**OFFENSIVE SECURITY**

Penetration Test Report for   
**OSCP** Exam

Exam date

Student e-mail

OSID: XXXXX



Copyright © 2021 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**

[**1.0 Offensive Security Exam Penetration Test Report** 4](#_Toc87472645)

[**1.1 Introduction** 4](#_Toc87472646)

[**1.2 Objective** 4](#_Toc87472647)

[**1.3 Requirements** 4](#_Toc87472648)

[**2.0 High-Level Summary** 5](#_Toc87472649)

[**3.0 Methodologies** 6](#_Toc87472650)

[**3.1 Information Gathering** 6](#_Toc87472651)

[**3.2 Report - Penetration** 7](#_Toc87472652)

[**3.3 System IP: x.x.x.x** 7](#_Toc87472653)

[**3.3.1 Service Enumeration** 7](#_Toc87472654)

[**3.3.2 Foothold** 9](#_Toc87472655)

[**3.3.3 Getting Reverse shell** 9](#_Toc87472656)

[**3.3.6 Getting Root: proof.txt** 10](#_Toc87472657)

[**3.3.7 Vulnerability Fix and Severity** 10](#_Toc87472658)

[**3.3.8 Proof of Concept Code and links** 10](#_Toc87472659)

[**3.4 System IP: x.x.x.x** 11](#_Toc87472660)

[**3.4.1 Service Enumeration** 11](#_Toc87472661)

[**3.4.2 Foothold** 13](#_Toc87472662)

[**3.4. Getting Reverse shell** 13](#_Toc87472663)

[**3.4.4 Getting User: local.txt** 13](#_Toc87472664)

[**3.4.5 Privilege Escalation** 13](#_Toc87472665)

[**3.4.6 Getting Root: proof.txt** 13](#_Toc87472666)

[**3.4.7 Vulnerability Fix and Severity** 13](#_Toc87472667)

[**3.4.8 Proof of Concept Code and links** 13](#_Toc87472668)

[**3.5 System IP: x.x.x.x** 14](#_Toc87472669)

[**3.5.1 Service Enumeration** 14](#_Toc87472670)

[**3.5.2 Foothold** 15](#_Toc87472671)

[**3.5.3 Getting Reverse shell** 15](#_Toc87472672)

[**3.5.4 Getting User: local.txt** 15](#_Toc87472673)

[**3.5.5 Privilege Escalation** 15](#_Toc87472674)

[**3.5.6 Getting Root: proof.txt** 15](#_Toc87472675)

[**3.5.7 Vulnerability Fix and Severity** 15](#_Toc87472676)

[**3.5.8 Proof of Concept Code and links** 16](#_Toc87472677)

[**3.6 System IP: x.x.x.x** 17](#_Toc87472678)

[**3.6.1 Service Enumeration** 17](#_Toc87472679)

[**3.6.2 Explanation of the development process** 18](#_Toc87472680)

[**1.6.2.1** **Set Up** 18](#_Toc87472681)

[**1.6.2.2** **Fuzzing** 18](#_Toc87472682)

[**1.6.2.3** **Replicating the crash and controlling the EIP register** 18](#_Toc87472683)

[**1.6.2.4** **Finding the bad characters** 19](#_Toc87472684)

[**1.6.2.5** **Finding a valid return address** 20](#_Toc87472685)

[**1.6.2.6** **Generating the shellcode** 20](#_Toc87472686)

[**1.6.2.7** **Testing the shellcode against the debugging machine** 20](#_Toc87472687)

[**3.6.3 Getting Reverse shell from the real target** 20](#_Toc87472688)

[**3.6.4 Getting the root: proof.txt** 20](#_Toc87472689)

[**3.6.5 Complete Buffer Overflow Code** 20](#_Toc87472690)

[**3.7 System IP: x.x.x.x** 21](#_Toc87472691)

[**3.7.1 Service enumeration** 21](#_Toc87472692)

[**3.8 Maintaining Access** 22](#_Toc87472693)

[**3.9 House Cleaning** 22](#_Toc87472694)

[**4.0 Additional Items** 23](#_Toc87472695)

[**Appendix 1 - Proof and Local Contents:** 23](#_Toc87472696)

[**Appendix 2 – Links and references** 24](#_Toc87472697)

**1.0 Offensive Security Exam Penetration Test Report**

**1.1 Introduction**

The Offensive Security Exam penetration test report contains all efforts that were conducted 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

**2.0 High-Level Summary**

I was tasked with performing an internal penetration test towards Offensive Security’s exam network. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, like those of an attacker and attempt to infiltrate Offensive Security’s internal exam systems – the THINC.local domain. 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 exam network. When performing the attacks, I was able to gain access to **X** out of five machines, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. The four systems were successfully exploited, and access granted. This is the list of all the tested systems:

* x.x.x.x
* x.x.x.x
* x.x.x.x
* x.x.x.x
* x.x.x.x

**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, they should be tested again to verify that the patches did not introduce new vulnerabilities. Moreover, they should remain on a regular patch program to protect additional vulnerabilities that are discovered later.

**2.2 Risk Exposure over Time**

The overall risk identified following the penetration test is very high. Due to the severity of some vulnerabilities, immediate action is desirable. It is reasonable to believe that an attacker would be able to successfully carry out an attack against all the tested systems.

**3.0 Methodologies**

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security’s exam environment is secured. Below is a breakout of how I was able to identify and exploit the variety of systems, it includes all the individual vulnerabilities found.

**3.1 Information Gathering**

The information gathering portion of a penetration test focuses on identifying the targets within the given scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

**Exam Network**

* x.x.x.x
* x.x.x.x
* x.x.x.x
* x.x.x.x
* x.x.x.x

**3.2 Report - Penetration**

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, OSID XXXX was able to successfully gain access to X out of the 5 systems.

**3.3 System IP: x.x.x.x**

**3.3.1 Service Enumeration**

|  |  |  |  |
| --- | --- | --- | --- |
| **Protocol** | **Port** | **Service** | **Version identified** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Nmap scans**

Below you can find the Nmap commands that I launched to identify the services and profiling the target, and their respective outputs.

**Nmap full TCP scan**

*Command used:*

/usr/bin/nmap

*Command output:*

|  |
| --- |
| *-snip-*  PORT STATE SERVICE REASON  *-snip-* |

**Nmap script scan and version scan on TCP ports found open**

*Command used:*

/usr/bin/nmap

*Command output:*

|  |
| --- |
| *-snip-*  PORT STATE SERVICE VERSION  *-snip-* |

**3.3.2 Foothold**

**Overview**

The initial foothold in this system was possible because of…..

**3.3.3 Getting Reverse shell**

To successfully abuse this vulnerability, changes to the exploit’s source code are needed.

|  |
| --- |
|  |

*Insert screenshot of the source code*

A snippet from the above screenshotted code with the highlighted changes.

|  |
| --- |
| *-snip-*  *-snip-* |

**3.3.6 Getting Root: proof.txt**

**3.3.7 Vulnerability Fix and Severity**

**3.3.8 Proof of Concept Code and links**

**3.4 System IP: x.x.x.x**

**3.4.1 Service Enumeration**

|  |  |  |  |
| --- | --- | --- | --- |
| **Protocol** | **Port** | **Service** | **Version identified** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Nmap scans**

Below you can find the Nmap commands that I launched to identify the services and profiling the target and their respective outputs.

**Nmap full TCP scan**

*Command used:*

/usr/bin/nmap

*Command output:*

|  |
| --- |
| *-snip-*  PORT STATE SERVICE REASON  *-snip-* |

**Nmap script scan and version scan on TCP ports found open**

*Command used:*

/usr/bin/nmap

*Command output:*

|  |
| --- |
| *-snip-*  PORT STATE SERVICE VERSION  *-snip-* |

**3.4.2 Foothold**

**3.4. Getting Reverse shell**

**3.4.4 Getting User: local.txt**

*Insert the local.txt proof screenshot.*

**3.4.5 Privilege Escalation**

**3.4.6 Getting Root: proof.txt**

*Insert the proof.txt proof screenshot.*

**3.4.7 Vulnerability Fix and Severity**

**Foothold vulnerability**

**Privilege escalation vulnerability**

**3.4.8 Proof of Concept Code and links**

**3.5 System IP: x.x.x.x**

**3.5.1 Service Enumeration**

|  |  |  |  |
| --- | --- | --- | --- |
| **Protocol** | **Port** | **Service** | **Version identified** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Nmap scans**

Below you can find the Nmap commands that I launched to identify the services and profiling the target and their respective outputs.

**Nmap full TCP scan**

*Command used:*

/usr/bin/nmap

*Command output:*

|  |
| --- |
| *-snip-*  PORT STATE SERVICE REASON  *-snip-* |

**Nmap script scan and version on TCP ports found open**

*Command used:*

/usr/bin/nmap

*Command output:*

|  |
| --- |
| *-snip-*  PORT STATE SERVICE VERSION  *-snip-* |

**3.5.2 Foothold**

**Exploitation**

**3.5.3 Getting Reverse shell**

**Post exploitation reconnaissance**

**3.5.4 Getting User: local.txt**

**3.5.5 Privilege Escalation**

**3.5.6 Getting Root: proof.txt**

**3.5.7 Vulnerability Fix and Severity**

**Foothold vulnerability**

**Privilege escalation vulnerability**

**3.5.8 Proof of Concept Code and links**

**3.6 System IP: x.x.x.x**

**3.6.1 Service Enumeration**

|  |  |  |  |
| --- | --- | --- | --- |
| **Protocol** | **Port** | **Service** | **Version identified** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Nmap scans**

Below you can find the Nmap commands that I have launched to identify the services and profiling the target and their respective outputs.

**Nmap full TCP scan**

*Command used:*

/usr/bin/nmap

*Command output:*

|  |
| --- |
| *-snip-*  PORT STATE SERVICE REASON  *-snip-* |

**Identifying vulnerable application**

**3.6.2 Explanation of the development process**

* + - 1. **Set Up**
      2. **Fuzzing**

|  |
| --- |
| *Insert here the fuzzer source code* |

* + - 1. **Replicating the crash and controlling the EIP register**

|  |
| --- |
|  |

|  |
| --- |
| Insert here the exploit source code |

|  |
| --- |
|  |

* + - 1. **Finding the bad characters**

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

* + - 1. **Finding a valid return address**
      2. **Generating the shellcode**
      3. **Testing the shellcode against the debugging machine**

**3.6.3 Getting Reverse shell from the real target**

**3.6.4 Getting the root: proof.txt**

**3.6.5 Complete Buffer Overflow Code**

|  |
| --- |
|  |

**3.7 System IP: x.x.x.x**

**3.7.1 Service enumeration**

|  |  |  |  |
| --- | --- | --- | --- |
| **Protocol** | **Port** | **Service** | **Version identified** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Nmap scans**

Below you can find the Nmap commands that I launched to identify the services and profiling the target and their respective outputs.

**Nmap full TCP scan**

*Command used:*

/usr/bin/nmap

*Command output:*

|  |
| --- |
| *-snip-*  PORT STATE SERVICE REASON  *-snip-* |

**Nmap script service and version scan**

*Command used:*

/usr/bin/nmap

*Command output:*

|  |
| --- |
| *-snip-*  PORT STATE SERVICE VERSION  *-snip-* |

**3.8 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.

**3.9 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.

**4.0 Additional Items**

**Appendix 1 - Proof and Local Contents:**

|  |  |  |
| --- | --- | --- |
| **IP** | **Local.txt content** | **Proof.txt content** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Appendix 2 – Links and references**

|  |  |  |
| --- | --- | --- |
| **Tool** | **Used for** | **Official link** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

In this section, I have put wherever possible the official links to the tools that I have used during my assessment to facilitate the technicians to better replicate the findings.

This Penetration Test report was written on *date*