to dynamically update the `Content-Length` header field in the HTTP request. In this way, Burp Repeater will automatically calculate the size of the modified request before sending it over the wire.

The **follow redirects** option allows to select whether Burp Repeater should display the actual web response or, instead follow all redirects (`302 Redirect` status code) and display the landing page.

By selecting the **process cookies in redirects** checkbox, it is possible to ad operate the request's session tokens during the application redirects.

Finally, Burp Repeater allows to create, delete, or rename tabs. If you have discovered a vulnerability and you are trying to build a working exploit, it is very useful to create new tabs on each trial and use the rename functionality to assign a meaningful title for each tentative, in order to avoid confusion.

Also, if you are analyzing **Cross-Site Request Forgery** (**CSRF**) vulnerabilities or developing Cross-Site Scripting attacks, you can automatically create a proof-of-concept by right-clicking on your request and select **engagement tools | generates CSRF PoC**. This feature allows you to generate an HTML page that triggers the vulnerability.

6 – Analysing application data randomness with Burp Sequencer

Burp Sequencer allows you to analyze the predictability of application data, such as session cookies and anti-CSRF tokens. The tool allows you to easily collect and analyze data. Let's see how this tool can be used on a real example:

1. After having configured Burp Proxy, point your browser to `https://www.packtpub.com/login`.
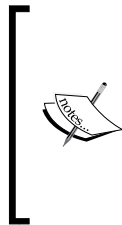
> Disclaimer! Although Burp Sequencer does not perform any injection attack, sending numerous requests to a remote web server may slow down the service and potentially interrupt the application. I suggest you experiment with Burp Sequencer against your own target. This example is provided for reference only.

2. In the **history** tab of Burp Proxy, select the **login** request. Right-click and select **send to sequencer**.

3. We have already imported the request to Burp Sequencer and we can now proceed with the setup. The **select request** table displays all the web requests imported in the tool. Having a single request, this item should be already selected. If not, click on the request in the table.

4. In the **identify token in response** section, within the **live capture** tab, it is necessary to configure how Burp Sequencer can identify tokens or other data that we want to analyze within the response page. To speed up the process, in the **cookie** and **form fields** drop-down lists, the tool will already display all cookies or form parameters that are present in the page. What is more, it is possible to manually select the data location.

5. Click on **manual selection**. In the response content, look for **edit-captcha-token**. This token is specific for the web application under analysis. You can also use the **search** textbox at the bottom of the form to easily find the token location.

6. As you can see, the value of the *edit-captcha-token* element contains a pseudo-random token. Let's study the entropy of this string. Move the mouse cursor over the entire string. Burp Sequencer will automatically fill the **after expression** and **at delimiter** text fields, positioned on the right. In practice, the tool helps to identify delimiters that can be used to extract the token from the response page.

7. Once done, click on **start** to collect samples. Burp Sequencer will repeat the same request multiple times and it will automatically extract the previously defined selection. The tool will also open a new window to monitor the process.
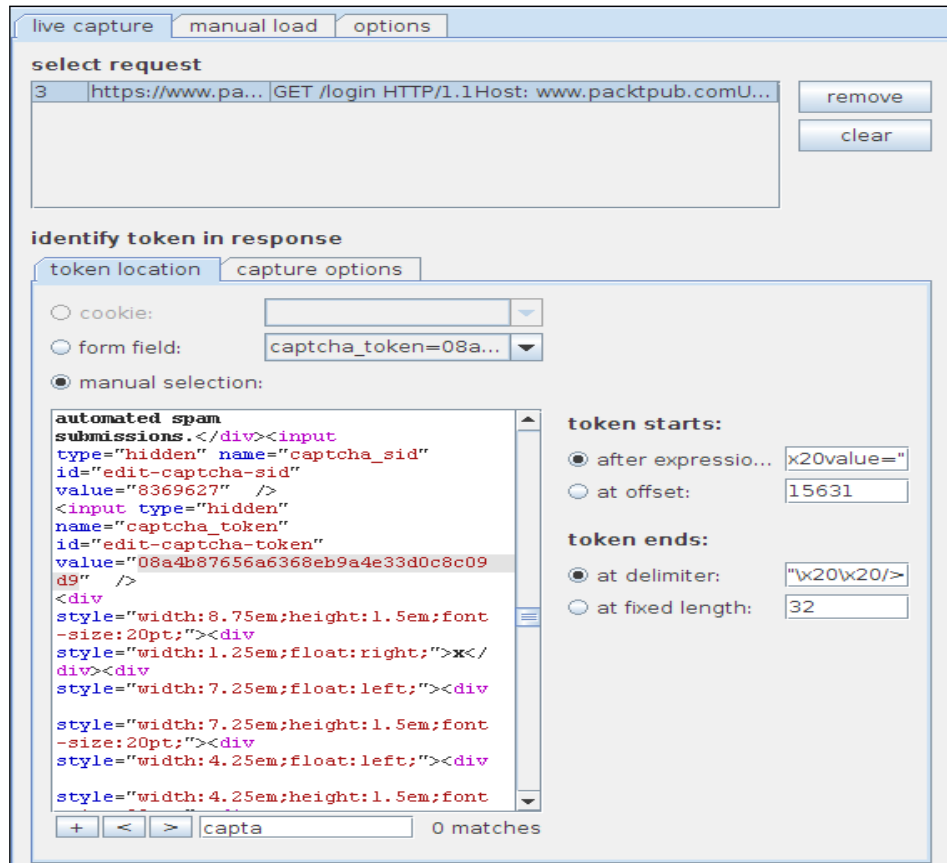

Configure Burp Sequencer live capture

Alternatively, it is possible to manually import tokens and application data from text files or by pasting the content of the clipboard. Go to the **manual load** tab of Burp Sequencer, if you prefer to load tokens in this way.

Upon collecting enough data, at least 100 samples, it is possible to pause the sample retrieval process by clicking the **pause** button. You will also notice that the **analyze now** button is active. Click on it to start the analysis phase. The following screenshot shows a paused live capture:
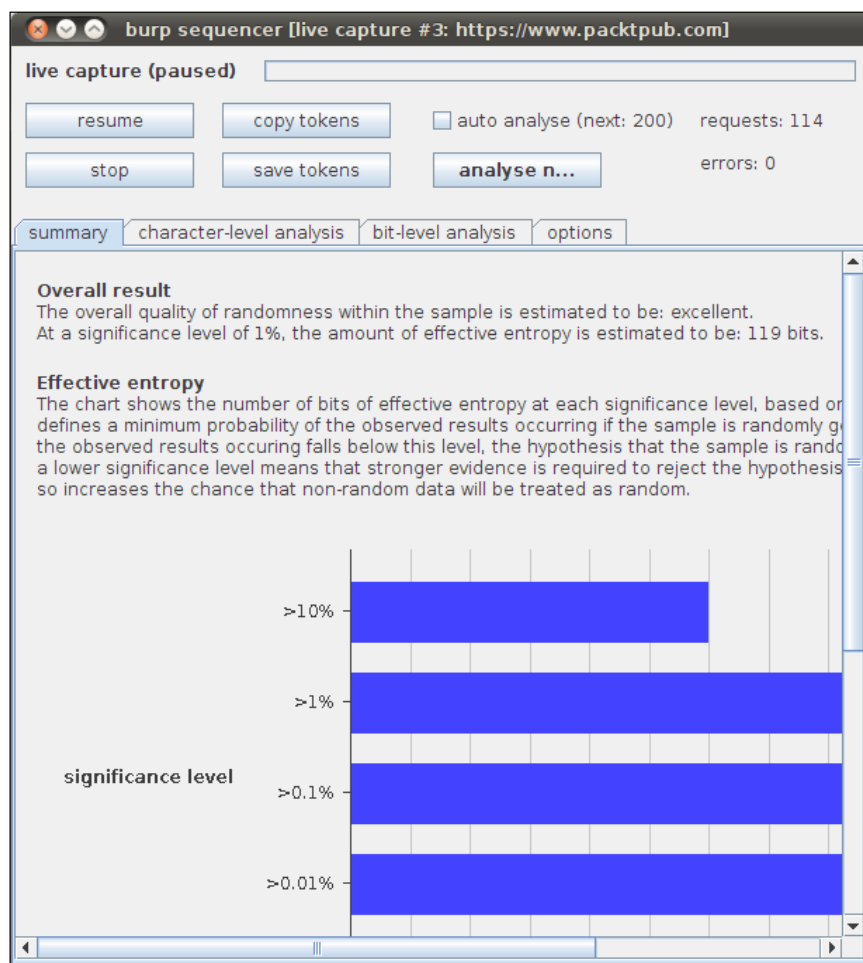
Burp Sequencer results

After a few seconds, Burp Sequencer should be able to display the results. If you want to verify the collected data, click on **copy tokens** and paste the content in your favorite text editor. Also, if you realize that you have not collected enough tokens, you can resume the process by clicking on the **resume** button.

Burp Sequencer results are displayed in the following three sections:

- ✦ The **summary** tab
- ✦ The **character-level analysis** tab
- ✦ The **bit-level analysis** tab

The **summary** tab provides a generic overview of the analysis. Usually, this view is sufficient to understand if the token is actually pseudo-random or not. In our example, the tool reports the following message:

The overall quality of randomness within the sample is estimated to be: excellent

Also, this tab reports an evaluation of the reliability of the analysis based on the number of collected samples.

Trying to evaluate the randomness of a non pseudo-random token will result in the following warning:

The overall quality of randomness within the sample is estimated to be: extremely poor

In general, Burp Sequencer will provide an estimate of the overall quality of randomness. Although the results are often correct, the tool is not always reliable. Advanced users can benefit from the **character-level** and **bit-level** tests to understand the actual predictability of the data. The **character-level analysis** tab includes multiple diagrams and comparison tables to understand the correlation between characters, positions, and character transitions within the token. The **bit-level analysis** tab is particularly useful to identify anomalies as it includes a chart indicating the degree of randomness confidence at each bit position.

7 – Decoding and encoding data with Burp Decoder

Burp Decoder is a simple but useful tool for encoding and decoding strings in multiple formats. During a web application's security audit, it is often necessary to evaluate the robustness of input validation security mechanisms. Encoding strings in multiple formats is a very common technique to bypass security controls and filters.

From any tool, you can import part of requests and responses in Burp Decoder by using the standard contextual menu item:

1. Select a string with the mouse cursor, right-click on it, and select **send to decoder**.
2. Once the string has been imported, it is possible to encode or decode it by selecting the appropriate encoding schema from the **decode as…** or **encode as…** scroll-down lists. The output is displayed in the text form below.

Users can visualize data in hexadecimal or text format, by selecting the **hex** or the **text** checkbox respectively. Encoding schemas supported by Burp include `URL encoding`, `HTML Entities` encoding, `Base64`, hexadecimal conversion, and `GZIP` compression encoding.
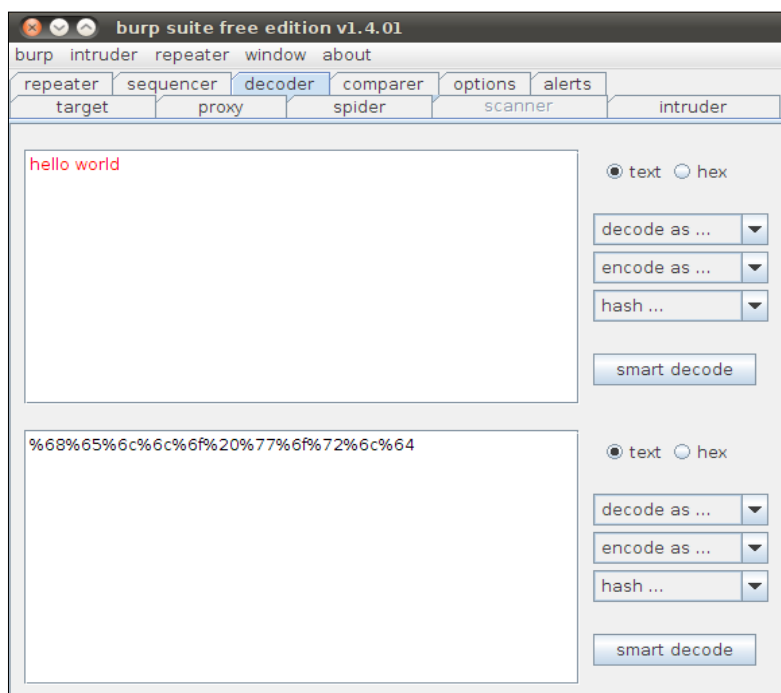
Burp Decoder

In addition, Burp Decoder allows to create message digests for common hash functions, including MD2, MD5, SHA, SHA256, and SHA512.

The output of a previous conversion can be used as input for a new conversion. This mechanism allows to concatenate multiple encoding techniques.

Also, by using the **smart decode** button, Burp will attempt to decode the content of a string by looking for recognizable formats. Although the heuristics do not always produce correct results, they can help during the identification of obfuscated content. To get an idea of possible encoding mechanisms, have a look at the *XSS Filter Evasion Cheat Sheet* available online at `https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet`.

8 – Comparing site maps

Uncovering access control vulnerabilities on large applications is a difficult task. Burp Suite's **compare site maps** functionality allows to compare two site maps and highlight differences. In a nutshell, this irreplaceable feature provides an easy way to map application resources using accounts with different access privileges and, sub-sequentially compare web responses.

For example, you can browse the application with a standard user account and then reiterate all requests using an administrative user. This approach may help to highlight privileges escalation bugs, normally referred to as vertical privileges escalation. Or, you can browse the application with two different users at the same level of privileges and verify access controls to resources. This approach may help to highlight horizontal privileges escalation bugs.

This feature is available in both the professional and the free version of the tool, although the latter does not allow you to import Burp state files as baseline for the comparison. Let's see how to take advantage of this feature with a concrete example. During this exercise, we will verify that a specific web application endpoint is available to authenticated users only:

1. After having configured Burp Proxy, point your browser to `https://www.packtpub.com/login`. Log in with your account and go to `https://www.packtpub.com/account`. Please note that the account page is a resource available to authenticated users only.

> Disclaimer! Although Burp's **compare site maps** functionality does not perform any injection attack, sending numerous requests to a remote web server may slow down the service and potentially interrupt the application. I suggest you experiment with this functionality against your own target. This example is provided for reference only.

2. In the **site map** tab of Burp Proxy, search and select the **account** endpoint.

3. As shown in the following screenshot, right-click on the endpoint and select **compare site maps**. Make sure to have selected the **account** item only:
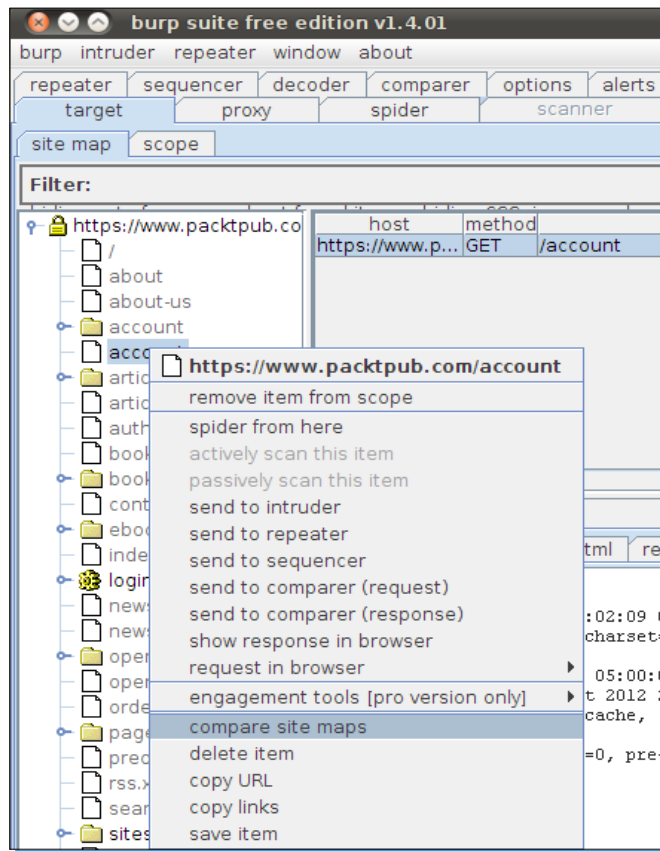

Launching "compare site map" from the site map contextual menu

4. Burp will open a new window containing the **compare site maps** wizard. The first step is to define the source for **site map 1**. This site map will be used as a baseline for our comparison. If you are using the free version of Burp, the **use current site map** option is the only option available to you. Select it and click **next**.

5. In the second step, you have to select the specific items to include during your comparison. For this example, select use only selected branches. This option will limit the compare site functionality to the single account endpoint. While testing your application, you may want to evaluate all endpoints by selecting use all items with responses. Also, you may want to limit the tool to Burp's in-scope sites only, by selecting **include in-scope items only**. Then, click **next**.
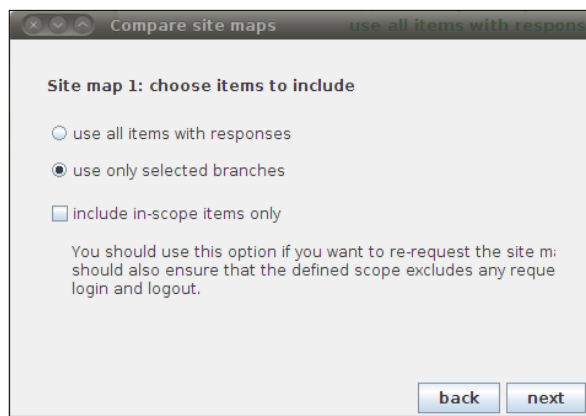
Defining all items to include in Burp compare site maps

6. In the third step, you have to define the source for **site map 2**. If you are using the free version of Burp, the **request map 1 again in a different session context** option is the only possibility available to you. Select it and click **next**.

7. Burp will use the current session (stored in Burp's **cookies jar**) to access all the resources defined in **site map 1**. During this exercise, we want to verify whether the **account** endpoint is available to both authenticated and unauthenticated users. In the previous steps, we have already recorded the **account** endpoint as seen by authenticated users. At this point, we need to invalidate our cookies and use the new session for **site map 2**. Minimize the **compare site maps** wizard and go to **options | sessions** in Burp Suite. Click on **view cookie jar**. This is the repository of all session tokens used by Burp.
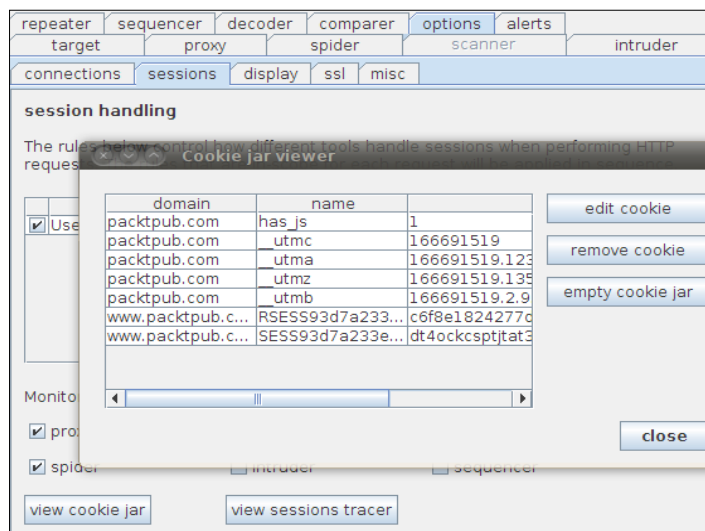

Opening Burp's cookies jar

8. As we want to simulate a non-authenticated user, we can simply tamper all cookies from the jar related to the `www.packtpub.com` domain. One by one, click on **edit cookie** and tamper the value by adding random strings. Confirm each operation by clicking on **done**. Finally, close the cookies jar using the **close** button.

9. Also, we need to force Burp Target site map to perform the comparison using those tampered cookies. In Burp Suite, go to **options | sessions**. Click on **edit**, on the right-hand side of the **session handling** table.

10. Burp will open a new window, named **Session handling rule editor**. Go to the **scope** tab and select the **target** checkbox in the **Tools scope** section. This is a very important step, thus make sure to properly configure the session handling as shown in the following screenshot. Finally, click on **Done** and get back to the **compare site maps** wizard window:
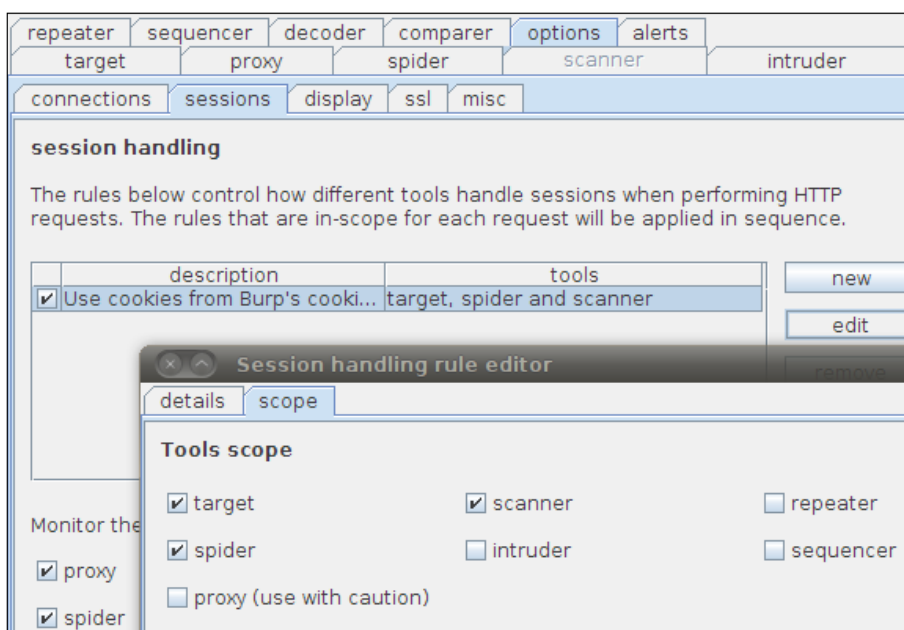
Burp's session handling rule editor

11. In the fourth step of the **compare site maps** wizard, Burp allows to customize the number of threads used during the analysis, in addition to other timing options. We can leave all options as they are and proceed further by clicking on **next**.

12. In the fifth step (**request matching**), it is suggested to use the default settings, as they will work effectively for most situations. Just click on **next**.

13. Again, in the sixth step (**responses comparisons**), it is suggested to use the default settings. Just click on **next**.

14. At this point, Burp will start requesting **site map 1** resources with the modified session, in order to build **site map 2**. Upon completing this process, Burp will automatically compute all differences and display the results to the user.
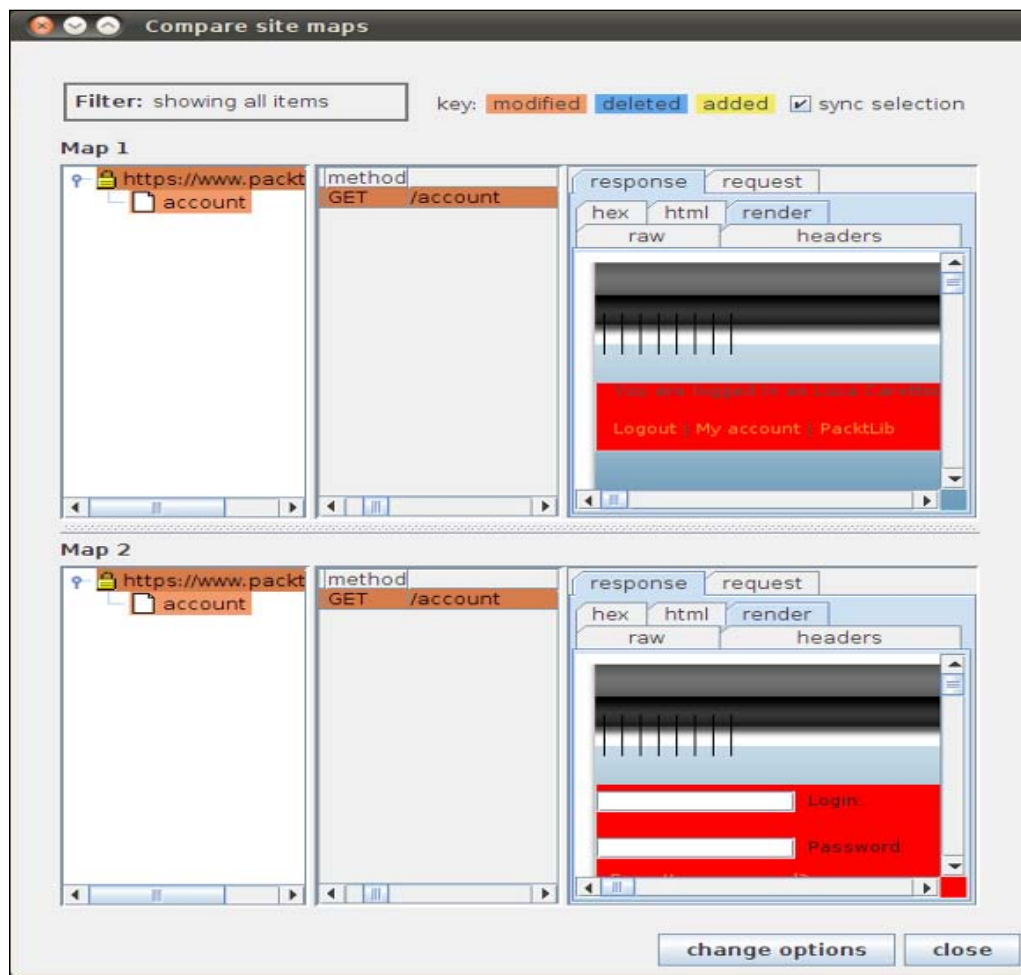
Burp's compare site maps results

The result page allows to easily compare resources from **site map 1** and **site map 2**. By using the **sync selection** checkbox, Burp will sync resources from the two sites, enabling you to simultaneously scroll down the two panels and items. In this case, we can easily verify that requesting the **account** endpoint using an unauthenticated session produced a different response—the user is redirected to the login page. As expected, this endpoint is not available for unauthenticated users.

During a real web application assessment, you can use this functionality to test all endpoints of your application. For instance, you can initially verify the differences between authenticated and unauthenticated sessions. Then, you can create **site map 1** and **site map 2** using two different users and verify the access control mechanisms in place. Finally, you may want to build **site map 1** and **site map 2** using a standard user and an admin user to make sure that privileged operations are not available for low-privileges accounts.

> Small differences between pages may be caused by dynamic components such as one-time tokens and time-dependent resources. In general, it is suggested to focus the analysis on added resources and significant changes within the responses.

People and places you should get to know

If you need help with Burp Suite, here are some people and places which will prove invaluable:

Official sites

- **Homepage**: `http://www.portswigger.net/`
- **Manual and documentation**: `http://www.portswigger.net/burp/help/`
- **FAQ**: `http://portswigger.net/burp/faq.html`

Articles and tutorials

- *Pentesting with Burp Suite: Taking the Web Back From Automated Scanners*: An all-in-one presentation that covers Burp Suite and the integration with other security tools. Worth checking it! Here are the links:
  - `http://www.securityaegis.com/pentesting-with-burp-suite-taking-the-web-back-from-automated-scanners/` (Slides and video)
  - `http://bit.ly/adYQrR` (Short link)
- A great tutorial on how to set up a development environment for building your Burp Extensions:
  - `http://console-cowboys.blogspot.com/2012/07/setting-up-burp-development-environment.html`
  - `http://bit.ly/RZFnPt` (Short link)
- A 15 minutes step-by-step video tutorial on how to use Burp. It covers most of the functionalities explained in this book. Nevertheless, it can help to refresh your Burp Suite skills; it can be found at `http://vimeo.com/11553558`.
- *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*. This book is probably one of the best resources for learning basic and advanced web application security techniques. Written by the same author of Burp Suite, the book covers web security using Burp as the reference tool; here is the link:
  - `www.amazon.com/The-Web-Application-Hackers-Handbook/dp/1118026470/`

Community

- **Official forums**: `http://forum.portswigger.net/`

  In particular, have a look at the **How do I?** and **Burp extensions** forums to learn how to perform advanced tasks and extend the tool with third-party add-ons.

Blog

✦ Although Burp Suite has been around for quite a few years and it is now considered the de-facto standard for testing web applications, there are not many blogs specifically dedicated to this tool. The best resource to receive updates and learn how to use new features, is probably the official blog: `http://blog.portswigger.net/`

Twitter

✦ Follow Dafydd Stuttard (the creator of Burp Suite) on Twitter at `https://twitter.com/PortSwigger`.

✦ Follow Luca Carettoni (the author of this book) on Twitter `https://twitter.com/_ikki`.

✦ Follow Michal Melewski on Twitter. Every now and then, he provides useful tips and tricks on how to use Burp; he can be found at

`https://twitter.com/carste1n`.

✦ Follow Jamie Finnigan on Twitter at `https://twitter.com/chair6`. He is the maintainer of Hiccup, a Python-based extension framework for Burp.

✦ For more Open Source information, follow Packt at `http://twitter.com/packtopensource`.

PACKT PUBLISHING

**Thank you for buying
Instant Burp Suite Starter**

About Packt Publishing

Packt, pronounced 'packed', published its first book "*Mastering phpMyAdmin for Effective MySQL Management*" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: www.packtpub.com.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to author@packtpub.com. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.
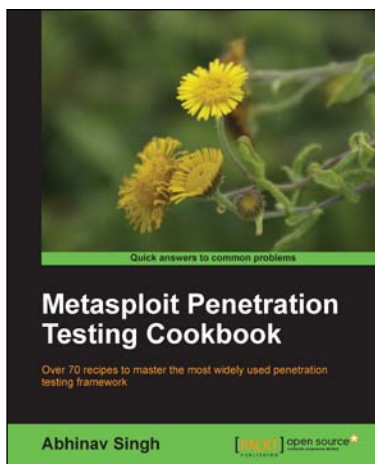
Metasploit Penetration Testing Cookbook

ISBN: 978-1-84951-742-3   Paperback: 268 pages

Over 70 recipes to master the most widely used penetration testing framework

1. More than 80 recipes/practical tasks that will escalate the reader's knowledge from beginner to an advanced level

2. Special focus on the latest operating systems, exploits, and penetration testing techniques

3. Detailed analysis of third party tools based on the Metasploit framework to enhance the penetration testing experience
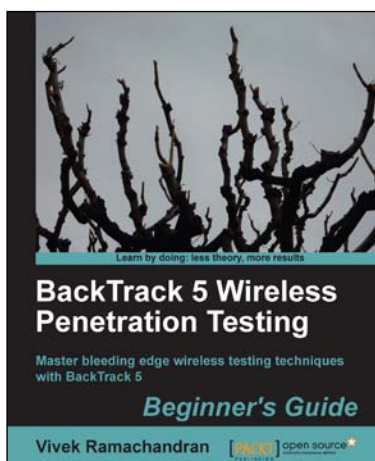
BackTrack 5 Wireless Penetration Testing Beginner's Guide

ISBN: 978-1-84951-558-0   Paperback: 220 pages

Master bleeding edge wireless testing techniques with BackTrack 5

1. Learn Wireless Penetration Testing with the most recent version of Backtrack

2. The first and only book that covers wireless testing with BackTrack

3. Concepts explained with step-by-step practical sessions and rich illustrations

4. Written by Vivek Ramachandran ¬— world renowned security research and evangelist, and discoverer of the wireless "Caffe Latte Attack"

Please check **www.PacktPub.com** for information on our titles

Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide

ISBN: 978-1-84951-774-4   Paperback: 414 pages

Learn to perform professional penetration testing for highly-secured environments with this intensive hands-on guide

1. Learn how to perform an efficient, organized, and effective penetration test from start to finish

2. Gain hands-on penetration testing experience by building and testing a virtual lab environment that includes commonly found security measures such as IDS and firewalls

3. Take the challenge and perform a virtual penetration test against a fictional corporation from start to finish and then verify your results by walking through step-by-step solutions
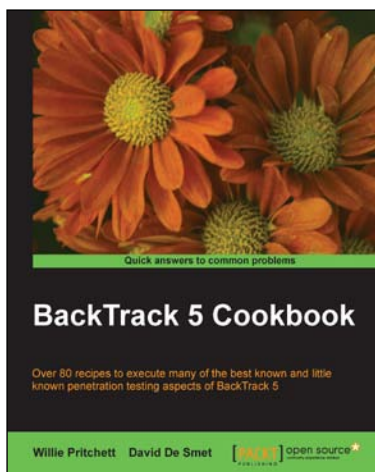
BackTrack 5 Cookbook

ISBN: 978-1-84951-738-6   Paperback: 296 pages

Over 90 highly-effective recipes to unleash your creativity with interactive art, graphics, computer vision, 3D, and more

1. Learn to perform penetration tests with BackTrack 5

2. Nearly 100 recipes designed to teach penetration testing principles and build knowledge of BackTrack 5 Tools

3. Provides detailed step-by-step instructions on the usage of many of BackTrack's popular and not-so-popular tools

Please check **www.PacktPub.com** for information on our titles