James Cook University

CP3414 Assignment 3

Vulnerability Assessment

MR. Steve Kerrison

Yunseo Choi

11th April 2025

Executive Summary

This report explains the results of a Nessus scan done on your company’s network of subnet 192.168.7.0/24. The Nessus scan was already completed, and this report emphasizes understanding findings and what countermeasures should be taken

This network contains a windows machine, Linux web server, and raspberry pi

In this report, several medium and high-risk vulnerabilities are found, some critical issues such as

Apache webserver been affected by vulnerabilities

Remote OS is unsupported

This Nessus scan report gives clear information of what vulnerabilities were found, listing vulnerabilities ordering by host and criticalness, steps to reduce the risk, in some cases of scanned risk, simply by updating settings can solve the problem

Summary of findings

192.168.7.6

This host has no critical issues, but 7 informational findings were reported. One of the findings revealed that the system is running an NTP server. While this is commonly used to synchronize clocks across devices, if not securely configured, it may expose version information or system details useful for attackers.

192.168.7.7

No major security issues were found on this host, but 29 informational alerts were detected. One notable finding is that the server lacks a strict content security policy header, which helps prevent clickjacking and cross-site scripting Without this, the site may be embedded into malicious domains, increasing the risk of browser-based attacks

192.168.7.8

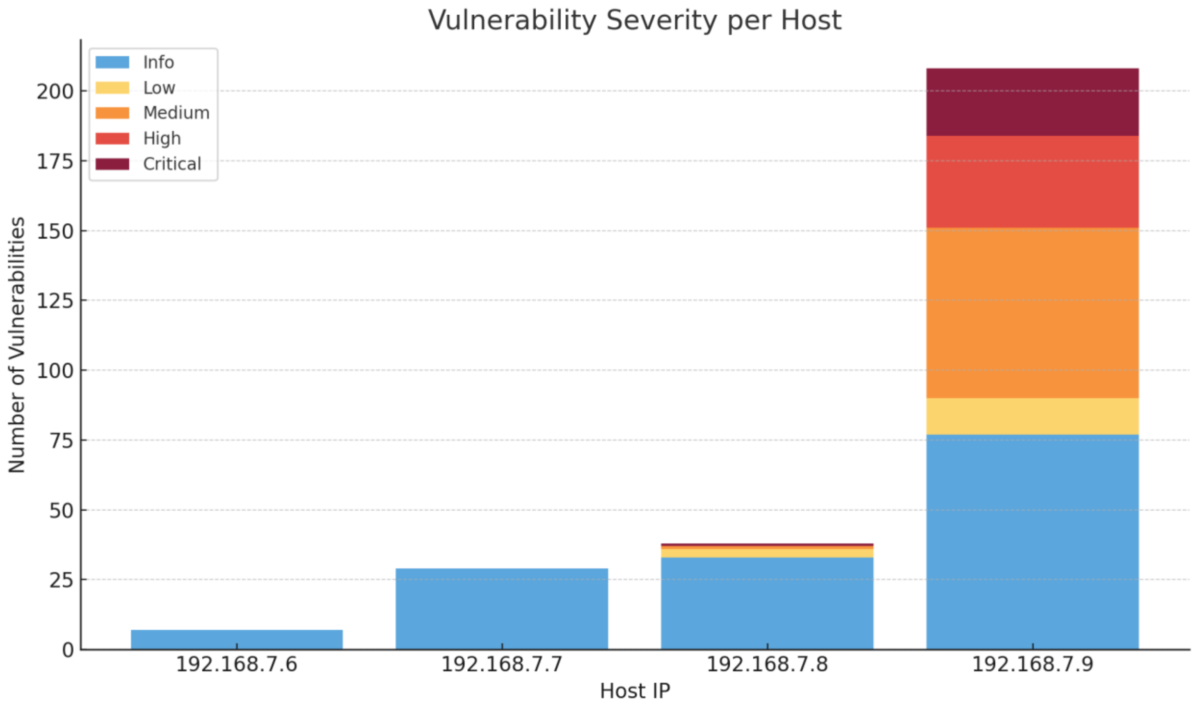
This host showed 1 critical vulnerability due to an unsupported Windows operating system, which may not receive security updates, leaving it open to exploitation. A medium vulnerability was found where SMB signing is not required, allowing a possible man-in-the-middle attack. Among 33 informational issues, the server was found to be running DCE/RPC services and responding to ICMP timestamp requests, both of which could be leveraged for further exploitation or bypassing time-based authentication such as OTP

192.168.7.9

This system has the most severe issues, with 24 critical, 33 high, 61 medium, 13 low, and 77 informational findings. The most critical problems relate to multiple outdated and unsupported versions of Apache HTTP Server, such as versions older than 2.4.53 and 1.3.x. These versions are affected by serious vulnerabilities, including request smuggling, memory overflows, and denial of service. Several high-risk issues involve known bugs in Apache's mod_rewrite module, chunked encoding handling, and an old version of OpenSSL that could lead to memory corruption and remote code execution

A medium risk vulnerability was found due to insecure SSL/TLS renegotiation, which could let attackers inject malicious data into a secure session and conduct a man-in-the-middle attack. Finally, a low-risk issue was identified with an outdated version of OpenSSH, which stores sensitive data like hostnames and IP addresses in plaintext, potentially exposing the system to local attackers

Graph of Criticality of Vulnerabilities



Technical findings

192.168.7.6

Vulnerability: Nessus Scan Information

Description:

This finding provides meta-information about the Nessus scan, such as the plugin set used, the scanner type, port scanning range, scan date and duration, and whether credentialed checks were enabled. It does not represent vulnerability but helps verify how the scan was conducted

Risk Level: Informational

Impact: None.

Vulnerability: ICMP Timestamp Request Remote Date Disclosure

Description:

The system responds to ICMP timestamp requests, which allows an attacker to determine the system's clock time. Knowing the exact time of a target machine can help an attacker bypass time-based security mechanism, such as one-time passwords or time-restricted tokens. This is often used during reconnaissance phases in more advanced attacks

Risk Level: Informational

Impact: Can let attackers perform timing attacks or enumerate time sensitive authentication mechanisms

192.168.7.7

Vulnerability: HTTP/2 Cleartext Detection

Description:

The remote server supports HTTP/2 over cleartext TCP, While HTTP/2 improves performance, using it without TLS exposes communication to potential eavesdropping or manipulation

Risk Level: Informational

Impact: An attacker on the same network could intercept or modify unencrypted traffic

Vulnerability: Missing or Permissive X-Frame-Options HTTP Response Header

Description:

The web server does not include a properly configured X-Frame-Options header in its HTTP responses. This makes the application vulnerable to clickjacking attacks, where a malicious site could trick users into interacting with a hidden interface, such as clicking buttons or submitting forms without realizing it

Risk Level: Informational

Impact: None

192.168.7.8

Vulnerability: Unsupported remote Windows OS

Description:

The remote system is running Microsoft Windows 7 Ultimate, which is no longer supported. Unsupported systems are likely to contain unpatched security vulnerabilities

Risk Factor: Critical

Impact: The system may be exposed to attacks that exploit known but unpatched vulnerabilities

Vulnerability: SMB Signing not required

Description:

The remote SMB server does not enforce signing. This means a remote, unauthenticated attacker could perform a man-in-the-middle attack against the server

Risk Factor: Medium

Impact: An attacker may intercept or modify SMB traffic between the server and client

192.168.7.9

Vulnerability: Apache 2.4.x < 2.4.53 Multiple Vulnerabilities

Description:

The Apache web server on this host is running version 1.3.20, which is older than 2.4.53. This version is affected by multiple vulnerabilities, including use of uninitialized values in mod_lua, HTTP request smuggling, potential buffer overflows with large LimitXMLRequestBody, and out of bounds writes in mod_sed

Risk Factor: High

Impact: These issues may allow attackers to crash the server, perform request smuggling, or write to memory outside of allocated bounds

Vulnerability: Apache 2.4.x < 2.4.54 Multiple Vulnerabilities

Description:

The Apache web server is running a version older than 2.4.54 and is affected by multiple additional vulnerabilities not covered in earlier advisories. These include issues such as request smuggling in mod_proxy_ajp, read beyond bounds in mod_isapi, denial of service via mod_lua, and dropped headers in mod_proxy, among others

Risk Factor: High

Impact: May allow denial of service, information disclosure, and request smuggling attacks

Vulnerability: Apache 2.4.x < 2.4.55 Multiple Vulnerabilities

Description:

The Apache web server is running a version earlier than 2.4.55 and is affected by several vulnerabilities. These include memory read/write issues triggered by specific headers, additional HTTP request smuggling in mod_proxy_ajp, and improper header handling that may cause security-related headers to be ignored by the client.

Risk Factor: High

Impact: These issues may lead to denial of service, request smuggling, or loss of header-based protection

Vulnerability: Apache < 1.3.28 Multiple Vulnerabilities

Description:

The Apache server is running a version older than 1.3.28. This version contains multiple flaws including denial of service in redirect handling, issues with control character processing in the rotatelogs utility, and a file descriptor leak when handling third party modules. This finding is based on version detection and may be a false positive

Risk Factor: High

Impact: These vulnerabilities may allow service disruption or resource leakage on the host

Vulnerability: Apache httpd SEoL

Description:

The server is running Apache httpd version 1.3.x or earlier, which is no longer supported by the vendor. Unsupported software does not receive security updates and may contain unpatched vulnerabilities

Risk Factor: High

Impact: The system may be at risk due to lack of future security patches

Vulnerability: OpenSSH < 3.4 Multiple Remote Overflows

Description:

The host appears to be running OpenSSH version 3.4 or older, which is known to have multiple remote overflow vulnerabilities. These could potentially be exploited by an attacker to gain shell access to the system. This detection is based on the SSH banner and may be a false positive if the system is patched without updating the version number

Risk Factor: Critical

Impact: Successful exploitation could allow remote access or control of the system

Vulnerability: OpenSSL Unsupported

Description:

The host is running a version of OpenSSL that is no longer supported. Unsupported versions no longer receive security patches, making them likely to contain vulnerabilities

Risk Factor: Critical

Impact: The system may be exposed to known or future security flaws without vendor-provided fixes

Vulnerability: SSL Version 2 and 3 Protocol Detection

Description:

The service accepts SSL 2.0 and/or SSL 3.0 connections. These protocols are outdated and contain known weaknesses such as insecure padding schemes and vulnerable session renegotiation mechanisms. Their use could allow attackers to decrypt or manipulate secure communications

Risk Factor: Critical

Impact: May allow man-in-the-middle attacks or decryption of encrypted traffic

Vulnerability: Apache Chunked Encoding Remote Overflow

Description:

The host is running a version of Apache web server that is vulnerable to a chunked encoding handling flaw. This issue may allow an attacker to execute arbitrary code remotely. The detection is based on the version number and could be a false positive if the system is patched without version update

Risk Factor: High

Impact: May allow remote code execution on the server

Vulnerability: Apache Server ETag Header Information Disclosure

Description:

The Apache web server includes inode values in the ETag header. This can result in information disclosure, as it reveals file system metadata such as inode numbers, which may aid an attacker during reconnaissance

Risk Factor: Medium

Impact: Could assist in fingerprinting or identifying files across requests

Vulnerability: Browsable Web Directories

Description:

The web server has one or more directories that are browsable. This means visitors can view the contents of these directories directly through a browser

Risk Factor: Medium

Impact: May expose sensitive files or information unintentionally

Vulnerability: HTTP TRACE / TRACK Methods Allowed

Description:

The web server allows HTTP TRACE and/or TRACK methods. These methods are typically used for debugging and should not be enabled on production systems

Risk Factor: Medium

Impact: May allow an attacker to exploit browser-based vulnerabilities or gather sensitive request information

Vulnerability: OpenSSH < 4.2 Multiple Vulnerabilities

Description:

The host is running an older version of OpenSSH that is affected by multiple issues. These include unintentional enabling of X11 forwarding, potential delegation of GSSAPI credentials, and authentication flaws that may lead to user enumeration or denial of service

Risk Factor: Low

Impact: Could allow exposure of credentials, user account enumeration, or service disruption under specific conditions

Recommendation

As discovered in the technical findings, most of the vulnerabilities were due to outdated software patches.

Keeping software up to date is a critical part of maintaining cybersecurity. Outdated software often contains known vulnerabilities that attackers can exploit to gain unauthorized access, steal data, or disrupt operations. Software developers regularly release patches to fix these flaws, and failing to apply updates leaves systems exposed to documented threats [1].

Regularly updating security patches is one of the most effective ways to protect systems from known vulnerabilities. When patches are delayed or ignored, it leaves devices exposed to cyber threats that could otherwise be prevented. Applying updates ensures that weaknesses identified by developers are resolved before attackers can exploit them. It is also recommended to automate updates and prioritize patching for critical systems and software components [2].

If ever software is unpatched up to date, it could lead to several major cybersecurity incidents directly linked to unpatched software. One of the most notorious cases was the Equifax breach in 2017, which compromised sensitive information of over 140 million individuals due to an unpatched Apache Struts vulnerability. Another significant example is the WannaCry ransomware attack, which exploited a known vulnerability in outdated Windows systems, impacting over 200,000 computers across 150 countries — including critical infrastructure like the UK’s National Health Service. Similarly, the Heartland Payment Systems breach resulted in the theft of millions of credit card details, all because of an unpatched SQL flaw. These incidents underscore that failing to apply available patches can lead to massive data loss, financial damage, and reputational harm [3].

Thus, patching Apache server up to date is crucial here is the process, Patching Apache HTTP Server should follow a structured and repeatable process to minimize risks during updates. Oracle recommends starting with version discovery, where administrators identify the current version of Apache running on each system. This helps determine whether the server is affected by any known

CVEs.

Once vulnerabilities are identified, administrators should review security advisories and change logs to assess the impact of each fix. Before applying patches, it is essential to back up Apache configuration files to ensure rollback capability in case of issues.

Next, the patch or upgrade should be tested in a staging environment that mirrors the production setup. This helps to ensure that the update does not disrupt functionality or introduce new issues.

Oracle further recommends creating a discovery and remediation runbook a documented procedure that outline process following next, Detect outdated versions, Assess vulnerabilities, Schedule maintenance windows, apply updates, test post patching behavior, and Record the patching history.

Finally, once testing is complete, the patch should be applied to production servers during a scheduled window, with monitoring in place to track server performance and errors. This disciplined approach not only ensures successful patching but also aligns with best practices for change management [4].

The next common vulnerability was due to ssh related problem,

SSH is a cryptographic network protocol used for secure access to remote machines and encrypted data transmission over untrusted networks. It is widely implemented in Linux and Unix based systems as the standard method for remote administration. SSH supports capabilities such as command execution, file transfers using SCP or SFTP, port forwarding, and secure tunneling. Additionally, it enables key based authentication and supports various encryption and authentication algorithms to safeguard communication integrity and confidentiality. Due to its critical role in system administration and automation, vulnerabilities in outdated SSH versions such as improper handling of user sessions or forwarded credentials can lead to unauthorized access or information disclosure. Ensuring SSH is regularly patched and securely configured is essential for maintaining system resilience [5].

Below demonstrates patching ssh up to date

The process of patching SSH involves checking the current version, identifying available updates, and upgrading both the SSH client and its dependencies. On Linux systems, administrators typically use package managers like apt, yum, or dnf to run commands such as sudo apt update sudo apt upgrade openssh server to fetch and apply the latest OpenSSH packages. On Windows, updating SSH may involve using built in tools like Windows PowerShell or the Windows Package Manager to check the installed version and apply updates. For macOS, SSH updates are managed through Homebrew or system updates. The source emphasizes the importance of verifying that updates were applied successfully by checking the version with ssh -V after patching. Regularly upgrading SSH ensures improved security, support for modern cryptographic standards, and fixes for vulnerabilities present in earlier builds [6].

Another crucial part, SSL

SSL certificates play a key role in protecting data shared between a user and a website. They ensure that information such as login details, financial records, and personal data is encrypted and secure from interception. According to eMazzanti Technologies, SSL also builds trust with users by showing visible security indicators, like the padlock icon in browsers [7]. However, having SSL is not enough organizations must also keep their SSL configurations and supporting software like OpenSSL up to date. Unpatched SSL implementations can contain vulnerabilities that undermine encryption, exposing data to attackers. Regular patching helps maintain strong encryption, fix bugs, and comply with updated security standards [7].

Other than those 3 most important and common vulnerabilities, recommendation for other vulnerabilities is

Subnet 192.168.7.8 is using Windows 7 Ultimate, which is no longer supported. This means it doesn't receive any security updates, making it easier for attackers to exploit known weaknesses. It is strongly recommended to upgrade this system to a newer version of Windows that still receives updates

192.168.7.9 allows TRACE and TRACK HTTP methods, which are mainly used for testing but can help attackers gather sensitive information. These methods should be turned off if not needed

References

[1] University of Idaho, “Why Keeping Your Software Up to Date is Important for Cybersecurity,” *University of Idaho - Knowledge Base*, Oct. 18, 2023. https://support.uidaho.edu/TDClient/40/Portal/KB/ArticleDet?ID=2770

[2] “Why Is It Important to Update Security Patches? - Abacus,” Aug. 24, 2021. https://goabacus.com/why-is-it-important-to-update-security-patches/

[3] “The 10 biggest security breaches from unpatched software,” 1E, Feb. 08, 2019. https://www.1e.com/blogs/10-unpatched-software-security-breaches/

[4] Oracle Help Center, 2024. https://docs.oracle.com/en/learn/oci-fam-patching-thirdparty/index.html#task-4-create-a-discovery-runbook-for-a-product (accessed Apr. 11, 2025).

[5] P. Loshin, “What is SSH (Secure Shell)? Definition from SearchSecurity,” SearchSecurity, Sep. 2021. https://www.techtarget.com/searchsecurity/definition/Secure-Shell

[6] “Learn how to update and upgrade your SSH client software and dependencies on Windows, Linux, and Mac, and how to check your SSH version and status.,” *Linkedin.com*, Jun. 15, 2023. https://www.linkedin.com/advice/1/how-do-you-update-upgrade-your-ssh-client-software (accessed Apr. 11, 2025).

[7] Kamil Smolag, “The Importance of SSL Certificates for Website Security - eMazzanti Technologies,” eMazzanti Technologies, Sep. 27, 2024. https://www.emazzanti.net/the-importance-of-ssl-certificates-for-website-security/ (accessed Apr. 11, 2025).