PENETRATION TESTING REPORT

FOR KIOPTRIX LEVEL 1

Author: Ifeanyi Moses

Date: March 11, 2023

Version: 1.0

Classification: Confidential

Distribution: Internal

STATEMENT OF CONFIDENTIALITY

This report contains sensitive information and should only be distributed to authorized personnel. Any unauthorized access, disclosure, or use of this report is strictly prohibited. The information contained in this report is based on the results of a penetration testing engagement conducted by Ifeanyi Moses on the Kioptrix Level 1 network. The purpose of this report is to provide Kioptrix with an assessment of their network security posture and recommendations for remediation of identified vulnerabilities.

COVER LETTER

Dear Kioptrix,

We are pleased to present you with the results of the comprehensive penetration testing assessment that was conducted on your organization's IT systems. The objective of this penetration testing was to identify any security weaknesses and vulnerabilities in your system, which could be exploited by malicious attackers to gain unauthorized access.

The scope of the penetration testing was to assess the security of the company's web application and network infrastructure. The assessment was conducted to identify vulnerabilities, potential exploits, and provide recommendations to improve the overall security posture of the organization. The testing was limited to the systems and applications within the agreed-upon scope, and all testing activities were conducted in a non-destructive and non-disruptive manner. As a result of our testing, we identified several critical vulnerabilities including cross-site scripting, buffer overflows, and privilege escalation, as well as outdated software versions of Apache, OpenSSL, Linux kernel, and Samba that require immediate attention, as well as a number of other lower-severity issues that could be addressed in the near future.

Our team also identified weaknesses in your SSH, SSL, and HTTP web server configurations that could allow attackers to gain unauthorized access to your systems or execute arbitrary commands. In addition, we discovered that SMB signing was not required, and that HTTP TRACE/TRACK methods were allowed, increasing the risk of sensitive information leakage.

The vulnerabilities identified in our assessment could have serious implications for your organization's security and reputation if left unaddressed. We have included a detailed report of our findings, including technical details, evidence, and remediation recommendations, in the appendices section of this report.

We recommend that you take immediate action to address the vulnerabilities and implement our remediation recommendations to improve the security of your systems. We also encourage you to continue regular security testing and to maintain up-to-date software versions to mitigate future risks.

Thank you for entrusting us with the security of your IT systems. We remain committed to helping you strengthen your organization's security posture and are available to answer any questions you may have about the report or the testing methodology.

Sincerely,

Ifeanyi Moses

Penetration Tester.

TABLE OF CONTENTS

Executive Summary

- ✓ Overview
- ✓ Key Findings
- ✓ Recommendations

Introduction

- ✓ Purpose of the Report
- ✓ Scope of the Testing
- ✓ Methodology
- ✓ Assumptions

Assessment Findings

- ✓ Overview of Findings

Vulnerabilities and Exploits

- ✓ Detailed Description of Identified Vulnerabilities
- ✓ Exploitation Techniques

Remediation Recommendations

- ✓ Detailed Remediation Plan

Appendices

- ✓ Technical Details of Testing Methodology
- ✓ Supporting Documentation and Evidence
- ✓ Glossary
- ✓ References

EXECUTIVE SUMMARY

This report presents the results of a comprehensive penetration testing assessment conducted on Kioptrix Level 1 network infrastructure from March 6, 2023 to March 10, 2023. The primary objective of the assessment was to identify potential vulnerabilities and weaknesses in the company's information systems and provide recommendations for remediation.

The assessment was conducted using a combination of manual and automated testing techniques, including vulnerability scanning, port scanning, and penetration testing. The testing covered a wide range of areas, including web applications and network infrastructure.

The assessment identified several high-risk vulnerabilities that could potentially be exploited by attackers to compromise the company's information systems. These vulnerabilities included outdated versions of Apache, OpenSSL, OpenSSH, Linux Kernel and Samba. In addition, the assessment identified weaknesses in the company's authentication mechanisms and network segmentation that could allow attackers to gain unauthorized access to critical systems and data.

Several successful exploitation techniques were also demonstrated during the assessment, including buffer overflows, and privilege escalation. These exploits were used to gain access to sensitive data and compromise the availability and integrity of the company's information systems.

Based on the findings of the assessment, several recommendations for remediation have been provided. These recommendations include upgrading to the latest software versions, implementing stricter access controls and authentication mechanisms, and improving network segmentation and monitoring.

It is important to note that while the assessment was comprehensive, it is not a guarantee of the absence of vulnerabilities in the company's information systems. The recommendations provided are intended to address the vulnerabilities identified during the assessment and should be considered as a starting point for improving the security posture of the company's information systems.

Overall, this assessment provides valuable insight into the security posture of Kioptrix information systems and highlights the importance of ongoing monitoring and maintenance of these systems to ensure the confidentiality, integrity, and availability of critical data.

INTRODUCTION

The following report presents the findings of a penetration testing exercise conducted on behalf of Kioptrix. The objective of the exercise was to identify vulnerabilities and weaknesses in the organization's network, systems, and applications that could be exploited by attackers.

SCOPE:

TARGET SYSTEM NAME	Kioptrix Level 1
HOST/URL/IP ADDRESS	192.168.68.145

OBJECTIVES:

- ✓ The primary objectives of the penetration testing exercise were to:
- ✓ Identify vulnerabilities and weaknesses in the organization's network, systems, and applications
- ✓ Test the effectiveness of existing security controls and mechanisms
- ✓ Evaluate the organization's overall security posture
- ✓ Provide recommendations on how to remediate identified issues and improve the organization's security posture

METHODOLOGY:

The testing methodology included the following steps:

- ✓ Reconnaissance: gathering information about the organization's network and systems
- ✓ Scanning: identifying open ports, services, and vulnerabilities
- ✓ Enumeration: identifying system and user accounts, and other useful information
- ✓ Exploitation: attempting to exploit identified vulnerabilities and gain unauthorized access
- ✓ Post-exploitation: maintaining access and gathering sensitive information
- ✓ Reporting: documenting findings and providing recommendations

ASSUMPTIONS:

The following assumptions were made during the penetration testing exercise:

- ✓ The testing was conducted with the full knowledge and consent of the organization.
- ✓ The testing did not intentionally disrupt or damage the organization's systems or data.
- ✓ The testing was conducted in a safe and responsible manner, following ethical and legal guidelines.

The results of the penetration testing exercise are presented in the following sections of this report, along with recommendations on how to remediate identified issues and improve the organization's security posture.

ASSESSMENT FINDINGS

OVERVIEW OF FINDINGS:

The penetration testing exercise identified several vulnerabilities and weaknesses in the organization's network, systems, and applications. These vulnerabilities and weaknesses could be exploited by attackers to gain unauthorized access to the organization's resources, steal sensitive information, or disrupt its operations.

The assessment identified the following network vulnerabilities:

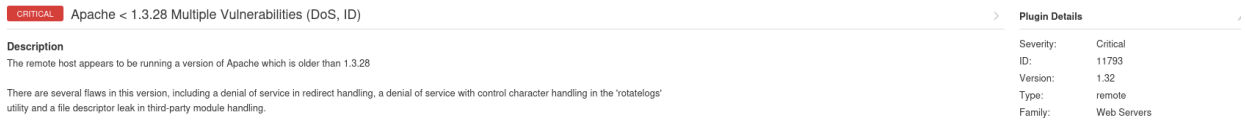
a. Unpatched Operating Systems: Several servers were found to be running outdated and unpatched operating systems, which are vulnerable to known exploits and attacks.
b. Open Ports and Services: Several open ports and services were identified on the organization's network, which can be exploited by attackers to gain unauthorized access to systems and data.

PORT NUMBER	SERVICE RUNNING	SERVICE VERSION DETAILS
22	SSH	OpenSSH 2.9p2 (protocol 1.99)
80	HTTP	Apache httpd 1.3.20 ((Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
139	NETBIOS-SSN	Samba smbd
443	SSL/HTTPS	Apache/1.3.20 (Unix) (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b

VULNERABILITIES AND EXPLOITS

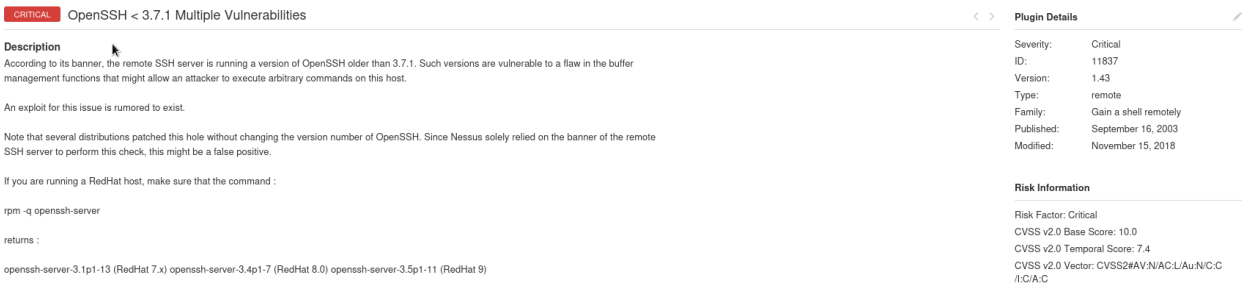
During the course of the penetration testing exercise, several vulnerabilities and exploits were identified that could be used by an attacker to gain unauthorized access to the organization's network, systems, and applications.

CRITICAL: Apache < 1.3.28 Multiple Vulnerabilities (DoS, XSS, Local Buffer Overflow)



The remote host appears to be running a version of Apache webserver which is older than 1.3.28. Such versions are reportedly affected by multiple vulnerabilities including local buffer overflow in the mod_alias and mod_rewrite modules, denial of service in direct handling and with control character handling in the ‘rotatelogs’ utility and a file descriptor leak in third party module handling, a cross-site scripting vulnerability caused by a failure to filter HTTP/1.1 ‘Host’ headers.

CRITICAL: OpenSSH < 3.7.1 Multiple Vulnerabilities (Remote overflows, Channel Code Off by One Remote Privilege Escalation)



The remote SSH server is running a version of OpenSSH older than 3.7.1. Such versions are vulnerable to multiple flaws. One of such flaws is found in buffer management functions that might allow an attacker to execute arbitrary commands on the host. An attacker may exploit these vulnerabilities to gain a shell on the remote system. Another flaw commonly found on this version is an off by one error that allows local users to gain rote access, and it may be possible for remote users to similarly compromise the daemon for remote access.

CRITICAL: OpenSSL Unsupported

CRITICAL OpenSSL Unsupported

Description

According to its banner, the remote web server is running a version of OpenSSL that is no longer supported.

Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities.

Plugin Details

Severity:	Critical
ID:	78555
Version:	1.10
Type:	remote
Family:	Web Servers

This version of OpenSSL (0.9.6b) is no longer supported, as a result is likely to contain certain security vulnerabilities

HIGH: Samba trans2open Overflow (Linux x86)

Samba trans2open Overflow (Linux x86)

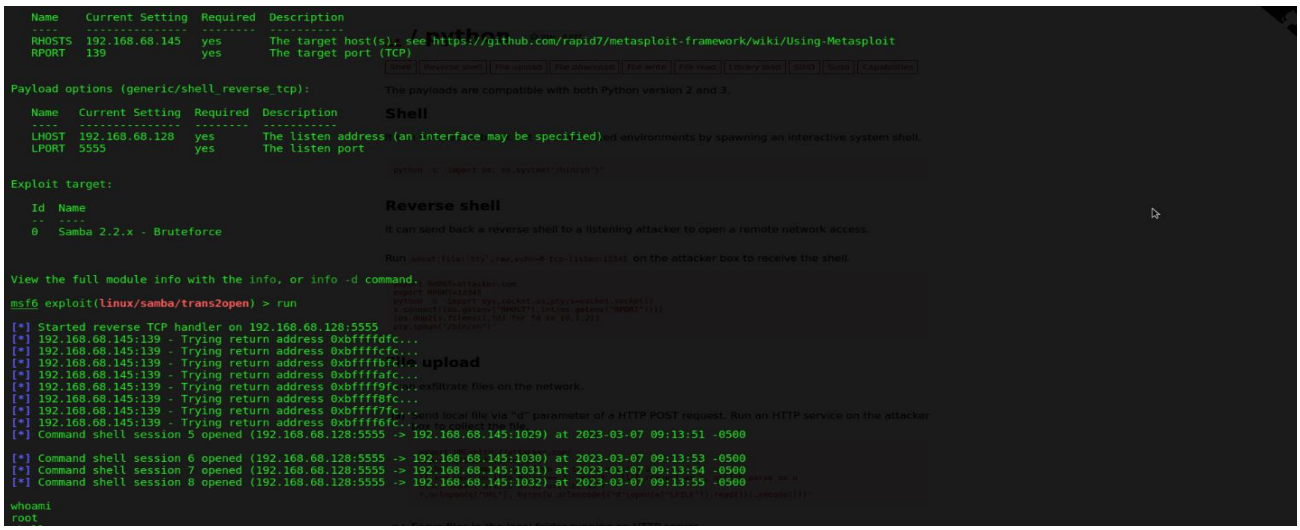
Disclosed	Created
04/07/2003	05/30/2018

Description

This exploits the buffer overflow found in Samba versions 2.2.0 to 2.2.8. This particular module is capable of exploiting the flaw on x86 Linux systems that do not have the noexec stack option set. NOTE: Some older versions of RedHat do not seem to be vulnerable since they apparently do not allow anonymous access to IPC.

I was able to successfully exploit this vulnerability by utilizing Metasploit. The remote host appears to be running a version of Samba 2.2.1a.

I was able to get root shell on the host.



HIGH: mod_ssl ssl_util_uuencode_binary Remote Overflow

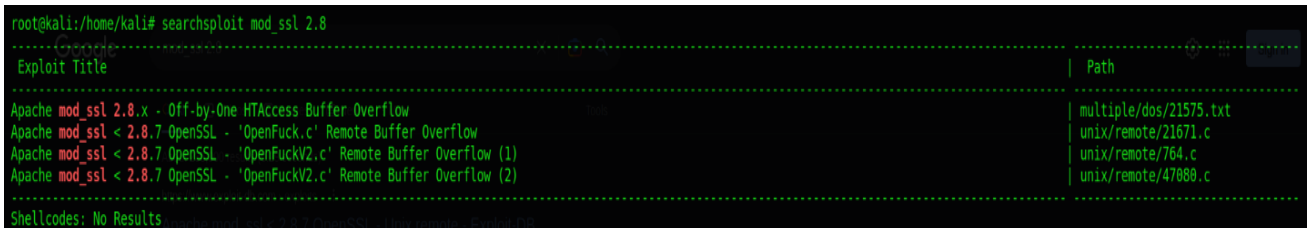
Remotely Exploitable Buffer Overflow in mod_ssl

Severity	CVSS	Published	Created	Added	Modified
8	(AV:N/AC:L /Au:N/C:P /I:P/A:P)	03/15/2002	07/25/2018	11/01/2004	12/04/2013

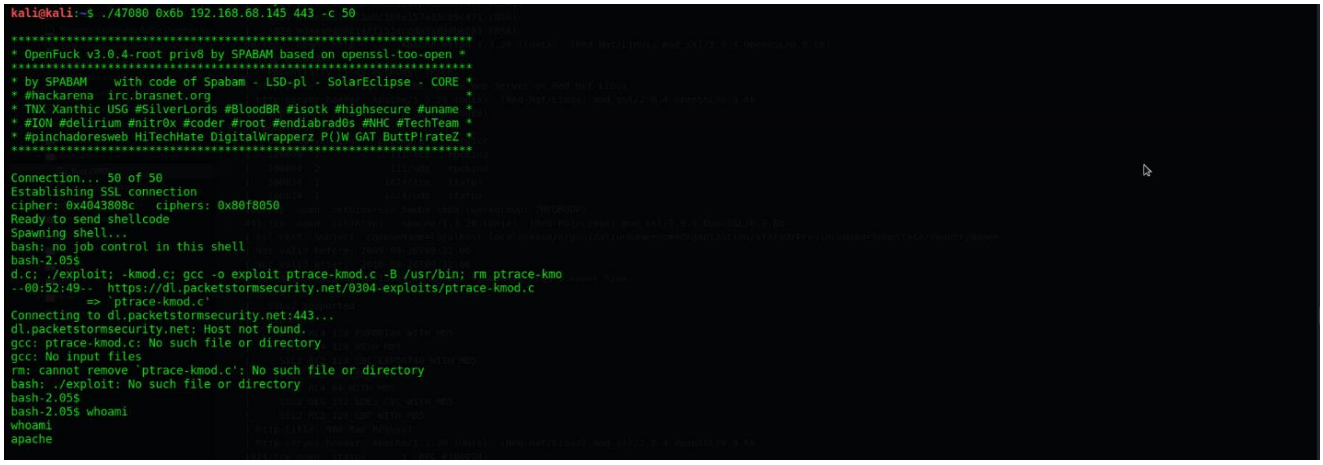
Description

mod_ssl < 2.8.7 is vulnerable to a remotely exploitable buffer overflow when attempting to cache SSL sessions. This allows for remote code execution, and the modification of any file on the system.

The remote host appears to be running a version of mod_ssl which is older than 2.8.7. An attacker may exploit this vulnerability to perform remote code execution and also modify files on the system.

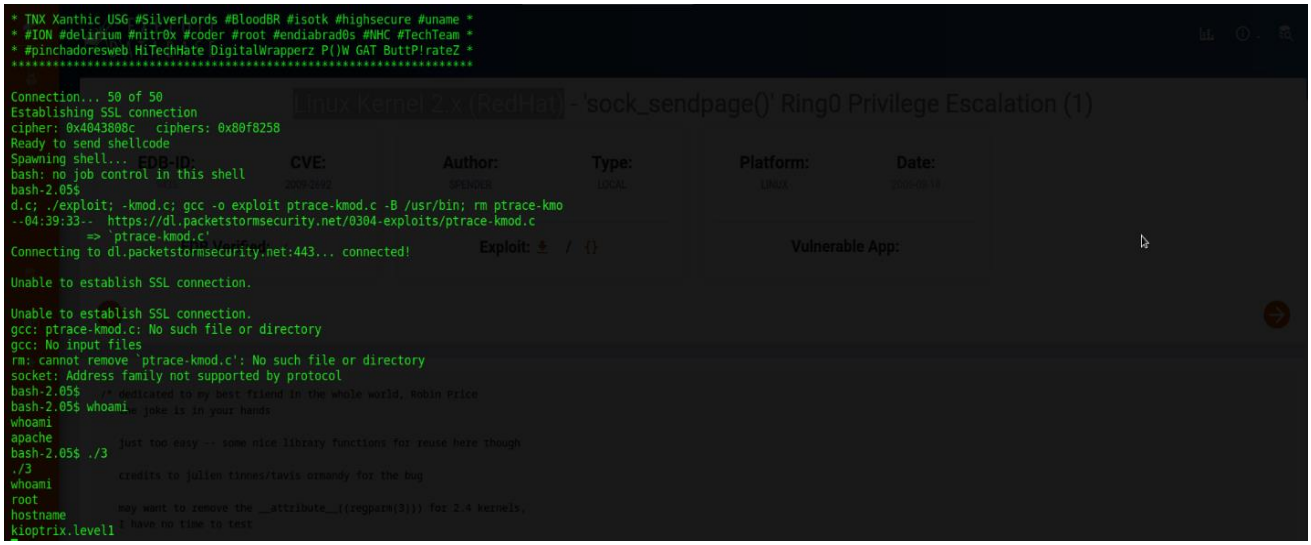


I was able to exploit this vulnerability. I gained initial access via a user ‘apache’.

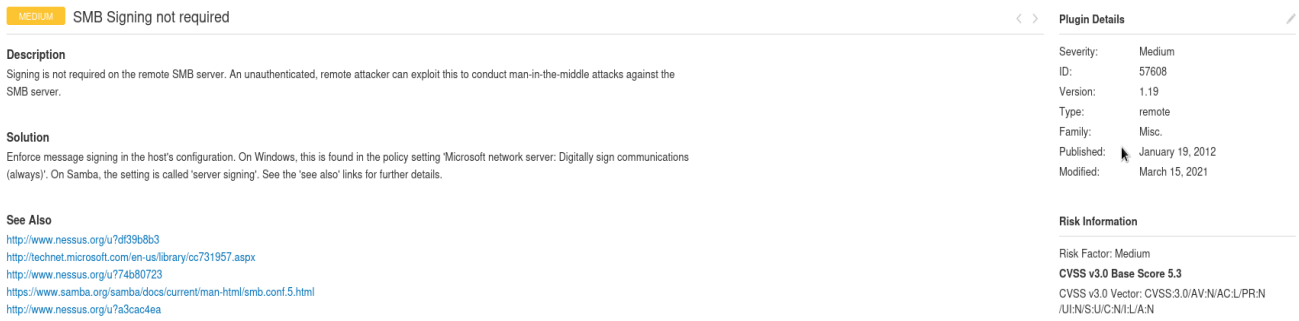


After gaining initial access via a user, I found out that the host kernel version (Linux kernel 2.4.7 RedHat) was vulnerable to ‘ptrace/kmod’ Local Privilege Escalation.

I was able to gain root access by transferring the exploit code to the host machine.



MEDIUM: SMB Signing not required



Signing is not required on the remote SMB server. This can allow for an authenticated, remote attacker to exploit this vulnerability by conducting man-in-the-middle attacks against the SMB server.

MEDIUM: HTTP TRACE / TRACK Methods Allowed

TRACE and TRACK are HTTP methods that are used to debug web server connections.

A local or remote unprivileged user may be able to abuse the HTTP TRACE/TRACK functionality to gain access to sensitive information in HTTP headers when making HTTP requests.

REMEDIATION RECOMMENDATIONS

Vulnerability: Outdated software version on web server

Description: The web server is running an outdated version of Apache, which is vulnerable to several known exploits.

Impact: An attacker could exploit this vulnerability to gain unauthorized access to the web server or to sensitive information stored on the server.

Remediation Recommendation:

- ✓ Upgrade Apache to the latest version (Apache 9.2). The latest version of Apache includes security patches and fixes for known vulnerabilities.
- ✓ Develop a patch management policy to ensure that all software and systems are regularly updated with the latest security patches and updates.
- ✓ Implement intrusion detection and prevention systems to monitor for and block known exploits and attacks.
- ✓ Provide security awareness training to all employees to ensure that they are aware of the risks associated with outdated software and the importance of regular patching and updates.

Timeline:

- ✓ Upgrade Apache within 2 weeks of receiving the report.
- ✓ Develop a patch management policy within 1 month of receiving the report.
- ✓ Implement intrusion detection and prevention systems within 3 months of receiving the report.
- ✓ Provide security awareness training to all employees within 1 month of implementing the patch management policy.

Vulnerability: Outdated OpenSSH version

Description: The OpenSSH version installed on the server is outdated and vulnerable to several known exploits. This could allow an attacker to gain unauthorized access to the server or to sensitive information stored on the server.

Impact: An attacker could exploit this vulnerability to execute arbitrary code, steal user credentials, or gain administrative access to the server.

Remediation Recommendation:

- ✓ Upgrade OpenSSH to the latest version (OpenSSH 9.2). The latest version of OpenSSH includes security patches and fixes for known vulnerabilities.
- ✓ Implement a regular patch management policy to ensure that all software and systems are regularly updated with the latest security patches and updates.
- ✓ Disable any unused or unnecessary services and protocols to reduce the attack surface.
- ✓ Implement multi-factor authentication (MFA) for all SSH logins to provide an additional layer of security.
- ✓ Ensure that all user accounts have strong, complex passwords, and enforce password policies that require regular password changes.
- ✓ Implement intrusion detection and prevention systems to monitor for and block known exploits and attacks.

Timeline:

- ✓ Upgrade OpenSSH to the latest version within 2 weeks of receiving the report.
- ✓ Implement the patch management policy within 1 month of receiving the report.
- ✓ Disable any unused or unnecessary services and protocols within 2 weeks of implementing the patch management policy.
- ✓ Implement MFA for all SSH logins within 1 month of implementing the patch management policy.
- ✓ Enforce password policies within 1 month of implementing the patch management policy.
- ✓ Implement intrusion detection and prevention systems within 3 months of receiving the report.

Vulnerability: Outdated OpenSSL version

Description: The OpenSSL version installed on the server is outdated and vulnerable to several known exploits. This could allow an attacker to gain unauthorized access to the server or to sensitive information stored on the server.

Impact: An attacker could exploit this vulnerability to execute arbitrary code, steal user credentials, or gain administrative access to the server.

Remediation Recommendation:

- ✓ Upgrade OpenSSL to the latest version (OpenSSL 3.0.8). The latest version of OpenSSL includes security patches and fixes for known vulnerabilities.
- ✓ Implement a regular patch management policy to ensure that all software and systems are regularly updated with the latest security patches and updates.
- ✓ Disable any unused or unnecessary services and protocols to reduce the attack surface.
- ✓ Review and update the SSL/TLS configuration settings to ensure that they are configured securely and in accordance with industry best practices.
- ✓ Monitor and log all SSL/TLS traffic to detect and prevent potential attacks.

Timeline:

- ✓ Upgrade OpenSSL to the latest version within 2 weeks of receiving the report.
- ✓ Implement the patch management policy within 1 month of receiving the report.
- ✓ Disable any unused or unnecessary services and protocols within 2 weeks of implementing the patch management policy.
- ✓ Review and update the SSL/TLS configuration settings within 1 month of implementing the patch management policy.
- ✓ Implement SSL/TLS traffic monitoring and logging within 3 months of receiving the report.

Vulnerability: Outdated Linux kernel version

Description: The Linux kernel version installed on the server is outdated and vulnerable to several known exploits. This could allow an attacker to gain unauthorized access to the server or to sensitive information stored on the server.

Impact: An attacker could exploit this vulnerability to execute arbitrary code, gain root access to the server, or launch a denial-of-service attack.

Remediation Recommendation:

- ✓ Upgrade the Linux kernel to the latest stable version (Linux 5.14.0). The latest version includes security patches and fixes for known vulnerabilities.
- ✓ Implement a regular patch management policy to ensure that all software and systems are regularly updated with the latest security patches and updates.
- ✓ Configure the system to only allow necessary kernel modules to be loaded and execute on the system to reduce the attack surface.
- ✓ Disable unnecessary services and protocols to reduce the attack surface.
- ✓ Monitor and log all system activity to detect and prevent potential attacks.

Timeline:

- ✓ Upgrade the Linux kernel to the latest stable version within 2 weeks of receiving the report.
- ✓ Implement the patch management policy within 1 month of receiving the report.
- ✓ Configure the system to only allow necessary kernel modules to be loaded and execute within 1 month of implementing the patch management policy.
- ✓ Disable unnecessary services and protocols within 2 weeks of implementing the patch management policy.
- ✓ Implement system activity monitoring and logging within 3 months of receiving the report.

Vulnerability: Outdated Samba version

Description: The Samba version installed on the server is outdated and vulnerable to several known exploits. This could allow an attacker to gain unauthorized access to the server or to sensitive information stored on the server.

Impact: An attacker could exploit this vulnerability to execute arbitrary code, gain root access to the server, or launch a denial-of-service attack.

Remediation Recommendation:

- ✓ Upgrade Samba to the latest stable version (Samba 4.17.5). The latest version includes security patches and fixes for known vulnerabilities.
- ✓ Configure Samba to only use secure protocols (e.g. SMBv3) and disable older, vulnerable protocols (e.g. SMBv1) to reduce the attack surface.
- ✓ Ensure that Samba is configured to use strong authentication protocols (e.g. Kerberos) and that all users have strong passwords.
- ✓ Implement a regular patch management policy to ensure that all software and systems are regularly updated with the latest security patches and updates.
- ✓ Monitor and log all Samba activity to detect and prevent potential attacks.

Timeline:

- ✓ Upgrade Samba to the latest stable version within 2 weeks of receiving the report.
- ✓ Configure Samba to only use secure protocols and ensure strong authentication within 2 weeks of upgrading to the latest version.
- ✓ Implement the patch management policy within 1 month of receiving the report.
- ✓ Monitor and log all Samba activity within 3 months of receiving the report.

Vulnerability: SMB signing not required

Description: The system is configured to allow unauthenticated SMB connections and does not require SMB signing, which can allow an attacker to perform man-in-the-middle attacks, tamper with data in transit, and potentially execute arbitrary code on the system.

Impact: An attacker can use this vulnerability to intercept and modify data in transit, execute arbitrary code on the system, and potentially gain unauthorized access to sensitive information.

Remediation Recommendation:

- ✓ Require SMB signing for all SMB connections to the system to ensure that data in transit is not tampered with or intercepted.
- ✓ Disable unauthenticated SMB connections to the system to prevent unauthorized access.
- ✓ Implement a regular patch management policy to ensure that all software and systems are regularly updated with the latest security patches and updates.
- ✓ Monitor and log all SMB activity to detect and prevent potential attacks.

Timeline:

- ✓ Require SMB signing for all SMB connections within 1 week of receiving the report.
- ✓ Disable unauthenticated SMB connections within 2 weeks of receiving the report.
- ✓ Implement the patch management policy within 1 month of receiving the report.
- ✓ Monitor and log all SMB activity within 3 months of receiving the report.

Vulnerability: HTTP TRACE/TRACK methods allowed

Description: The system is configured to allow HTTP TRACE and TRACK methods, which can allow an attacker to perform cross-site scripting (XSS) attacks, inject malicious scripts, and potentially steal sensitive information.

Impact: An attacker can use this vulnerability to inject malicious scripts into a webpage, steal sensitive information, and potentially gain unauthorized access to the system.

Remediation Recommendation:

- ✓ Disable the HTTP TRACE and TRACK methods to prevent potential attacks.
- ✓ Implement web application firewalls (WAFs) to block any attempts to exploit the vulnerability.
- ✓ Implement secure coding practices to prevent XSS attacks and other types of injection attacks.
- ✓ Educate developers and system administrators on the potential risks of allowing HTTP TRACE and TRACK methods and the importance of regularly reviewing and updating security policies.

Timeline:

- ✓ Disable HTTP TRACE and TRACK methods within 1 week of receiving the report.
- ✓ Implement WAFs within 2 weeks of receiving the report.
- ✓ Implement secure coding practices within 1 month of receiving the report.
- ✓ Educate developers and system administrators within 2 months of receiving the report.

APPENDICES

DETAILED METHODOLOGY:

The penetration testing assessment was conducted using a combination of manual and automated testing techniques, with the goal of identifying vulnerabilities and exploits that could be used to compromise the security of the target system.

The testing methodology followed the following steps:

- ✓ Reconnaissance – The first step was to gather information on the target system, including IP addresses, domain names, and network topology. This information was obtained using both passive and active reconnaissance techniques, including port scanning and OS fingerprinting.

- ✓ Enumeration – The next step was to identify the services and applications running on the target system, as well as any users and groups that have access to those systems. This was accomplished using a combination of manual and automated techniques, including banner grabbing, directory enumeration, and brute-force password guessing.

- ✓ Vulnerability Scanning – Once the target system was identified and its services and applications enumerated, vulnerability scanning tools were used to identify potential vulnerabilities. These tools included Nessus and Nmap scripts.

- ✓ Exploitation – After identifying potential vulnerabilities, I attempted to exploit them using both manual and automated techniques. This included using publicly available exploit code as well as modifying custom exploits to target specific vulnerabilities.

- ✓ Post-Exploitation – I was successful in exploiting a vulnerability, the next step was to escalate privileges and maintain access to the system. This involved a variety of post-exploitation techniques, including privilege escalation, backdoor creation, and data exfiltration.

- ✓ Reporting – Finally, I compiled a detailed report of the findings, including descriptions of vulnerabilities, potential impacts, and remediation recommendations.

TECHNICAL DETAILS

Vulnerability: Outdated Software Version of Apache 1.3.28

Apache version 1.3.28 was identified on the target system. This version of Apache was released on November 19, 2002, and is no longer supported. There are several known vulnerabilities associated with this version of Apache, including the following:

- ✓ The 'mod_alias' module in Apache 1.3.x through 1.3.31 allows remote attackers to read arbitrary files via a .. (dot dot) in an HTTP request, as demonstrated by /..%2f sequences in a URI. This vulnerability is documented in **CVE-2004-0885.**
- ✓ The 'mod_cgid' module in Apache 1.3 through 1.3.31, and 2.0 through 2.0.46, allows remote attackers to execute arbitrary scripts as root via a buffer overflow attack on the nargv[0] variable, a different vulnerability **than CVE-2004-0174.** This vulnerability is documented in **CVE-2004-0493.**
- ✓ The 'mod_ssl' module in Apache 1.3.x through 1.3.29, and 2.0.x through 2.0.48, does not properly revoke SSL session keys when Apache is restarted, which could allow remote attackers to bypass intended access restrictions. This vulnerability is documented in **CVE-2003-0192.**

Given the age of this version of Apache, it is likely that other vulnerabilities exist that have not been publicly disclosed.

Recommendation:

Upgrade to the latest version of Apache (9.2.x) which is still being supported. If the system cannot be upgraded, apply the latest patches available from the Apache website to mitigate the known vulnerabilities. It is also recommended to implement additional security measures such as web application firewalls to provide an additional layer of protection.

Vulnerability: Outdated Software Version of OpenSSH 3.1

Description: OpenSSH version 3.1 is installed on the target system, which is outdated and no longer supported. This version contains several known vulnerabilities that can be exploited by attackers to gain unauthorized access to the system.

Impact: Attackers can exploit vulnerabilities in OpenSSH 3.1 to gain unauthorized access to the system and potentially compromise sensitive data or perform other malicious actions.

Exploitation Method: Attackers can exploit vulnerabilities in OpenSSH 3.1 by using various techniques such as brute-force attacks, dictionary attacks, and other methods to guess weak passwords or gain access to cryptographic keys used by the system. Additionally, attackers can exploit known vulnerabilities in OpenSSH 3.1 to bypass authentication mechanisms and gain unauthorized access to the system.

Technical Details: The following known vulnerabilities are present in OpenSSH 3.1:

- ✓ **CVE-2002-0083:** This vulnerability allows remote attackers to execute arbitrary code or cause a denial of service (DoS) via a buffer overflow attack.
- ✓ **CVE-2002-0084:** This vulnerability allows remote attackers to execute arbitrary code via a buffer overflow attack.
- ✓ **CVE-2002-0085:** This vulnerability allows remote attackers to execute arbitrary code via a buffer overflow attack.
- ✓ **CVE-2002-0086:** This vulnerability allows remote attackers to cause a denial of service (DoS) via a malformed SSH1 packet.

Recommendation:

Upgrade OpenSSH to the latest stable version to mitigate the known vulnerabilities present in version 9.2. Additionally, implement best practices for secure authentication such as using strong passwords or multi-factor authentication to prevent unauthorized access to the system.

Vulnerability - Outdated Software Version of OpenSSL 0.9.6b

Description: OpenSSL is an open-source software library for secure communication over the internet. OpenSSL version 0.9.6b is installed on the target system, which was released on December 7, 2001. This version of OpenSSL contains several known vulnerabilities that can allow attackers to bypass security controls, perform man-in-the-middle attacks, and potentially compromise the confidentiality, integrity, and availability of data.

Impact: An attacker could exploit the known vulnerabilities in OpenSSL 0.9.6b to gain unauthorized access to sensitive information, such as user credentials, and potentially take control of the system. This could result in a data breach, system downtime, and reputational damage.

Technical Details: The following vulnerabilities were identified in OpenSSL 0.9.6b:

- ✓ **CVE-2003-0543:** The SSL/TLS protocol does not properly handle handshake messages that lack certain fields, which could allow an attacker to cause a denial of service (crash) via malformed messages.
- ✓ **CVE-2002-0656:** The SSL/TLS protocol does not properly handle certain errors, which could allow an attacker to cause a denial of service (crash) via malformed messages.
- ✓ **CVE-2002-0655:** The SSL/TLS protocol does not properly handle certain errors, which could allow an attacker to cause a denial of service (crash) via malformed messages.

Recommendation:

Upgrade to a newer version of OpenSSL that addresses the identified vulnerabilities. The latest version of OpenSSL (3.0.8) is not affected by these vulnerabilities and includes additional security features and improvements. Additionally, ensure that regular patch management is implemented to keep all software up to date.

Vulnerability: Outdated Software Version of Linux Kernel 2.4.7

During the penetration testing engagement, an outdated version of Linux Kernel 2.4.7 was identified on the target system. The Linux Kernel 2.4.7 was released on August 6, 2001, and is considered an outdated version. As of the time of this penetration testing engagement, the latest version of the Linux Kernel was 5.14.x. Running an outdated version of the kernel could expose the system to known vulnerabilities that could be exploited by attackers. Additionally, the outdated kernel version may not have the latest security patches, making it easier for attackers to gain unauthorized access.

The following technical details were observed during the testing:

- ✓ **CVE-2002-0839:** A vulnerability in the IPv4 handling code could allow remote attackers to cause a denial of service (system crash) by sending a large number of packets to a victim machine. This vulnerability affects the Linux kernel versions up to 2.4.19.
- ✓ **CVE-2003-0244:** A vulnerability in the signal handling code could allow local users to cause a denial of service (system crash) by sending a signal to a victim process. This vulnerability affects the Linux kernel versions up to 2.4.21.
- ✓ **CVE-2004-1235:** A vulnerability in the Linux kernel could allow local users to gain root privileges by exploiting a race condition in the sysctl() system call. This vulnerability affects the Linux kernel versions up to 2.4.27.
- ✓ **CVE-2005-1761:** A vulnerability in the IPsec code could allow remote attackers to cause a denial of service (system crash) by sending a specially crafted packet to a victim machine. This vulnerability affects the Linux kernel versions up to 2.4.30.
- ✓ **CVE-2006-1242:** A vulnerability in the keyring handling code could allow local users to cause a denial of service (system crash) or possibly gain privileges by accessing a keyring object that is being modified concurrently. This vulnerability affects the Linux kernel versions up to 2.4.32.
- ✓ **CVE-2008-3276:** A vulnerability in the NFS client code could allow remote attackers to cause a denial of service (system crash) by sending a specially crafted packet to a victim machine. This vulnerability affects the Linux kernel versions up to 2.4.36.

It is recommended that the system be updated to the latest stable version of the Linux kernel (5.14.0) to address these vulnerabilities. Additionally, it is recommended that security patches and updates be applied on a regular basis to ensure the system is protected against the latest threats.

Vulnerability: Outdated Software Version of Samba 2.2.1a

The Samba service is a network file and print service that enables file and printer sharing across different operating systems. Samba version 2.2.1a was found to be running on the target system during the penetration testing engagement.

Impact: The outdated version of Samba (2.2.1a) may be vulnerable to known exploits, which could be used by attackers to gain unauthorized access to the system, execute remote commands or escalate privileges.

Technical Details:

- ✓ **CVE-2007-2447:** This vulnerability allows attackers to execute remote commands with root privileges by sending a specially crafted packet to the target system. Samba version 2.2.1a is affected by this vulnerability and should be updated to a more recent version.
- ✓ **CVE-2017-7494:** This vulnerability allows attackers to execute arbitrary code with root privileges by exploiting a flaw in the Samba file sharing service. Samba version 2.2.1a is affected by this vulnerability and should be updated to a more recent version.
- ✓ **CVE-2018-1057:** This vulnerability allows attackers to execute arbitrary code with root privileges by exploiting a flaw in the Samba file sharing service. Samba version 2.2.1a is affected by this vulnerability and should be updated to a more recent version.
- ✓ **CVE-2019-12435:** This vulnerability allows attackers to execute arbitrary code with root privileges by exploiting a flaw in the Samba file sharing service. Samba version 2.2.1a is affected by this vulnerability and should be updated to a more recent version.

Recommendation:

Upgrade Samba to the latest version: Samba version 2.2.1a is no longer supported and should be upgraded to a more recent version (4.17.5) to address the known vulnerabilities. It is recommended to install the latest security patches for Samba and enable the necessary security features to prevent unauthorized access to the system.

EVIDENCE

Vulnerability Report: The vulnerability report generated by the vulnerability scanner (Nessus) identified several critical vulnerabilities present in the target systems. These include:

- ✓ Outdated version of OpenSSL (0.9.6b) with several known vulnerabilities
- ✓ Outdated version of Apache (1.3.28) with several known vulnerabilities
- ✓ Outdated version of OpenSSH (3.1) with several known vulnerabilities

192.168.68.145

Vulnerabilities Total: 130

SEVERITY	CVSS V3.0	PLUGIN	NAME
CRITICAL	7.5	34460	Unsupported Web Server Detection
CRITICAL	7.2	11915	Apache < 1.3.29 Multiple Modules Local Overflow
CRITICAL	7.1	11793	Apache < 1.3.28 Multiple Vulnerabilities (DoS, ID)
CRITICAL	10.0	10883	OpenSSH < 3.1 Channel Code Off by One Remote Privilege Escalation
CRITICAL	10.0	11031	OpenSSH < 3.4 Multiple Remote Overflows
CRITICAL	10.0	11837	OpenSSH < 3.7.1 Multiple Vulnerabilities
CRITICAL	10.0	78555	OpenSSL Unsupported
HIGH	9.3	17760	OpenSSL < 0.9.8f Multiple Vulnerabilities
HIGH	9.3	57459	OpenSSL < 0.9.8s Multiple Vulnerabilities
HIGH	7.5	11137	Apache < 1.3.27 Multiple Vulnerabilities (DoS, XSS)
HIGH	7.5	31654	Apache < 1.3.37 mod_rewrite LDAP Protocol URL Handling Overflow
HIGH	7.5	11030	Apache Chunked Encoding Remote Overflow
HIGH	7.5	13651	Apache mod_ssl ssl_engine_log.c mod_proxy Hook Function Remote Format String
HIGH	7.5	10771	OpenSSH 2.5.x - 2.9 Multiple Vulnerabilities
HIGH	7.5	44072	OpenSSH < 3.2.3 YP Netgroups Authentication Bypass
HIGH	7.5	11712	OpenSSH < 3.6.2 Reverse DNS Lookup Bypass
HIGH	7.5	44077	OpenSSH < 4.5 Multiple Vulnerabilities
HIGH	7.5	44078	OpenSSH < 4.7 Trusted X11 Cookie Connection Policy Bypass

HIGH	7.5	10954	OpenSSH Kerberos TGT/AFS Token Passing Remote Overflow
HIGH	7.5	17751	OpenSSL 0.9.6 CA Basic Constraints Validation Vulnerability
HIGH	7.5	17746	OpenSSL < 0.9.6e Multiple Vulnerabilities
HIGH	7.5	17752	OpenSSL < 0.9.7-beta3 Buffer Overflow
HIGH	7.5	58799	OpenSSL < 0.9.8w ASN.1 asn1_d2i_read_bio Memory Corruption
HIGH	7.5	10882	SSH Protocol Version 1 Session Key Retrieval
HIGH	7.5	12255	mod_ssl ssl_util_uuencode_binary Remote Overflow
HIGH	7.2	10823	OpenSSH < 3.0.2 Multiple Vulnerabilities
HIGH	7.2	17702	OpenSSH < 3.6.1p2 Multiple Vulnerabilities
HIGH	7.1	20007	SSL Version 2 and 3 Protocol Detection
HIGH	5.0	35291	SSL Certificate Signed Using Weak Hashing Algorithm
HIGH	5.0	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
MEDIUM	6.9	31737	OpenSSH X11 Forwarding Session Hijacking
MEDIUM	6.8	10802	OpenSSH < 3.0.1 Multiple Flaws
MEDIUM	6.5	44079	OpenSSH < 4.9 'ForceCommand' Directive Bypass
MEDIUM	6.4	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.4	57582	SSL Self-Signed Certificate
MEDIUM	6.1	104743	TLS Version 1.0 Protocol Detection
MEDIUM	5.8	17762	OpenSSL < 0.9.8j Signature Spoofing
MEDIUM	5.8	42880	SSL / TLS Renegotiation Handshakes MiTM Plaintext Data Injection
MEDIUM	5.1	17765	OpenSSL < 0.9.8l Multiple Vulnerabilities
MEDIUM	5.0	40984	Browsable Web Directories
MEDIUM	5.0	11213	HTTP TRACE / TRACK Methods Allowed
MEDIUM	5.0	44073	OpenSSH With OpenPAM DoS
MEDIUM	5.0	59076	OpenSSL 0.9.8 < 0.9.8x DTLS CBC Denial of Service

MEDIUM	5.0	17747	OpenSSL < 0.9.6f Denial of Service
MEDIUM	5.0	17748	OpenSSL < 0.9.6k Denial of Service
MEDIUM	5.0	17749	OpenSSL < 0.9.6l Denial of Service
MEDIUM	5.0	17750	OpenSSL < 0.9.6m / 0.9.7d Denial of Service
MEDIUM	5.0	12110	OpenSSL < 0.9.6m / 0.9.7d Multiple Remote DoS
MEDIUM	5.0	17759	OpenSSL < 0.9.8 Weak Default Configuration
MEDIUM	5.0	17761	OpenSSL < 0.9.8i Denial of Service
MEDIUM	5.0	17763	OpenSSL < 0.9.8k Multiple Vulnerabilities
MEDIUM	5.0	58564	OpenSSL < 0.9.8u Multiple Vulnerabilities
MEDIUM	5.0	44074	Portable OpenSSH < 3.8p1 Multiple Vulnerabilities
MEDIUM	5.0	57608	SMB Signing not required
MEDIUM	5.0	15901	SSL Certificate Expiry
MEDIUM	5.0	45411	SSL Certificate with Wrong Hostname
MEDIUM	4.6	44076	OpenSSH < 4.3 scp Command Line Filename Processing Command Injection
MEDIUM	4.3	17696	Apache HTTP Server 403 Error Page UTF-7 Encoded XSS
MEDIUM	4.3	88098	Apache Server ETag Header Information Disclosure
MEDIUM	4.3	11267	OpenSSL < 0.9.6j / 0.9.7b Multiple Vulnerabilities
MEDIUM	4.3	56996	OpenSSL < 0.9.8h Multiple Vulnerabilities
MEDIUM	4.3	51892	OpenSSL SSL_OP_NETSCAPE_REUSE_CIPHER_CHANGE_BUG Session Resume Ciphersuite Downgrade Issue
MEDIUM	4.3	90317	SSH Weak Algorithms Supported
MEDIUM	4.3	89058	SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and Weakened eNcryption)
MEDIUM	4.3	65821	SSL RC4 Cipher Suites Supported (Bar Mitzvah)
MEDIUM	4.3	26928	SSL Weak Cipher Suites Supported
MEDIUM	4.3	81606	SSL/TLS EXPORT_RSA <= 512-bit Cipher Suites Supported (FREAK)

MEDIUM	4.3	78479	SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE)
MEDIUM	4.3	10816	Webalizer < 2.01-09 Multiple XSS
MEDIUM	4.0	44065	OpenSSH < 5.2 CBC Plaintext Disclosure
LOW	3.5	19592	OpenSSH < 4.2 Multiple Vulnerabilities
LOW	2.6	64532	OpenSSL < 0.9.8y Multiple Vulnerabilities
LOW	2.6	70658	SSH Server CBC Mode Ciphers Enabled
LOW	2.6	71049	SSH Weak MAC Algorithms Enabled
LOW	2.6	83738	SSL/TLS EXPORT_DHE <= 512-bit Export Cipher Suites Supported (Logjam)
LOW	2.1	17754	OpenSSL < 0.9.7f Insecure Temporary File Creation
LOW	2.1	53841	Portable OpenSSH ssh-keysign ssh-rand-helper Utility File Descriptor Leak Local Information Disclosure
LOW	1.2	44075	OpenSSH < 4.0 known_hosts Plaintext Host Information Disclosure
LOW	1.2	44080	OpenSSH X11UseLocalhost X11 Forwarding Port Hijacking
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	10223	RPC portmapper Service Detection
INFO	N/A	48204	Apache HTTP Server Version
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	132634	Deprecated SSLv2 Connection Attempts
INFO	N/A	54615	Device Type
INFO	N/A	35716	Ethernet Card Manufacturer Detection
INFO	N/A	86420	Ethernet MAC Addresses
INFO	N/A	49704	External URLs
INFO	N/A	84502	HSTS Missing From HTTPS Server
INFO	N/A	43111	HTTP Methods Allowed (per directory)
INFO	N/A	10107	HTTP Server Type and Version

INFO	N/A	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	11011	Microsoft Windows SMB Service Detection
INFO	N/A	106716	Microsoft Windows SMB2 and SMB3 Dialects Supported (remote check)
INFO	N/A	50344	Missing or Permissive Content-Security-Policy frame-ancestors HTTP Response Header
INFO	N/A	50345	Missing or Permissive X-Frame-Options HTTP Response Header
INFO	N/A	11219	Nessus SYN scanner
INFO	N/A	19506	Nessus Scan Information
INFO	N/A	11936	OS Identification
INFO	N/A	117886	OS Security Patch Assessment Not Available
INFO	N/A	57323	OpenSSL Version Detection
INFO	N/A	66334	Patch Report
INFO	N/A	11111	RPC Services Enumeration
INFO	N/A	53335	RPC portmapper (TCP)
INFO	N/A	70657	SSH Algorithms and Languages Supported
INFO	N/A	149334	SSH Password Authentication Accepted
INFO	N/A	10881	SSH Protocol Versions Supported
INFO	N/A	10267	SSH Server Type and Version Information
INFO	N/A	56984	SSL / TLS Versions Supported
INFO	N/A	45410	SSL Certificate 'commonName' Mismatch
INFO	N/A	10863	SSL Certificate Information
INFO	N/A	70544	SSL Cipher Block Chaining Cipher Suites Supported
INFO	N/A	21643	SSL Cipher Suites Supported
INFO	N/A	57041	SSL Perfect Forward Secrecy Cipher Suites Supported
INFO	N/A	58768	SSL Resume With Different Cipher Issue
INFO	N/A	94761	SSL Root Certification Authority Certificate Information

INFO	N/A	53360	SSL Server Accepts Weak Diffie-Hellman Keys
INFO	N/A	51891	SSL Session Resume Supported
INFO	N/A	22964	Service Detection
INFO	N/A	25220	TCP/IP Timestamps Supported
INFO	N/A	110723	Target Credential Status by Authentication Protocol - No Credentials Provided
INFO	N/A	10287	Traceroute Information
INFO	N/A	20094	VMware Virtual Machine Detection
INFO	N/A	135860	WMI Not Available
INFO	N/A	91815	Web Application Sitemap
INFO	N/A	11032	Web Server Directory Enumeration
INFO	N/A	49705	Web Server Harvested Email Addresses
INFO	N/A	11422	Web Server Unconfigured - Default Install Page Present
INFO	N/A	10662	Web mirroring
INFO	N/A	10150	Windows NetBIOS / SMB Remote Host Information Disclosure

GLOSSARY

- ✓ **Arbitrary commands:** Commands that are executed without restriction or limitations, potentially allowing an attacker to perform unauthorized actions on a system or network.
- ✓ **Attacker:** A person or entity that attempts to compromise the security of a system or network for malicious purposes.
- ✓ **Buffer overflow:** A type of software vulnerability that occurs when a program attempts to write more data to a buffer than it can hold, potentially allowing an attacker to execute arbitrary code or crash the program.
- ✓ **Cross-site scripting (XSS):** A type of security vulnerability that allows an attacker to inject malicious code into a web page viewed by other users.
- ✓ **Denial of Service (DoS) Attack:** An attack that floods a network or system with traffic in order to overload it and prevent legitimate traffic from being processed.
- ✓ **Exploit:** A technique or piece of software that takes advantage of a vulnerability to gain unauthorized access or control over a system.
- ✓ **HTTP web server:** A software application that runs on a web server and serves HTTP content to web clients, such as web browsers.
- ✓ **Intrusion Detection System (IDS):** A security system that monitors network traffic for suspicious activity and alerts security personnel when potential intrusions are detected.
- ✓ **Patch:** A software update that fixes known vulnerabilities or bugs in a system.
- ✓ **Penetration Testing:** A simulated attack on a system or network designed to identify vulnerabilities and weaknesses that could be exploited by malicious actors.
- ✓ **Privilege escalation:** The process of elevating user privileges beyond what is normally allowed, potentially allowing an attacker to gain access to sensitive information or perform malicious actions.
- ✓ **SSH (Secure Shell):** A cryptographic network protocol that allows secure communication between networked devices, typically used for remote command-line login and remote command execution.
- ✓ **SSL (Secure Sockets Layer):** A cryptographic protocol that provides secure communication over a network, typically used to secure web traffic.
- ✓ **Vulnerability:** A flaw or weakness in a system that can be exploited to compromise the security of that system.

REFERENCES

- ✓ **Exploit Database** https://www.exploit-db.com/
- ✓ **Linux Privilege Escalation** https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Linux%20-%20Privilege%20Escalation.md
- ✓ **Metasploit Framework** https://www.metasploit.com/
- ✓ **Nessus Professional** https://www.tenable.com/products/nessus/professional