Hacker Highschool
SECURITY AWARENESS FOR TEENS

LESSON 6
MALWARE

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM
INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES

"License for Use" Information

The following lessons and workbooks are open and publicly available under the following terms and conditions of ISECOM:

All works in the Hacker Highschool project are provided for non-commercial use with elementary school students, junior high school students, and high school students whether in a public institution, private institution, or a part of home-schooling. These materials may not be reproduced for sale in any form. The provision of any class, course, training, or camp with these materials for which a fee is charged is expressly forbidden without a license including college classes, university classes, trade-school classes, summer or computer camps, and similar. To purchase a license, visit the LICENSE section of the Hacker Highschool web page at www.hackerhighschool.org/license.

The HHS Project is a learning tool and as with any learning tool, the instruction is the influence of the instructor and not the tool. ISECOM cannot accept responsibility for how any information herein is applied or abused.

The HHS Project is an open community effort and if you find value in this project, we do ask you support us through the purchase of a license, a donation, or sponsorship.

Table of Contents

Contributors

Simon Biles, Computer Security Online Ltd.

Kim Truett, ISECOM

Pete Herzog, ISECOM

Marta Barceló, ISECOM

ISECOM
INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES

/// computer security online

6.0 Introduction

"Malware" are programs or parts of programs that have a malicious ("Mal") or unpleasant effect on your computer security. This covers many different terms that you may have heard before, such as "Virus", "Worm" and "Trojan" and possibly a few that you haven't like "Rootkit", "Logicbomb" and "Spyware". This lesson will introduce, define and explain each of these subdivisions of malware, will give you examples, and will explain some of the countermeasures that can be put into place to restrict the problems caused by malware.

6.1 Viruses (Virii)

6.1.1 Introduction

Virus – this is the most common type of malware that people will be aware of. The reason that it is known as a virus, rather than anything else, is historical. The press ran the stories of the first computer virus at the same time as articles concerning the spread of AIDS. At the time, there were simple parallels that could be easily drawn between the two, propagation through interaction with a contaminated party, the reliance on a host and the ultimate "death" of anything infected. This resulted, and still does occasionally, in concerns that people could become "infected" with a computer virus.

6.1.2 Description

Viruses or virii are self-replicating pieces of software that, similar to a biological virus, attach themselves to another program, or, in the case of "macro viruses", to another file. The virus is only run when the program or the file is run or opened. It is this which differentiates viruses from worms. If the program or file is not accessed in any way, then the virus will not run and will not copy itself further.

There are a number of types of viruses, although, significantly, the most common form today is the macro virus, and others, such as the boot sector virus are now only found "in captivity".

6.1.2.1 Boot Sector Viruses

The boot sector virus was the first type of virus created. It hides itself in the executable code at the beginning of bootable disks. This meant that in order to infect a machine, you needed to boot from an infected floppy disk. A long time ago, (15 years or so) booting from floppy was a relatively regular occurrence, meaning that such viruses were actually quite well spread by the time that people figured out what was happening. This virus (and all other types) should leave a signature which subsequent infection attempts detect, so as not to repeatedly infect the same target. It is this signature that allows other software (such as Anti-Virus-software) to detect the infection.

6.1.2.2 The Executable File Virus

The Executable File virus attaches itself to files, such as .exe or .com files. Some viruses would specifically look for programs which were a part of the operating system, and thus were most likely to be run each time the computer was turned on, increasing their chances of successful propagation. There were a few ways of adding a virus to an

executable file, some of which worked better than others. The simplest way (and the least subtle) was to overwrite the first part of the executable file with the virus code. This meant that the virus executed, but that the program would subsequently crash, leaving it quite obvious that there was an infection – especially if the file was an important system file.

6.1.2.3 The Terminate and Stay Resident (TSR) Virus

TSR is a term from DOS where an application would load itself into memory, and then remain there in the background, allowing the computer to run as normal in the foreground. The more complex of these viruses would intercept system calls that would expose them and return false results - others would attach themselves to the 'dir' command, and then infect every application in the directory that was listed – a few even stopped (or deleted) Anti-Virus software installed onto the systems.

6.1.2.4 The Polymorphic Virus

Early viruses were easy enough to detect. They had a certain signature to identify them, either within themselves as a method to prevent re-infection, or simply that they had a specific structure which it was possible to detect. Then along came the polymorphic virus. Poly – meaning multiple and morphic – meaning shape. These viruses change themselves each time they replicate, rearranging their code, changing encryption and generally making themselves look totally different. This created a huge problem, as instantly there were much smaller signatures that remained the same – some of the "better" viruses were reduced to a detection signature of a few bytes. The problem was increased with the release of a number of polymorphic kits into the virus writing community which allowed any virus to be recreated as a polymorph.

6.1.2.5 The Macro Virus

The Macro Virus makes use of the built-in ability of a number of programs to execute code. Programs such as Word and Excel have limited, but very powerful, versions of the Visual Basic programming language. This allows for the automation of repetitive tasks, and the automatic configuration of specific settings. These macro languages are misused to attach viral code to documents which will automatically copy itself on to other documents, and propagate. Although Microsoft has turned off the feature by default now on new installations, it used to be that Outlook would automatically execute certain code attached to e-mails as soon as they were read. This meant that viruses were propagating very quickly by sending themselves to all of the e-mail addresses that were stored on the infected machine.

Exercises:

1) Using the internet, try to find an example of each of the above types of virus.

2) Research the Klez virus:

- what is its "payload"
- the Klez virus is well know for SPOOFING. What is spoofing, and how does Klez use it?
- you just learned that your computer is infected with Klez. Research how to remove it.

3) You just received an email with the following Subject "Warning about your email account". The body of the message explains that your inappropriate use of email will

result in your losing Internet privileges and that you should see the attachment for details. But you haven't done anything weird with email as far as you know. Are you suspicious? You should be. Research this information and determine what virus is attached to this message. (HINT: When you start thinking of breakfast – you're correct.)

6.2 Worms

6.2.1 Introduction

Worms are older than viruses. The first worm was created many years before the first virus. This worm made use of a flaw in the UNIX finger command to quickly bring down most of the Internet (which was much smaller at that time). This following section deals with worms.

6.2.2 Description

A worm is a program that, after it has been started, replicates without any need for human intervention. It will propagate from host to host, taking advantage of an unprotected service or services. It will traverse a network without the need for a user to send an infected file or e-mail. Most of the large incidents in the press recently have been worms rather than viruses.

Exercises:

1) Using the internet, see if you can find the first worm that was ever created.

2) Find out what vulnerability the Code Red and Nimda worms use to propagate.

6.3 Trojans and Spyware

6.3.1 Introduction

The first Trojan Horse was created by the Greeks several thousand years ago. (Think about the film "Troy" if you have seen it). The basic concept is that you sneak something nasty into an otherwise secure computer in the guise of something nicer. This can range from a downloaded game trailer to an e-mail promising naked pictures of your favorite celebrity. This section covers trojans and spyware.

6.3.2 Description

Trojans are pieces of malware which masquerade as something either useful or desirable in order to get you to run them. At this point they may well do something unpleasant to your computer such as install a backdoor or rootkit (see section 6.4), or - even worse - dial a premium rate phone number that will cost you money.

Spyware is software that installs itself surreptitiously, often from websites that you might visit. Once it is installed it will look for information that it considers valuable. This may be usage

statistics regarding your web surfing, or it might be your credit card number. Some pieces of spyware blow their cover by rather irritatingly popping up advertisements all over your desktop.

Exercises:

1) Using the internet, find an example of a trojan and of spyware.

6.4 Rootkits and Backdoors

6.4.1 Introduction

Often when a computer has been compromised by a hacker, they will attempt to install a method to retain easy access to the machine. There are many variations on this, some of which have become quite famous – have a look on the Internet for "Back Orifice" !

6.4.2 Description

Rootkits and backdoors are pieces of malware that create methods to retain access to a machine. They could range from the simple (a program listening on a port) to the very complex (programs which will hide processes in memory, modify log files, and listen to a port). Often a backdoor will be as simple as creating an additional user in a password file which has super-user privileges, in the hope that it will be overlooked. This is because a backdoor is designed to bypass the system's normal authentication. Both the Sobig and MyDoom viruses install back doors as part of their payload.

Exercises:

1) Find on the Internet examples of rootkits and backdoors.
2) Research "Back Orifice", and compare its functionality to the commercially available offering for remote systems management from Microsoft.

6.5 Logicbombs and Timebombs

6.5.1 Introduction

Systems programmers and administrators can be quite odd people. It has been known for there to be measures on a system that will activate should certain criteria be met. For example: a program could be created that, should the administrator fail to log in for more than three weeks, would start to delete random bits of data from the disks. This occurred in a well-known case involving a programmer at a company called General Dynamics in 1992. He created a logicbomb which would delete critical data and which was set to be activated after he was gone. He expected that the company would then pay him significant amounts to come back and fix the problem. However, another programmer found the logic bomb before it went off, and the malicious programmer was convicted of a crime and fined $5,000

US dollars. The judge was merciful – the charges the man faced in court carried fines of up to $500,000 US dollars, plus jail time.

6.5.2 Description

Logicbombs and Timebombs are programs which have no replication ability and no ability to create an access method, but are applications or parts of applications that will cause damage to data should they become active. They can be stand-alone, or part of worms or viruses. Timebombs are programmed to release their payload at a certain time. Logicbombs are programmed to release their payload when a certain event occurs.

The idea behind timebombs, however, is also a useful one. Timebomb programming is used to allow you to download and try a program for a period of time – usually 30 days. At the end of the trial period, the program ceases to function, unless a registration code is provided. This is an example of non-malicious timebomb programming.

Exercises:

1) What other reasonable (and legal) uses might there be for timebomb and logicbomb coding.

2) Think about how you might detect such a program on your system.

6.6 Countermeasures

6.6.1 Introduction

There are a number of ways that you can detect, remove and prevent malware. Some of these are common sense, others are technological alternatives. The following section highlights some of these, with a brief explanation and examples.

6.6.2 Anti-Virus

Anti-Virus-software is available in many commercial and Open Source versions. These all work following the same method. They each have a database of known viruses and they will match the signatures of these against the files on the system to see if there are any infections. Often though, with modern viruses, these signatures are very small, and there can often be false positives - things that appear to be viruses that are not. Some virus scanners employ a technique known as heuristics, which means that they have a concept of what a virus "looks like" and can determine if an unknown application matches these criteria. Recently AntiVirus software has also crossed the boundary into Host Based Intrusion Detection, by keeping a list of files and checksums in order to increase the speed of scanning.

6.6.3 NIDS

Network intrusion detection is similar to AntiVirus software. It looks for a particular signature or behavior from a worm or virus. It can then either alert the user, or automatically stop the network traffic carrying the malware.

6.6.4 HIDS

Host based Intrusion Detection systems, such as Tripwire, are capable of detecting changes made to files. It is reasonable to expect that an application, once it is compiled, should not need to change, so watching various aspects of it, such as its size, last modification date and checksum, make it instantly obvious that something is wrong.

6.6.5 Firewalls

Worms propagate across the network by connecting to vulnerable services on each host. Apart from ensuring that none of these vulnerable services are running, the next best thing is to ensure that your firewall does not allow connections to these services. Many modern firewalls will provide some form of packet filtering similar to a NIDS which will rule out packets matching a certain signature. (Firewalls are discussed in more detail in section 7.1.2).

6.6.6 Sandboxes

The concept of a sandbox is simple. Your application has its own little world to play in and can't do anything to the rest of your computer. This is implemented as standard in the Java programming language, and can also be implemented through other utilities such as chroot in Linux. This restricts the damage that any malware can do to the host operating system by simply denying it the access required. Another option is to run a full machine inside a machine using a virtual machine product such as VMWare. This isolates the virtual machine from the host operating system, only allowing access as defined by the user.

Example – http://www.vmware.com – VMWare virtual machines

Exercises:

1. Matching Game: Research each of the following and match it to the type of countermeasure that it is:
   1. http://www.vmware.com — NIDS
   2. http://www.tripwire.org — Antivirus
   3. http://www.snort.org — Firewalls
   4. http://www.checkpoint.com — Sandboxes
   5. http://www.sophos.com — HIDS
2. Research Spybot Search and Destroy and determine what type of malware it protects your computer again.
3. Research how NIDs and HIDS works.
4. Research Firewall solutions on the net.
5. Look up "chroot" on the internet. Read about this type of "jail" or "sandbox".

6.7 Good Safety Advice

There are a number of simple things that you can do in order to minimize your risk to Malware.

- Only download from reputable sources (that means no W4R3Z, please.)
- Don't open e-mail attachments from people you don't know.
- Don't leave macros enabled by default in your applications.
- Keep your OS and applications up to date with patches.
- If downloading and installing software with a checksum – check the checksum.

Further Reading

AV Vendor Sites -

http://www.sophos.com
http://www.symantec.com
http://www.fsecure.com

All of these sites have databases listing details of trojans, viruses and other malware. There are also detailed descriptions of the functioning of the above.

http://www.cess.org/adware.htm
http://www.microsoft.com/technet/security/topics/virus/malware.mspx
http://www.zeltser.com/sans/gcih-practical/revmalw.html
http://www.securityfocus.com/infocus/1666
http://www.spywareguide.com/
http://www.brettglass.com/spam/paper.html
http://www.lavasoft.nu/ - AdAware Cleaning Software (Freeware Version)
http://www.claymania.com/removal-tools-vendors.html
http://www.io.com/~cwagner/spyware.html
http://www.bo2k.com/
http://www.sans.org/rr/catindex.php?cat_id=36