Offensive Security

Penetration Test Report for Exam

email@address.com

OSID: OS-

Table of Contents

[1.0 - High-Level Summary 3](#_Toc40695114)

[1.1 – Recommendations 3](#_Toc40695115)

[2.0 – Methodologies 4](#_Toc40695116)

[OS-XXXXX utilized a widely adopted approach to performing penetration testing that is effective in 4](#_Toc40695117)

[testing how well the Offensive Security Labs and Exam environments are secure. Below is a summary of 4](#_Toc40695118)

[how OS-XXXXX was able to identify and exploit a number of systems. 4](#_Toc40695119)

[2.1 - Information Gathering 4](#_Toc40695120)

[2.2 - Service Enumeration 4](#_Toc40695121)

[2.3 – Penetration 4](#_Toc40695122)

[**System Vulnerable:** 0.0.0.0 (<Hostname>) 6](#_Toc40695123)

[2.4 – Report – House Cleaning 7](#_Toc40695124)

# 1.0 - High-Level Summary

OS-XXXXX was tasked with performing an internal penetration test in the Offensive Security Labs and

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 lab systems – the THINC.local domain, and the exam network. OSXXXXX’s

overall objective was to evaluate the network, identify systems, and exploit flaws while

reporting the findings back to Offensive Security.

While conducting the internal penetration test, there were several alarming vulnerabilities that were

identified within Offensive Security’s network. For example, OS-XXXXX was able to gain access to

multiple machines, primarily due to outdated patches and poor security configurations. During testing,

OS-XXXXX 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:

|  |  |
| --- | --- |
| Target | Access Description |
| 1 | Obtained a shell with System privileges by exploiting a stack buffer overflow vulnerability. |
| 2 | Uploaded PHP backdoor using outdated libraries on Nagios XI network monitor. Gained low level access via python reverse shell. Elevated privileges via misconfigured user ‘sudo’ permissions. |
| 3 | Discovered database admin credentials using directory traversal. Accessed database and changed WordPress admin password. Uploaded PHP webshell to WordPress and gained low level access to the server. Elevated privileges via misconfigured user permissions on a service running at the System level. |
| 4 | Used weak user credentials to gain access to October CMS. Upload a php reverse shell to gain low level access to the server. Elevated privileges via a vulnerable version of NfSen. |
| 5 | Unable to gain access due to time restraints |

## 1.1 – Recommendations

OS-XXXXX recommends patching the vulnerabilities identified during the penetration test 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 in order to

mitigate additional vulnerabilities that may be discovered at a later date.

# 2.0 – Methodologies

## OS-XXXXX utilized a widely adopted approach to performing penetration testing that is effective in

## testing how well the Offensive Security Labs and Exam environments are secure. Below is a summary of

## how OS-XXXXX was able to identify and exploit a number of systems.

## 2.1 - Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the

penetration test. During this penetration test, OS-XXXXX was tasked with exploiting the lab and exam

network. The specific IP addresses were:

**Exam Network**

0.0.0.0

0.0.0.0

0.0.0.0

0.0.0.0

0.0.0.0

## 2.2 - 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 into 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.

|  |  |
| --- | --- |
| Server IP Address | Ports Open - Service |
| 0.0.0.0 | **TCP:**  80 |
|  |  |
|  |  |
|  |  |
|  |  |

## 2.3 – 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.

Console text color legend:

Standard console text

Commands inputted by the penetration tester

Output we wish to highlight

{...} Abbreviated output for brevity

Note: The events in this penetration test did not necessarily occur in order as described in this report. Additionally, some output displayed in the console format above was reconstructed or modified (e.g. aliases replaced). Both of these changes have been made to increase the readability of this report.

|  |
| --- |
| **System Vulnerable:** 0.0.0.0 (<Hostname>) **Vulnerabilities Exploited:**  **Vulnerability Explanation:**  **Vulnerability Fix:**  **Severity:** **Critical**  **Proof of Concept:**  **Local.txt:**  **Privilege Escalation:**  **Proof.txt** |

## 2.4 – Report – House Cleaning

Clean up was conducted after assessment of each target to remove any artifacts from the penetration test. The removals included any user accounts created and any files uploaded to the targets. Additionally, any configuration changes made were reverted to their original state.