

Penetration Test Report

Example of a complete penetration test report

Marmeus

01-12-2022



Contents

1	Offensive Security Exam Penetration Test Report	1
1.1	Introduction	1
1.2	Objective	1
1.3	Requirements	1
2	High-Level Summary	2
2.1	Recommendations	2
3	Methodologies	3
3.1	Information Gathering	3
3.2	Report - Service Enumeration	3
3.2.1	IP-1	3
3.2.2	IP-2	4
3.2.3	IP-3	4
3.2.4	IP-4	4
3.2.5	IP-5	4
3.3	Report - Penetration	4
3.4	Report – House Cleaning	5
4	Annexe	6
4.1	Annexe 1 - Local and Proof contents	6
4.2	Anex 2 - Metasploit/Meterpreter	6

List of Figures

3.1 Local File Inclusion	5
------------------------------------	---

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 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 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

2 High-Level Summary

I was tasked with performing an internal penetration test towards the 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. 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:

- IP-1
- IP-2
- IP-3
- IP-4
- IP-5

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.

3 Methodologies

I utilised a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environment is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and include 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

- IP-1
- IP-2
- IP-3
- IP-4
- IP-5

3.2 Report - 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 to an attacker as it provides detailed information on potential attack vectors in a system. Understanding what applications are running on the system provides an attacker with vital information before conducting the actual penetration test. In some cases, some ports may not be listed.

3.2.1 IP-1

Server IP Address	Ports Open	Service Banner
192.168.31.218	TCP: 80, 3389 **UDP**:	Apache / RDP

3.2.2 IP-2

3.2.3 IP-3

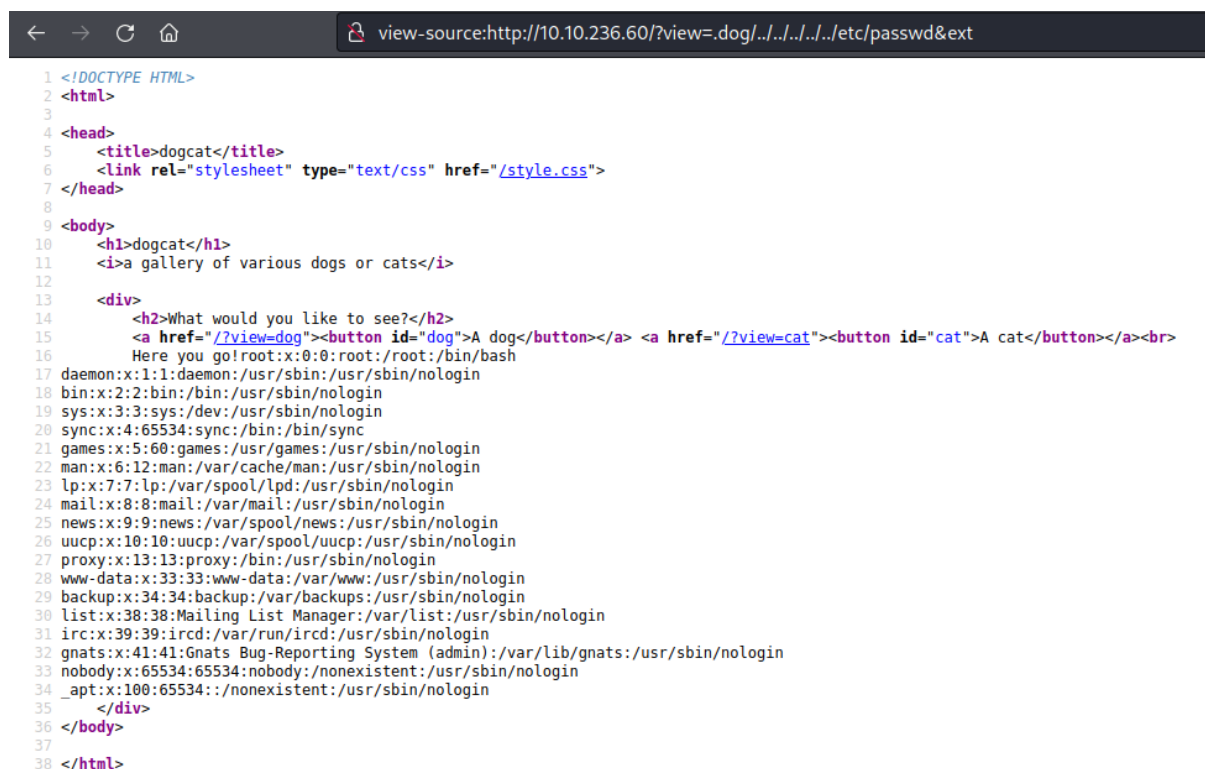
3.2.4 IP-4

3.2.5 IP-5

3.3 Report - Penetration

The penetration testing portion of the assessment focuses heavily on gaining access to a variety of systems. During this penetration test, OS-XXXXX was able to successfully gain access to 10 out of the 50 systems

- **Vulnerability Exploited:** KikChat - (LFI/RCE) Multiple Vulnerability
- **System Vulnerable:** 192.168.31.218
- **Vulnerability Explanation:** The “KikChat” web application suffers from a Local File Include (LFI), as well as a Remote Code Execution (RCE) vulnerability. A combination of these vulnerabilities was used to obtain a low privilege shell.
- **Privilege Escalation Vulnerability:** Named Pipe Impersonation (In Memory/Admin)
- **Severity:** Critical High Medium Low
- **Proof Of Concept:**



```
1 <!DOCTYPE HTML>
2 <html>
3
4 <head>
5   <title>dogcat</title>
6   <link rel="stylesheet" type="text/css" href="/style.css">
7 </head>
8
9 <body>
10  <h1>dogcat</h1>
11  <i>a gallery of various dogs or cats</i>
12
13  <div>
14    <h2>What would you like to see?</h2>
15    <a href="/?view=dog"><button id="dog">A dog</button></a> <a href="/?view=cat"><button id="cat">A cat</button></a><br>
16    Here you go!root:x:0:0:root:/root:/bin/bash
17 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
18 bin:x:2:2:bin:/bin:/usr/sbin/nologin
19 sys:x:3:3:sys:/dev:/usr/sbin/nologin
20 sync:x:4:65534:sync:/bin:/bin/sync
21 games:x:5:60:games:/usr/games:/usr/sbin/nologin
22 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
23 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
24 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
25 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
26 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
27 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
28 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
29 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
30 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
31 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
32 gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
33 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
34 _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
35 </div>
36 </body>
37
38 </html>
```

Figure 3.1: Local File Inclusion

- **Confirming RCE (ScreenShot):**

- Linux: `hostname && whoami && ip address && cat proof.txt`
- Winodws (Powershell): `hostname ; whoami ; ipconfig ; type proof.txt`

- **Vulnerability Fix:** No known patch or update for this issue.

3.4 Report – House Cleaning

The house-cleaning portion of the assessment ensures that remnants of the penetration test are removed. Oftentimes, 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 leftover is of paramount importance. After the objectives on both the lab network and exam network were successfully completed, OS-XXXXX 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 any of the systems.

4 Annexe

4.1 Annexe 1 - Local and Proof contents

Host	Local.txt	Proof.txt

4.2 Anex 2 - Metasploit/Meterpreter

For the exam, I used my Metasploit/Meterpreter allowance on the following machine:

-