Offensive Security

Penetration Test Report for

OSCP Exam

Your Name

your.email@domain.tld

OSID: OS-XXXXX



October 4, 2026

©

All rights reserved to Offensive Security, 2020

No part of this publication, in whole or in part, may be reproduced, copied, transferred or any other right reserved to its copyright owner, including photocopying and all other copying, any transfer or transmission using any network or other means of communication, any broadcast for distant learning, in any form or by any means such as any information storage, transmission or retrieval system, without prior written permission from Offensive Security.

Contents

1 Overview **3**

1.1 Introduction 3

1.2 High-Level Summary 3

1.3 Technical Summary 3

2 example **5**

2.1 Service Enumeration 5

2.2 Remote Access Exploitation 5

2.2.1 Vulnerability Discussion 5

2.2.2 Recommendations 5

2.2.3 Proof of Concept 5

2.3 Privilege Escalation 7

2.3.1 Vulnerability Discussion 7

2.3.2 Recommendations 7

2.3.3 Proof of Concept 8

A Appendix **9**

A.1 Changes Made to Dirty Cow Exploit 9

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.

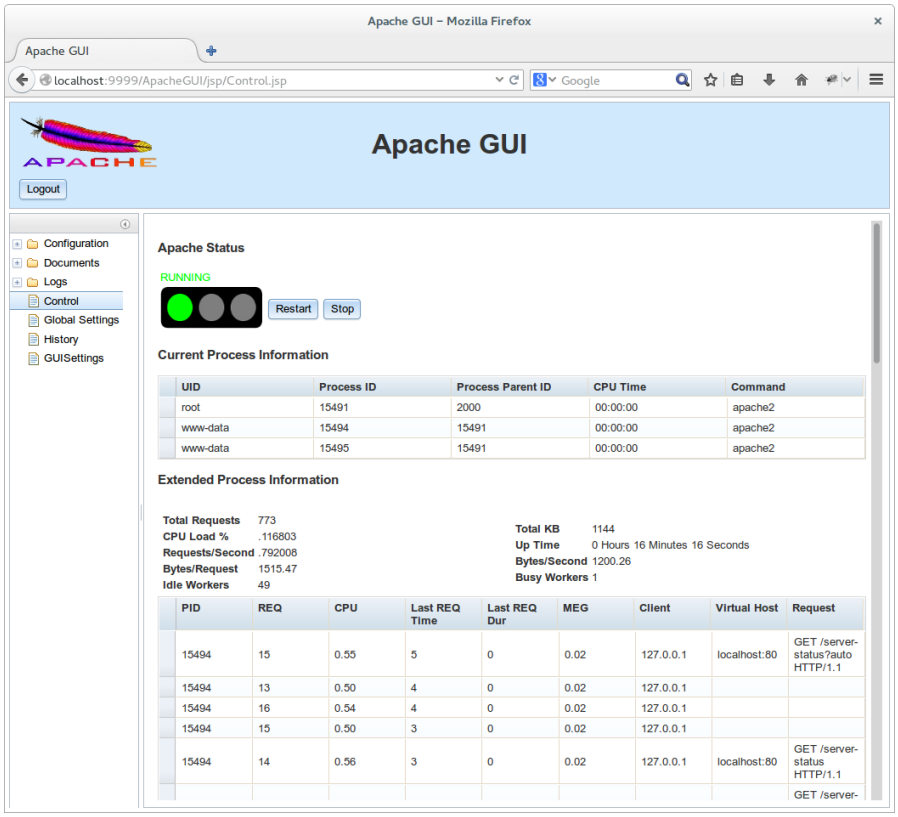


Figure 1: Management interface. Note: Image sourced from `http://www.apachegui.net/images/Control.png`.

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.

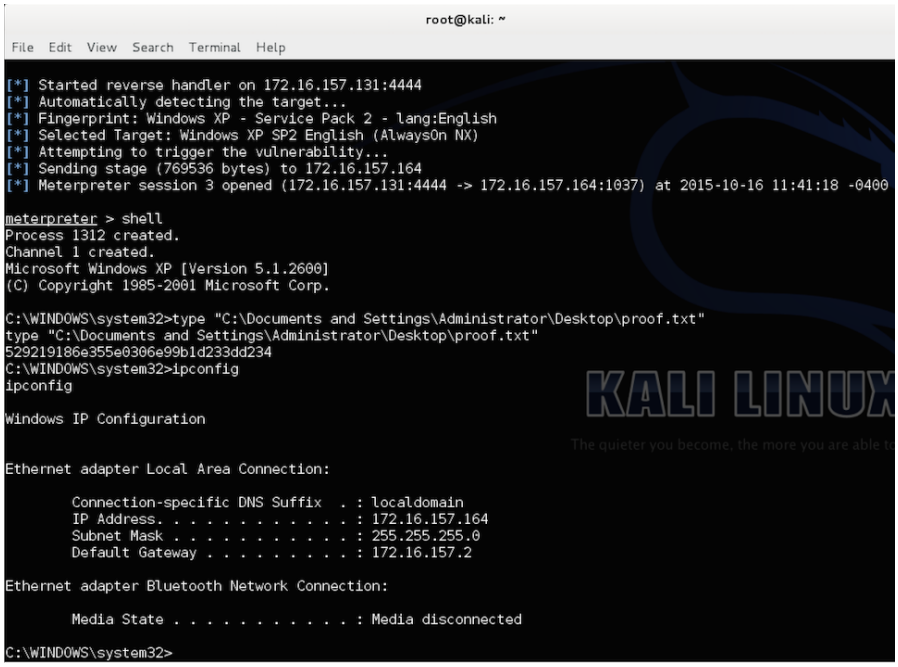


Figure 2: Proof of succesful low-privilege remote access to *example (XX.XX.XX.XX)*. Note: Image sourced from `https://support.offensive-security.com/oscp-exam-guide/`. Ensure `type` or `cat` are used to print the flag and `ipconfig` or its counterparts to display the machine's address.

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.

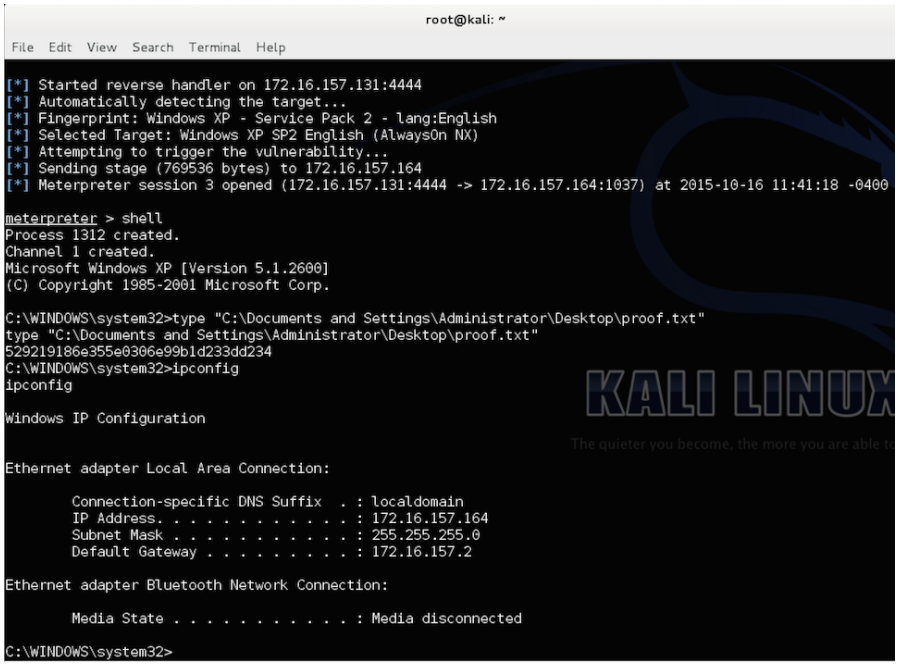


Figure 3: Proof of successful *root* access to *example (XX.XX.XX.XX)*.
Note: Image sourced from `https://support.offensive-security.com/oscp-exam-guide/`. Ensure `type` or `cat` are used to print the flag and `ipconfig` or its counterparts to display the machine's address.

[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/`