PENETRATION TESTING FUNDAMENTALS

A Hands-On Guide to Reliable Security Audits

CHUCK EASTTOM

Penetration Testing Fundamentals

A Hands-On Guide to Reliable Security Audits

Chuck Easttom

PEARSON

800 East 96th Street, Indianapolis, Indiana 46240 USA

Flame

No modern discussion of viruses would be complete without a discussion of Flame. This virus, which first appeared in 2012, targeted Windows operating systems. The first item that makes this virus notable is that it was specifically designed by the U.S. government for espionage. Many sources, including the Washington Post, UK Telegraph, and Reuters, have all stated that Flame was created by the U.S. government. It was discovered in May 2012 at several locations, including Iranian government sites. Flame is spyware that can monitor network traffic and take screenshots of the infected system.

Trojan Horses

Recall from earlier chapters that *Trojan horse* is a term for a program that looks benign but actually has a malicious purpose. We have already seen viruses that are delivered via a Trojan horse. You might receive or download a program that appears to be a harmless business utility or game. More likely, the Trojan horse is just a script attached to a benign-looking email. When you run the program or open the attachment, it does something else other than or in addition to what you thought it would. It might

- Download harmful software from a website.
- Install a key logger or other spyware on your machine.
- Delete files.
- Open a backdoor for a hacker to use.

It is common to find combination virus plus Trojan horse attacks. In those scenarios, the Trojan horse spreads like a virus. The MyDoom virus opened a port on your machine that a later virus, doomjuice, would exploit, thus making MyDoom a combination virus and Trojan horse.

A Trojan horse could also be crafted especially for an individual. If a hacker wished to spy on a certain individual, such as the company accountant, he could craft a program specifically to attract that person's attention. For example, if he knew the accountant was an avid golfer, he could write a program that computed handicap and listed best golf courses. He would post that program on a free web server. He would then email a number of people, including the accountant, telling them about the free software. The software, once installed, could check the name of the currently logged-on person. If the logged-on name matched the accountant's name, the software could then go out, unknown to the user, and download a key logger or other monitoring application. If the software

did not damage files or replicate itself, then it would probably go undetected for quite a long time. There have been a number of Trojan horses through the years. One of the earliest and most widely known was Back Orifice.

Such a program could be within the skill set of virtually any moderately competent programmer. This is one reason that many organizations have rules against downloading *any* software onto company machines. I am unaware of any actual incident of a Trojan horse being custom tailored in this fashion. However, it is important to remember that those creating virus attacks tend to be innovative people.

It is also important to note that creating a Trojan horse does not require programming skill. There are free tools on the Internet, such as EliteWrapper, that allow someone to combine two programs, one hidden and one not. So one could easily take a virus and combine it with, for example, a poker game. The end user would only see the poker game, but when it was run it would launch the virus.

Another scenario to consider is one that would be quite devastating. Without divulging programming details, the basic premise will be outlined here to illustrate the grave dangers of Trojan horses. Imagine a small application that displays a series of unflattering pictures of Osama Bin Laden. This application would probably be popular with many people in the United States, particularly people in the military, intelligence community, or defense-related industries. Now assume that this application simply sits dormant on the machine for a period of time. It need not replicate like a virus because the computer user will probably send it to many of his associates. On a certain date and time, the software connects to any drive it can, including network drives, and begins deleting all files. If such a Trojan horse were released "in the wild," within 30 days it would probably be shipped to thousands, perhaps millions, of people. Imagine the devastation when thousands of computers begin deleting files and folders.

This scenario is mentioned precisely to frighten you a little. Computer users, including professionals who should know better, routinely download all sorts of things from the Internet, such as amusing flash videos and cute games. Every time an employee downloads something of this nature, there is the chance of downloading a Trojan horse. One need not be a statistician to realize that if employees continue that practice long enough, they will eventually download a Trojan horse onto a company machine. If they do, hopefully the virus will not be as vicious as the theoretical one just outlined here.

Because Trojan horses are usually installed by users themselves, the security countermeasure for this attack is to prevent downloads and installations by end users. From a law enforcement perspective, the investigation of a crime involving a Trojan horse would involve a forensic scan of the computer hard drive, looking for the Trojan horse itself.

Other Forms of Malware

This chapter discusses the most prominent forms of malware. Many other forms of attack exist, however. It is beyond the scope of this book to explore each of these, but you should be aware of the existence of these other forms of malware. Simply being aware can go a long way toward enabling you to defend your system efficiently. This section will touch upon just a few other forms of malware.

Rootkit

A *rootkit* is software used to obtain administrator-level access to a computer or computer network. The intruder installs a rootkit on a computer after first obtaining user-level access, either by exploiting a known vulnerability or cracking a password. Or the rootkit can be delivered in the same manner as any virus. The rootkit then collects user IDs and passwords to other machines on the network, thus giving the hacker root or privileged access.

A rootkit may consist of utilities that also:

- Monitor traffic and keystrokes
- Create a backdoor into the system for the hacker's use
- Alter log files
- Attack other machines on the network
- Alter existing system tools to circumvent detection

The presence of a rootkit on a network was first documented in the early 1990s. At that time, Sun and Linux operating systems were the primary targets for a hacker looking to install a rootkit. Today, rootkits are available for a number of operating systems and are increasingly difficult to detect on any network.

Malicious Web-Based Code

A *malicious web-based code*, also known as a *web-based mobile code*, simply refers to a code that is portable to all operating systems or platforms such as HTTP, Java, and so on. The "malicious" part implies that is it a virus, worm, Trojan horse, or some other form of malware. Simply put, the malicious code does not care what the operating system may be or what browser is in use. It infects them all blindly.

Where do these codes come from, and how are they spread? The first generation of the Internet was mostly indexed text files. However, as the Internet has grown into a graphical, multimedia user experience, programmers have created scripting languages and new application technologies to enable a more interactive experience. As with any new technology, programs written with scripting languages run the gamut from useful to poorly crafted to outright dangerous.

Technologies such as Java and ActiveX enable these buggy or untrustworthy programs to move to and execute on user workstations. (Other technologies that can enable malicious code are executables, JavaScript, Visual Basic Script, and plug-ins.) The Web acts to increase the mobility of code without differentiating between program quality, integrity, or reliability. Using available tools, it is quite simple to "drag and drop" code into documents that are subsequently placed on web servers and made available to employees throughout the organization or individuals across the Internet. If this code is maliciously programmed or just improperly tested, it can cause serious damage.

Not surprisingly, hackers have used these very useful tools to steal, alter, and erase data files as well as gain unauthorized access to corporate networks. A malicious code attack can penetrate corporate networks and systems from a variety of access points, including websites, HTML content in email messages, or corporate intranets.

Today, with billions of Internet users, new malicious code attacks can spread almost instantly through corporations. The majority of damage caused by malicious code happens in the first hours after a first-strike attack occurs—before there is time for countermeasures. The costs of network downtime or theft of intellectual property (IP) make malicious code a top priority.

Logic Bombs

A *logic bomb* is a type of malware that executes its malicious purpose when a specific criteria is met. The most common factor is date/time. For example, a logic bomb might delete files on a certain date/time. An example is the case of Roger Duronio. In June 2006, Roger Duronio, a system administrator for UBS, was charged with using a logic bomb to damage the company's computer network. His plan was to drive the company stock down due to damage from the logic bomb, so he was charged with securities fraud. Duronio was later convicted and sentenced to 8 years and 1 month in prison and ordered to pay $3.1 million restitution to UBS.

Another example occurred at the mortgage company Fannie Mae. On October 29, 2008, a logic bomb was discovered in the company's systems. This logic bomb had been planted by a former contractor, Rajendrasinh Makwana, who had been terminated. The bomb was set to activate on January 31, 2009 and completely wipe all of the company's servers. Makwana was indicted in a Maryland court on January 27, 2009 for unauthorized computer access. On December 17, 2010 he was convicted and sentenced to 41 months in prison, followed by 3 years of probation after release. What is most interesting about this case is that Makwana planted the logic bomb between the time he was terminated and the time the network administrators cancelled his network access.

This illustrates the importance of ensuring that the accounts of former employees are deactivated immediately when their employment is terminated. That applies whether it is an involuntary termination, retirement, or voluntary quit.

Creating Malware

This section examines a number of methods that will allow you to create various types of malware. It should be noted that this should be done with some care. Our goal, as penetration testers, is not to wreak havoc on the target network—quite the opposite. There is a twofold reason for learning basic malware writing skills. The first is so that you will better understand the threat that malware poses. If you can write a simple virus yourself, then you know how easy it is to do, and you understand why malware is so prevalent. A second reason is that in some cases, innocuous malware can be useful as part of a penetration test. As we examine various examples, I will point out scenarios to use malware as part of a pen test, and scenarios which should never be used.

In later chapters, we will be spending significant time exploring Metasploit. That will also give you the opportunity to embed exploits into PDFs, executables, and other files, thus effectively creating malware.

Levels of Malware Writing Skill

As you can imagine, simply reading a single chapter in a single book will not make you a master of malware creation. To truly master writing malware, you will need significant programming expertise; however, in this chapter we will explore some basic techniques. But first let us define the skill levels of malware creators.

1. The lowest skill level are those using GUI tools. There are, in fact, a wide range of tools you can use to create simple malware. Anyone can download these tools from the Internet and in a few short minutes have their own malware created. This requires no programming skills at all. However, it should be noted that malware created in this fashion has a significant chance of being detected by antivirus software.
2. The next level up are those using batch files and simple scripts. These require rudimentary script writing skills, very basic programming. They may or may not be detected by antivirus software. They are often useful in penetration tests.
3. The third level up are those who take existing malware code and modify it. There exist a number of online repositories where one can download the source code for well-known viruses such as *Melissa* and *I Love You*. Then you only need to make a few modifications to create your own virus.
4. The fourth level involves more substantial programming, because you create the entire virus from scratch. Such a new virus is far less likely to be detected by antivirus software, at least at first.
5. The ultimate level is to create malware that is both stealthy and self-destructs when no longer needed. Obviously, this requires sophisticated programming skills and is far less common.

The preceding skill list hierarchy is my own creation, but one I find quite useful. It is important to note that the skill level of the malware writer is proportional to the sophistication of the malware produced.

GUI Tools

There are a number of tools that have a graphical user interface (GUI) that allow one to quickly and easily create a basic virus. It should be noted that these tools usually produce viruses that are not subtle, and are likely to be detected by antivirus software. One of the best known such tools is TeraBIT Virus Maker, shown in Figure 5-5.

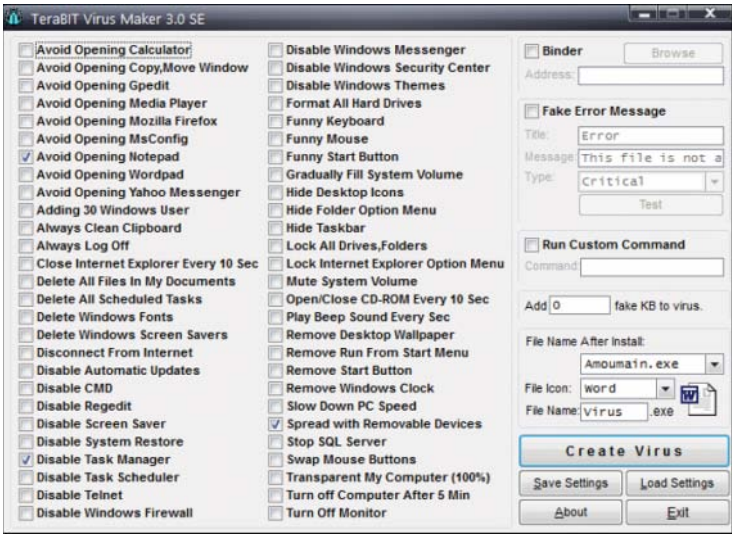

FIGURE 5-5 TeraBIT Virus Maker.

Another interesting GUI virus maker is Virus Maker from BlackHost http://www.blackhost.xyz, as shown in Figure 5-6. There are several interesting things about this tool. In addition to the normal things (like changing mouse behavior), it can open a website. This makes it useful for penetration testing. You can have it simply open a website that describes why one should be careful with attachments.