The threat to identity from new and unknown malware

P W Hodgson

Malicious code in the form of computer viruses, worms, trojans and spam bots represents a most dangerous and costly threat to fully interconnected networked information systems. Infected Web pages can seriously compromise client machines and networks. Infected electronic messages, commonly sent in the form of e-mail viruses, may not only damage individual machines but may also cause serious denial of service damage by flooding networks. Socially engineered messages in the form of phishing attacks can cause a general lack of confidence in electronic commerce. Conventional approaches to these problems focus on identifying legitimate messages, which is not always an easy or obvious task. Validating the identity of an electronic communication is a fundamental problem in the modern wired world.

The aim of this paper is to highlight the problem of authenticating the identity of electronic communications and to demonstrate an extra layer of protection with respect to e-mail systems, whereby attacks based upon the falsification of identity can be detected and eliminated with minimum impact on the system.

1. Introduction

Given that future networked systems will increase the frequency and complexity of connections between electronic devices for information gathering and exchange, it has become extremely important to find a method of detecting new malware that does not significantly impair the fast and efficient running of the network.

Modern computer viruses spread extremely rapidly and have done serious and widespread damage to both corporate and personal computers. The main approach to date in countering rapid propagation has been to limit the functionality of transmission programs (such as e-mail) or to limit the frequency of connections to and from machines. Examples of the former include preventing the transmission of file attachments within e-mail and disabling scripting. This has been the approach taken by the Microsoft e-mail security updates. The main problem with limiting functionality is that it limits the usefulness of systems. The problem with limiting connections [1] is that firstly, given the increase in connectivity that is taking place between components of networked systems, it will slow down communications, and, secondly, it does not prevent the spread of any malware, but merely slows it down.

One of the main problems in dealing with new and unknown malware is the problem of false positives. If the anti-virus software is not sufficiently clever to distinguish between legitimate and unknown malicious code it will generate unwarranted responses such as triggering alarms distracting administrators, quarantining or deleting files and generally disrupting the smooth running of the system. Examples of this problem can be found in neural-net approaches to behaviour classification which have been found wanting [2]. Traditional signature matching and heuristic prediction detection techniques do not stop all new and unknown malware. The most realistic solution is to try to allow for the maximum throughput of information between networked systems combined with a very effective way of detecting novel malicious code and rapidly preventing its spread.

Determining if something is malicious or not is a more general problem and is not only confined to e-mail viruses. Successful verification of the authenticity of any communication is fundamental to a successful transaction between different parties. As the modern world comes to rely on digital communications the modern criminal has also begun to learn how to manipulate the identity and authenticity of these communications.

2. Authenticity

In many ways the Achilles' heel of modern networked computer systems is the problem of authenticity. In the real world it is very difficult to impersonate another person. In the virtual world it is fairly straightforward

given the right information. How do you know that the e-mail you have received comes from whom it purports to come from? How do you know it is not carrying a malicious payload? How do you know the Web site you have reached is legitimate? In short, can you trust the communications you receive?

Access to modern computer systems hinges on those measures used to verify authenticity and is very much an all-or-nothing process. A criminal without digital 'proof' will not get very far but one that supplies the right information to the system will be trusted without question.

Not only can criminals impersonate people but they have become adept at impersonating organisations. Phishing attacks simulate established and reputable organisation's Web sites and trick the user into providing personal information that is then used by the criminal to either steal from the victim or use the victim's identity to commit further crimes. In a recent survey of US financial institutions it was found that 70% of accounts with bad debts were in fact bogus accounts.

Different forms of malware are now being combined to perform sophisticated attacks. Phishers have two main obstacles to overcome if they are to be successful. Firstly, they have to find customers of the organisation they have chosen to impersonate. To do this they use spam bots to send out vast numbers of fake messages. The majority of these will be caught by anti-spam software or will be binned by the recipient. However, a small proportion will get through to a potential target. Secondly, the victim has to be persuaded that what they are being asked to do is completely authentic. This is done by a combination of technology and social engineering. Victims are directed to Web sites that are so authentic (to the point of containing phishing warnings) it is impossible to tell the difference. Some attacks extract personal information by combining spam and trojans. Spam messages concerning an 'order' the user has made direct the user to a bogus Web site for further information. This Web site will contain no such information but will download a trojan program on to the victim's computer. The trojan will install a keystroke logger that is programmed to record a certain number of keystrokes after a certain trigger such as a bank name. This information is then sent off to the Phisher who uses it for criminal activity. The victim's machine is also compromised by the trojan in such a way that it can be used as a spam proxy to send out endless streams of spam to prospective new victims.

It is clear that criminals are now employing the services of expert hackers to construct sophisticated exploits. Most attacks are initiated through e-mail and are a form of social engineering in that the recipient is persuaded to open the e-mail or an attachment. E-mail techniques are now being combined with Web page spoofing by the criminal fraternity to create sophisticated attacks on people to extract identity data and money.

E-mail viruses spread by two basic methods. Either the mail message tries to trick the user into opening the attachment containing the virus (thus running it) by, for example, pretending to be some document or photo that would be of interest, or the virus writer exploits a vulnerability in the mail reader program or operating system so that the virus program will run when the e-mail is downloaded to the target computer, without requiring any action by the user. The second method is far more dangerous, but also less common. The only surefire way to prevent e-mail viruses spreading is not to use the e-mail system. However, less extreme safety measures can be taken such as not opening attachments or e-mails with suspicious subject lines. There have been many examples of e-mail subject lines designed to trick the recipient into opening the e-mail. They do this by spoofing identities.

Melissa and Loveletter were among the first viruses to illustrate the problem with e-mail attachments and trust. They made use of the trust that exists between friends or colleagues by getting the recipient to open the attach-ment. This happened with Melissa, AnnaKournikova, SirCam and several other similar e-mail viruses. Upon running, such viruses usually proceed to send themselves out to e-mail addresses harvested from the victim's address book, previous e-mails and Web page caches on the local machine.

W97M/Melissa contains the subject line 'Important Message From (username)' and it is usually from someone known because, whether that person knows it or not, they've been infected with Melissa and now they're passing the virus on. A new variant of the virus has a blank subject line, making the virus harder to notice. It executes a macro in a document attached to an e-mail, which forwards the document to 50 people in the user's Outlook address book. Melissa spread faster than any previous virus, infecting an estimated 1 million PCs.

It is clear that the bulk of e-mail viruses to date have been spread through a form of social engineering that manipulates the recipient into opening the e-mail and the attachment, thus spreading the virus. The 'from' field, the subject line and the message body are all designed to fool the reader into believing that the identity of the message sender is authentic. It is very difficult to guard against this sort of exploit as it relies on the inherent trust of the recipient in the veracity of the communication, and in many cases it is difficult to

tell that the communication is malicious. This means that some exploits will succeed and infect large numbers of machines by replicating rapidly across networks. In this event a successful e-mail virus will mean an increase in the number of e-mail messages sent from a client machine.

3. E-mail exploits

According to informed sources [3] e-mail viruses accounted for 90% of all virus attacks in 2001. Late in 2001 two viruses/worms that were spread by e-mail, Nimda and Goner, led to damage estimated at over $12 billion [4]. E-mail is the most widely used Internet application and hence the best method of rapidly distributing malicious code.

Technically speaking there is nothing that a plain text e-mail can do to damage a computer. However, code that is connected in some way to an e-mail can execute and propagate itself rapidly. The three main ways in which code can be connected to e-mail are through attachments, MIME exploits and embedded script. Attachments are infected files that are attached to an e-mail and only infect if they are opened and executed. Examples of these are any executable file but in particular those with the following extensions — COM, EXE, PIF, SCR, VBS. Files can of course be disguised as something that they are not and as being non-executable. In this case they are known as trojans and can masquerade as any type of file. Multipurpose Internet Mail Extensions (MIME) have in certain situations allowed code to be executed automatically. Viruses embedded in HTML e-mail have exploited this weakness whereby when the e-mail page is displayed the code executes. This is very similar to embedded script whereby virus writers exploit the fact that HTML mail can include JavaScript or VBScript code. These exploits rely on Windows Scripting Host which is installed automatically with Internet Explorer to execute scripts embedded in HTML. Code runs as soon as the page is viewed which means that nothing is explicitly opened by the user.

Many recent viruses are in fact more accurately described as worms, or combinations of worms, viruses and trojans. Technically a worm is code that copies itself from machine to machine without explicit user action. They rely on file-sending mechanisms that are built into many modern applications to send themselves to other machines. Recent examples of highly destructive worms are Nimda, Goner and CodeRed.

Most systems administrators have reacted to e-mail viruses by restricting the functionality of e-mail systems. Rather than risk the possibility that someone might open an infected file attachment, many administrators have banned the usage of certain attachments. Likewise use of HTML based e-mail systems such as Microsoft Hotmail has been discouraged to prevent malicious scripts from executing. Users are now encouraged not to open e-mails from unknown users or those that contain unusual content. This whole situation is akin to a postal service that recommends users do not open mail from unknown sources for fear of attack.

Antivirus scanners alone cannot prevent e-mail attacks. They can protect against known and documented viruses, but will do very little against new, unknown viruses that use novel malicious techniques. The heuristics employed by these scanners to recognise unknown viruses very often cause more trouble than they are worth in the form of false positives. Restrictive ways for the client to protect against e-mail attacks, therefore, are to disable the automatic downloading of files and ActiveX controls, and to disable scripting. For corporate networks additional preventive measures can be taken on the e-mail server(s). This is done by installing an e-mail gateway to provide a barrier between the Internet and the corporate intranet. These gateways can be configured to automatically scan all attachments (including compressed files) and block selected types such as COM, VBS, etc. Examples of this are Symantec AntiVirus enterprise edition and McAfee Webshield. Very recent systems allow the user to configure behaviours to scan for, and consequent actions to take. Examples of this are McAfee Outbreak Manager. The problem with e-mail scanners is that they will sometimes block valid e-mail as well as potential viruses. More importantly, configurable behaviour-based systems will always be vulnerable to behaviours that have not been specified in the search criteria, or are very far removed from known exploits.

Unsurprisingly, companies such as Microsoft have put measures in place to enhance security on e-mail systems. Patches have been released for both Outlook and Outlook Express which limit functionality and make e-mail harder to use. On Exchange servers, Microsoft has implemented the ability to monitor the number of incoming and outgoing messages in the message queue. This is useful because it gives a direct indication of e-mail activity within the network. The only problem with this is the fact that it does not give an indication of exactly where the growth in e-mail activity is coming from.

4. Viral E-mail Network Utilisation Symptomiser (VENUS)

The VENUS system has been designed and tested as a research project to specifically address this problem. A rapid increase in the number of outgoing messages is a definite sign that a rapidly spreading virus/worm is at

work. In recent years virus writers have tried to cause maximum damage as fast as possible by getting malicious code to spread very rapidly. If the e-mail system can monitor the rate of flow of outgoing messages and know exactly where they are coming from, it can prevent the spread of viruses very quickly and thereby minimise viral damage.

Tools such as NT Performance Monitor can monitor overall message volumes, but cannot monitor volumes from individual client machines. Furthermore, and more importantly, they are not proactive in preventing the spread of any virus.

VENUS is a solution to the above problem on the Microsoft platform. It works by rapidly detecting novelty and preventing the spread of any infection in the following way.

A Microsoft Exchange server logs a record of every incoming and outgoing e-mail in the server tracking log. These are updated in real time as traffic flows through the network. VENUS reads a number of entries (user-definable) from the end of the tracking-log file at user-definable intervals. Every tracking log entry contains a number of fields, such as message ID, time, date and size, describing the e-mail. The originator field shows which company user sent the message. Many organisations use organisational unit codes (OUCs) as well as unique user IDs to identify their employees. VENUS parses the log file selection and extracts a list of originators (userids). These userids are then matched against an OUC/userid database to extract the relevant OUC for every userid. BT comprises 115 000 userids which are divided up into 10739 OUCs. A two-dimensional grid is constructed to accommodate all BT OUCs. A list of all unique OUCs is compiled and the number of messages emanating from each OUC is computed.

The maximum number of OUCs in any sample is computed and a colour legend is normalised using 40 different colours to show the full range of OUC e-mail emissions (see Fig 1). In this way a visual map is used to display not only the number of e-mail emissions for any given time period but also the pattern of emissions throughout the organisation, and therefore the spread of any infection.

Recent research [5] has shown that e-mail networks exhibit scale-free properties which means that the threshold for the propagation rate above which a network infection spreads, is much lower than in disordered or random networks. If highly connected network nodes can be identified, viral propagation will be greatly reduced and controlled. Organisational unit codes are used because in certain cases viruses will use local address books as opposed to global address lists to find new victims, in which case infections are more likely to be grouped within organisational units. Furthermore,

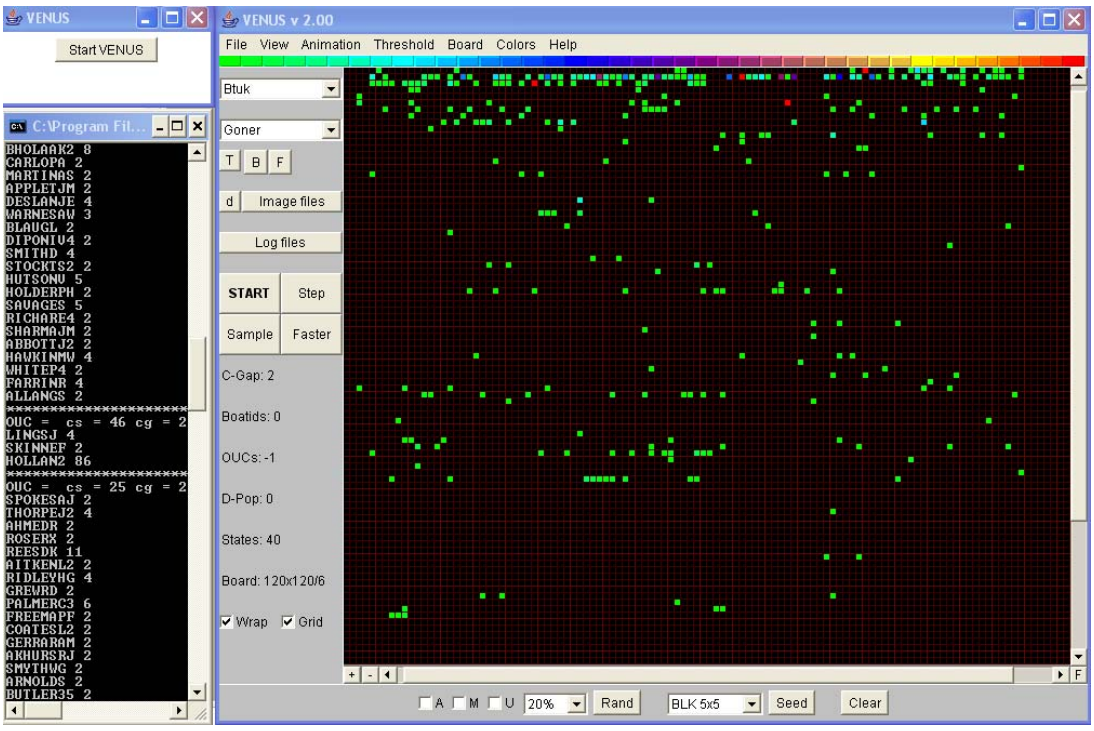

Fig 1 The VENUS user interface.

administrators can use this visual representation to see how any infection is spreading throughout the corporate intranet.

Once a threshold of emissions (user-defined) has been exceeded by any single client machine for any given sample, the system administrator has the option of capturing an example of the viral suspect and sending it off to a virus laboratory for analysis. Deciding whether these emissions are legal is a major problem with anomaly detection technology. False positives can be extremely costly in terms of administrator time and resources. VENUS overcomes this problem in a novel way by forcing the user to register bulk e-mail emissions in a user database that automatically registers bulk mailings over a certain threshold. Every time a suspect number of e-mails are detected from any one client, the user database is checked for legality in order to minimise false positives.

If shown to be anomalous, all messages sent from a suspect client are put into recall. This is another innovation of the VENUS system because Microsoft Exchange does not currently permit automated mass recall of e-mail messages. This effectively means that any messages sent from the client server to any destination server will be recalled or quarantined while the suspect code is being analysed. If the suspect proves positive, all suspects are deleted and infected machines are automatically barred from the network until cleaned. If negative, all e-mails are allowed to continue. The only impact on the system will be the time taken to send off a sample for analysis to a virus laboratory. Analysis of the novel code is the only part of the whole process that requires human intervention.

In many cases a large company will have more than one e-mail server and will want to automate the whole process of novel virus detection, whereby the e-mail administrator plays no active part as all decisions are fully automated. The situation might arise where a new virus manages to infect a client machine within the corporate intranet. If successful, it will select new addresses to infect from the company's global address book. It will then mail itself to as many victims as fast as possible (see Fig 2). If we assume that the first machine to be infected is client A which is served by server X, infected client A will send out messages across the intranet to clients B, C, D, E and F. If these clients are in different parts of the company they could easily be situated in different physical locations across the world, which means they will be served respectively by servers X,Y and Z. If client B becomes infected it could well send out messages to further clients which are in turn served by different servers. Clearly one can see how rapidly infection can spread across the company. If a VENUS system is installed on all company servers it is easy to

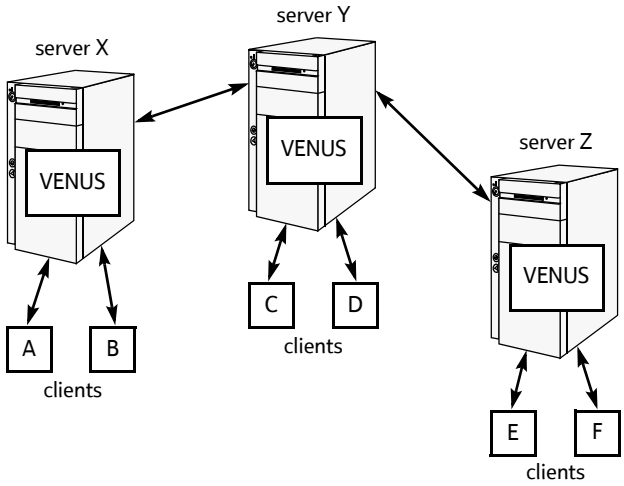

Fig 2 Interconnected corporate solution.

see how viral spread can be contained. If an alert is raised on server X (which will be monitoring client A) all destination servers of client A messages (i.e. servers Y and Z) will be notified, and client A messages will be quarantined on those servers. If a message has been opened on destination client C for example, and has gone on to infect further clients, then server Y VENUS will notify the respective destination servers and a catch-up process will begin. The only scenario in which this catch-up will fail is if clients open their e-mail as soon as it appears in their inbox, because opening e-mail means that the message is copied from the server message store to the client and executed on the client. However, most users do not open e-mail immediately and in many cases a significant time gap takes place. This time gap means that interconnecting VENUS systems on all company e-mail servers can rapidly quarantine all suspect e-mails and contain a potential threat. If it is found that the collected e-mail sample is benign, all quarantined messages are unfrozen and allowed to continue.

In this way the solution to the whole problem of anomalous e-mail emissions can be automated throughout the company, and does not require administrator intervention. This even includes sending off a viral sample to a virus laboratory (where human analysis might be required) and acting on the response. Furthermore, this solution minimises the problem of false positives for rapid infection viruses.

5. Conclusions

It is not always obvious that digital communications are what they appear to be. Web pages can be easily spoofed and malicious e-mails can be disguised to hide the true identity of the sender. If the malicious intent is not apparent then methods can be used to detect higher level symptoms of an infection as shown here in the case of e-mail viruses.

References

1 Williamsom M M: 'Throttling Viruses: Restricting propagation to defeat malicious mobile code', HP Labs, UK (2002).

2 Kephart J O, Sorkin G B, Swimmer M and White S R: 'Blueprint for a Computer Immune System', Virus Bulletin International Conference, San Francisco (1997).

3 Kaspersky Lab — http://www.kaspersky.com/

4 Networks Associates Technology Inc — http://www.nai.com/

5 Ebel H, Mielsch L and Bornholdt S: 'Scale-free topology of e-mail networks', Physical Review E 66, 035103(R) (2002).

Prior to working for BT Paul Hodgson worked on computational novelty and creativity at the University of Sussex. In 1994 he won a British Computer Society Award Medal for innovation in IT and in 1995 was invited to become a Fellow of the Royal Society of Arts. Prior to this he worked in Europe as a musician, teacher and computer music developer, having developed the first ever computer program to play live on television with a world-class jazz musician. He has worked for BT as a senior researcher since 1997. After working on content management he moved to the artificial life research group in 1998 to look at nature-inspired solutions to computer networking problems. Having researched immune systems he moved to the newly formed Security Research Centre and devised the VENUS system as a fully automated response to the penetration of corporate e-mail systems by malicious code. He is currently working on novel approaches to wireless security.