Vulnerability Management Report

Prepared by: Group 4

Date: April 10th, 2025

Sprint Two

CyBlack SOC Academy, 3rd Cohort.

This Report is a collective work compiled by all of the members of the CyBlack SOC Academy 3rd cohort, group 4

Group Members

- **Adebisi Mololuwa**
- **Aderinola Kehinde**
- **Aleka Moses Ogamodey**
- **Comfort Ukangwobia**
- **Gyekye Micaiah Ampofo**
- **Nwobodo Chigozie**
- **Odunayo Balogun**
- **Oghenetejiri Love Brume**
- **Victoria Simon**

Executive Summary

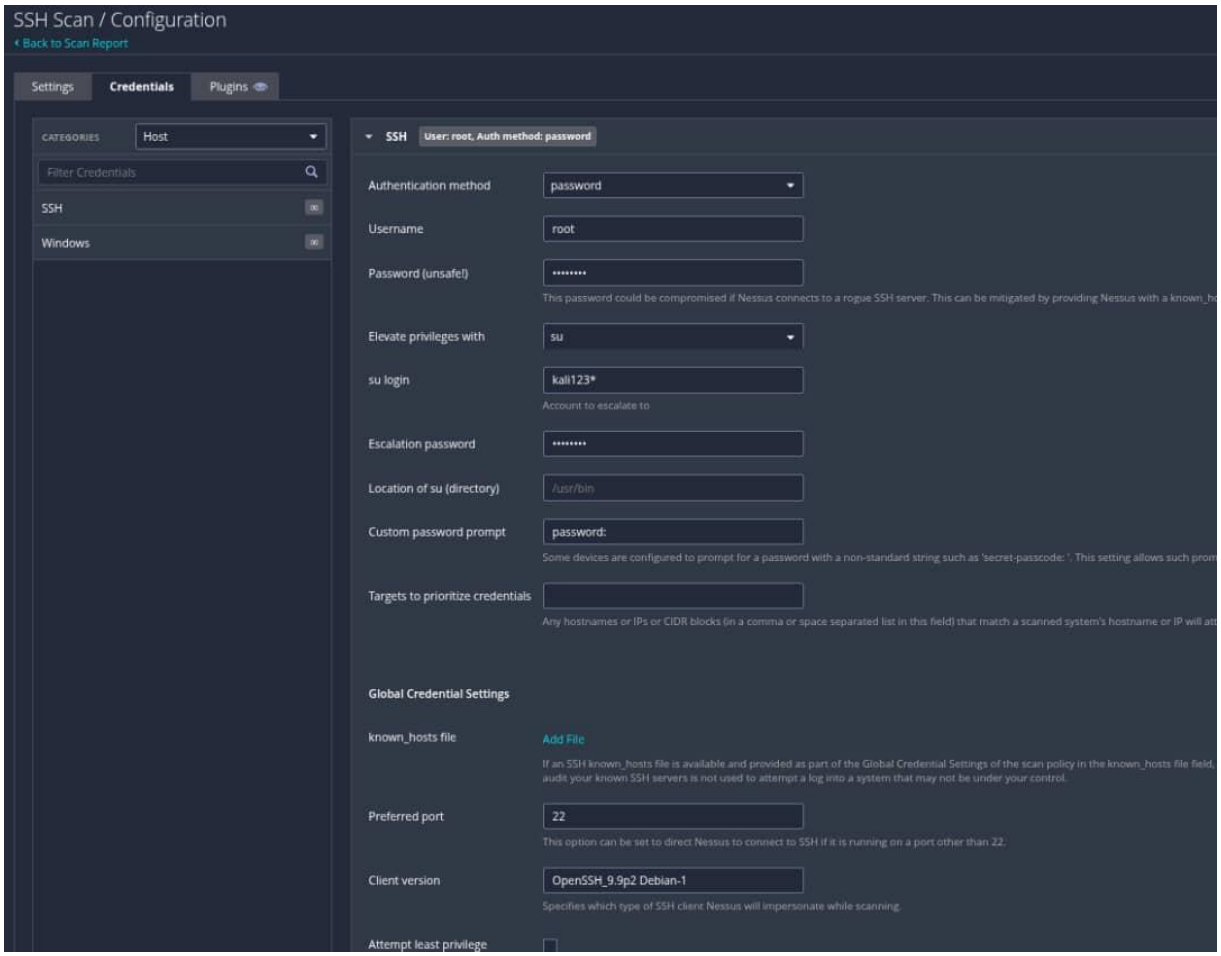
This report presents the findings from a credentialed scan of Linux systems and a web application vulnerability scan conducted using Nessus. The goal was to assess the security posture of internal infrastructure, highlight potential threats, and provide actionable recommendations for remediation. Key findings included multiple medium to critical vulnerabilities affecting system services such as Nginx, OpenSSH, and other underlying libraries. Immediate patching and security hardening steps have been proposed.

TASK 1: Credentials Scan Configuration

Scan Overview

To ensure compliance with ssecurity policy requiring regular audits of Linux systems, a credentialed vulnerability scan was carried out using the following configuration steps:

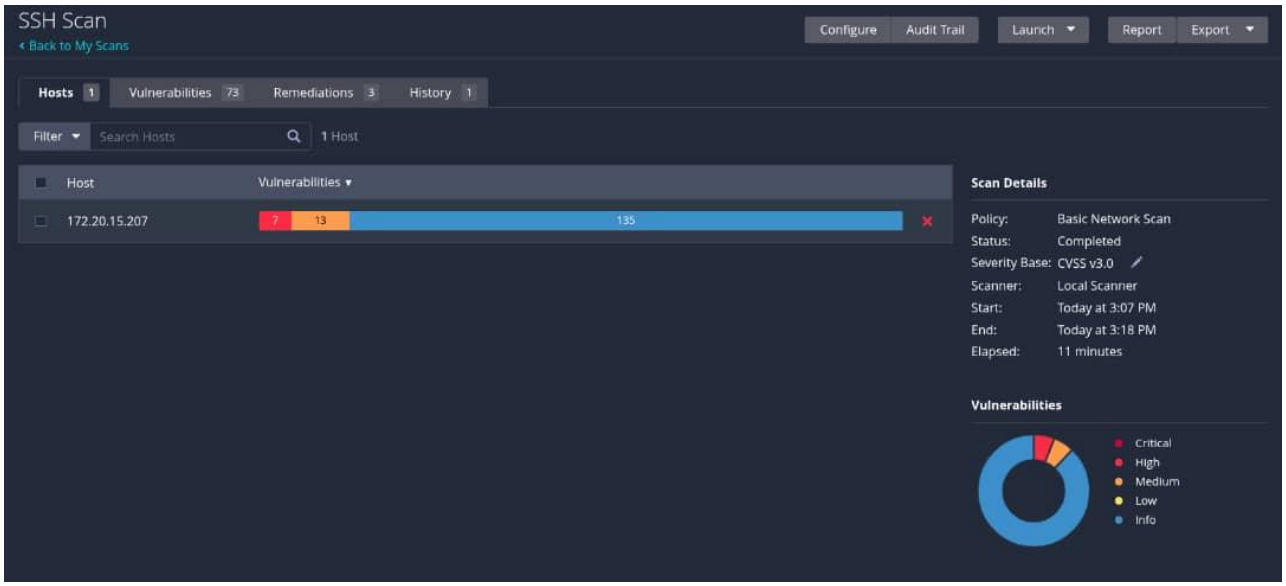
1. **Scan Type Selection**
    - Launched Nessus and created a new scan.
    - Selected **"Advanced Scan"** to enable credentialed access options.
2. **Target Definition**
    - Entered the IP address of the target Linux server in the **Targets** field.
3. **SSH Credential Configuration**
    - Navigated to the **Credentials** tab.
    - Selected **SSH** from the available authentication methods.
    - Configured the following fields:
        - **Username:** root
        - **Password:** kali (or the appropriate root password for Ubuntu)
        - **Privilege Elevation Method: su**
        - **Custom Password Prompt:** password:



(***SSH Configuration***)

4. **Verifying SSH Version**
    - Ran the command **ssh -V** on the terminal to identify the version of OpenSSH running on the target system.
    - Example output: **OpenSSH_9.9p2 Debian-1**
    - Input this version information in the **Scan Configuration Notes** for reference and analysis.
5. **Scan Execution**
    - Saved the scan configuration and initiated the credentialed scan.

- Nessus successfully authenticated via SSH and performed a deep inspection of services, packages, and security configurations on the Linux server.



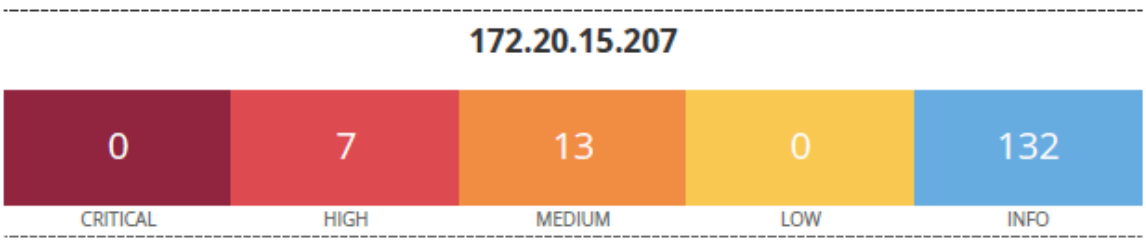
(***SSH Scan Output***)

6. Post-Scan Validation

- Verified that vulnerabilities were discovered with elevated access, confirming the credentialed scan was successful.
- Exported results for reporting and remediation planning.

Summary of SSH Scan Findings

The credentialed vulnerability scan on the SSH server revealed a total of 53 security issues categorized by risk levels as follows:



RECOMMENDATIONS

Following the credentialed scan and vulnerability assessment on the Linux server and hosted applications, the following remediation steps are recommended to improve the organization's security posture:

1. System and Software Patching:

- Upgrade **OpenSSH** to the latest secure version to address known vulnerabilities.
- Apply the latest **Linux kernel patches**, including urgent updates such as CVE-2022-0185.
- Upgrade **Node.js** to **version 18.20.1 or later** to mitigate risks like request smuggling, insecure randomness, and memory management vulnerabilities (e.g., CVE-2024-27980, CVE-2024-21891, CVE-2024-21892).
- Enable **automatic patch updates** or integrate with a centralized **patch management system** to ensure timely application of security fixes.

2. SSH Service Hardening

- Disable **weak SSH algorithms**, ciphers (e.g., CBC, Arcfour), and MACs.
- Enforce **key-based authentication** and disable password login where possible.
- Configure **custom SSH port** and restrict SSH access using firewall rules or TCP wrappers.
- Limit access to specific IPs and implement **two-factor authentication (2FA)** for remote connections.

3. User and Access Control

- Disable **root login via SSH**; use sudo for privilege elevation.
- Remove **unused or legacy user accounts** and enforce **strong password policies**.
- Set up **account lockout mechanisms** to prevent brute-force attacks.
- Audit user groups and privileges to ensure **least privilege** is enforced.

4. Host and Network Hardening

- Disable unnecessary services and close unused ports.
- Use hardening tools like **Lynis**, **OpenSCAP**, or **CIS Benchmarks** to enforce best practices.

- Implement **firewall rules** to limit network exposure and prevent lateral movement.

5. Monitoring, Logging, and Detection

- Enable **detailed logging** for SSH, sudo, and system events.
- Integrate logs with a **SIEM platform** for real-time monitoring and alerting.
- Monitor for suspicious behavior and anomalous access patterns.

6. Application and Dependency Security

- Run npm audit or yarn audit to scan and fix Node.js dependencies.
- Regularly test applications with **dynamic and static analysis tools** (DAST/SAST).
- Use a **Web Application Firewall (WAF)** to block known web-based attacks.

7. Backup and Recovery

- Schedule regular **backups** of configurations and critical data.
- Test **disaster recovery** and **backup restoration** procedures periodically.

8. Security Awareness and Governance

- Conduct regular **security training** for system administrators and developers.
- Maintain updated **security policies**, procedures, and asset documentation.
- Perform periodic **vulnerability assessments** and **penetration tests** to stay proactive.

TASK 2: WEB APPLICATION VULNERABILITY SCAN

Scan Overview

Scan Type: Web Vulnerability Scan
Target: nginx (nginx/1.15.5)

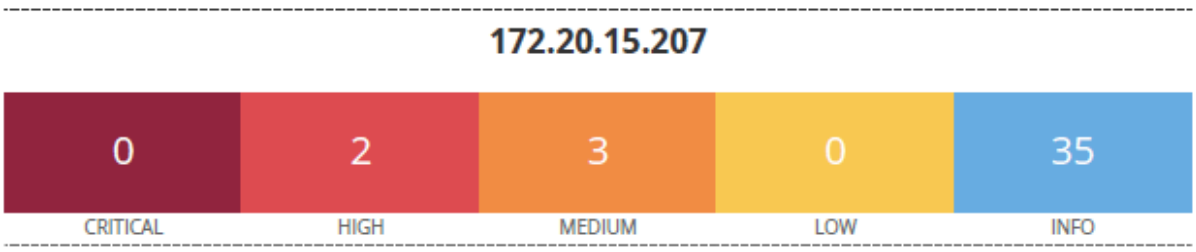
1. Initiate Web Application Scan

- Launched **Nessus** and navigate to the **New Scan** section.
- Selected the **"Web Application Tests"** scan template from the list.
- Set the **target IP address or domain** of the web application running on the Linux server.
- Used **default scan settings**.

2. Execute the Scan

- Clicked **Save** and then **Launch** the scan.
- Monitor the progress until the scan completes.
- After completion, review the list of discovered vulnerabilities in the **Results** tab.

Nginx Vulnerabilities Breakdown



Vulnerabilities

1. **nginx 0.6.x < 1.20.1 1-Byte Memory Overwrite RC:**
   - **Description**: A security issue in the **nginx** resolver was identified, which might allow an unauthenticated remote attacker to cause a 1-byte memory overwrite by using a specially crafted DNS response. This could result in a worker process crash or, potentially, arbitrary code execution.

- **Exploit Method**: Remote attackers can exploit this vulnerability by sending specially crafted DNS responses to the affected nginx server, leading to a 1-byte memory overwrite.
- Nessus ID: 150154
- Severity: High
- CVE: CVE-2021-23017
- Exploit Available: Yes
- Exploit Ease: High (*Oracle, 2022*)
- Patch Publication Date: May 25, 2021
- Disclosure Date: May 25, 2021
- NIST Description: NIST has assigned CVE-2021-23017 with a base score of 7.7, indicating a high severity vulnerability that allows remote code execution.
- It is recommended to upgrade to nginx version 1.20.1 or later to mitigate this vulnerability.

2. **nginx 1.9.5 < 1.16.1 / 1.17.x < 1.17.3 Multiple Vulnerabilities:**
   - **Description**: Versions of nginx prior to 1.16.1 or 1.17.x prior to 1.17.3 are affected by multiple denial of service vulnerabilities in the HTTP/2 protocol stack. These vulnerabilities arise from improper handling of exceptional conditions, allowing unauthenticated, remote attackers to cause a denial of service by manipulating window sizes, stream priorities, or sending zero-length header names and values.
   - **Exploit Method**:
     - Manipulating the window size and stream priority of large data requests.
     - Creating multiple request streams and continually changing their priorities.
     - Sending headers with zero-length names and values. These actions can lead to a denial of service condition.
   - **Nessus ID**: **150154**
   - **Severity**: High
   - **CVE**: CVE-2019-9511, CVE-2019-9513, CVE-2019-9516
   - Exploit Available: Yes
   - Exploit Ease: High

- **Patch Publication Date**: August 16, 2019
- **Disclosure Date**: August 16, 2019
- **NIST Description**: NIST has assigned CVE-2019-9513 with a base score of 7.8, indicating a high severity vulnerability that allows remote code execution.

It is recommended to upgrade to nginx version 1.16.1 or later to mitigate these vulnerabilities.

3. **nginx 1.x < 1.14.1 / 1.15.x < 1.15.6 Multiple Vulnerabilities**
   - **Description**: Versions of nginx prior to 1.14.1 or 1.15.x prior to 1.15.6 are affected by multiple vulnerabilities.
     - **CVE-2018-16843**: An issue in the 'ngx_http_v2_module' module that can lead to excessive memory usage.
     - **CVE-2018-16844**: An issue in the 'ngx_http_v2_module' module that can cause excessive CPU usage
     - **CVE-2018-16845**: An issue in the 'ngx_http_mp4_module' module that can result in worker process crashes or memory disclosure.
   - **Exploit Method**: Remote attackers can exploit these vulnerabilities by sending specially crafted requests to the affected nginx server, leading to resource exhaustion or potential information disclosure.
   - **Nessus ID**: 118956
   - **Severity**: Medium
   - **CVE**: CVE-2018-16843, CVE-2018-16844, CVE-2018-16845
   - **Exploit Available**: No known exploits are available
   - **Exploit Ease**: Exploits are not publicly available
   - **Patch Publication Date**: November 6, 2018
   - **Disclosure Date**: January 6, 2018
   - **NIST Description**: NIST has assigned CVE-2018-16844 with a base score of 7.5, indicating a high severity vulnerability that allows remote attackers to cause a denial of service via resource exhaustion.

To mitigate these vulnerabilities, it is recommended to upgrade to nginx version 1.14.1 or later.

4. **nginx < 1.17.7 Information Disclosure:**
   - **Description**: The installed version of nginx is prior to 1.17.7, which is affected by an information disclosure vulnerability.
   - **Exploit Method**: Attackers can exploit this vulnerability by analyzing the Server response header to determine the nginx version, potentially aiding in further attacks.
   - **Nessus ID**: 134220
   - **Severity**: Medium
   - **CVE**: CVE-2019-20372
   - **Exploit Available**: Yes
   - **Exploit Ease**: Exploits are available
   - **Patch Publication Date**: December 24, 2019
   - **Disclosure Date**: December 24, 2019
   - **NIST Description**: NIST has assigned CVE-2019-20372 with a base score of 4.3, indicating a medium severity vulnerability that allows information disclosure.

   To mitigate this vulnerability, it is recommended to upgrade to nginx version 1.17.7 or later.

Remediation and Patch Status

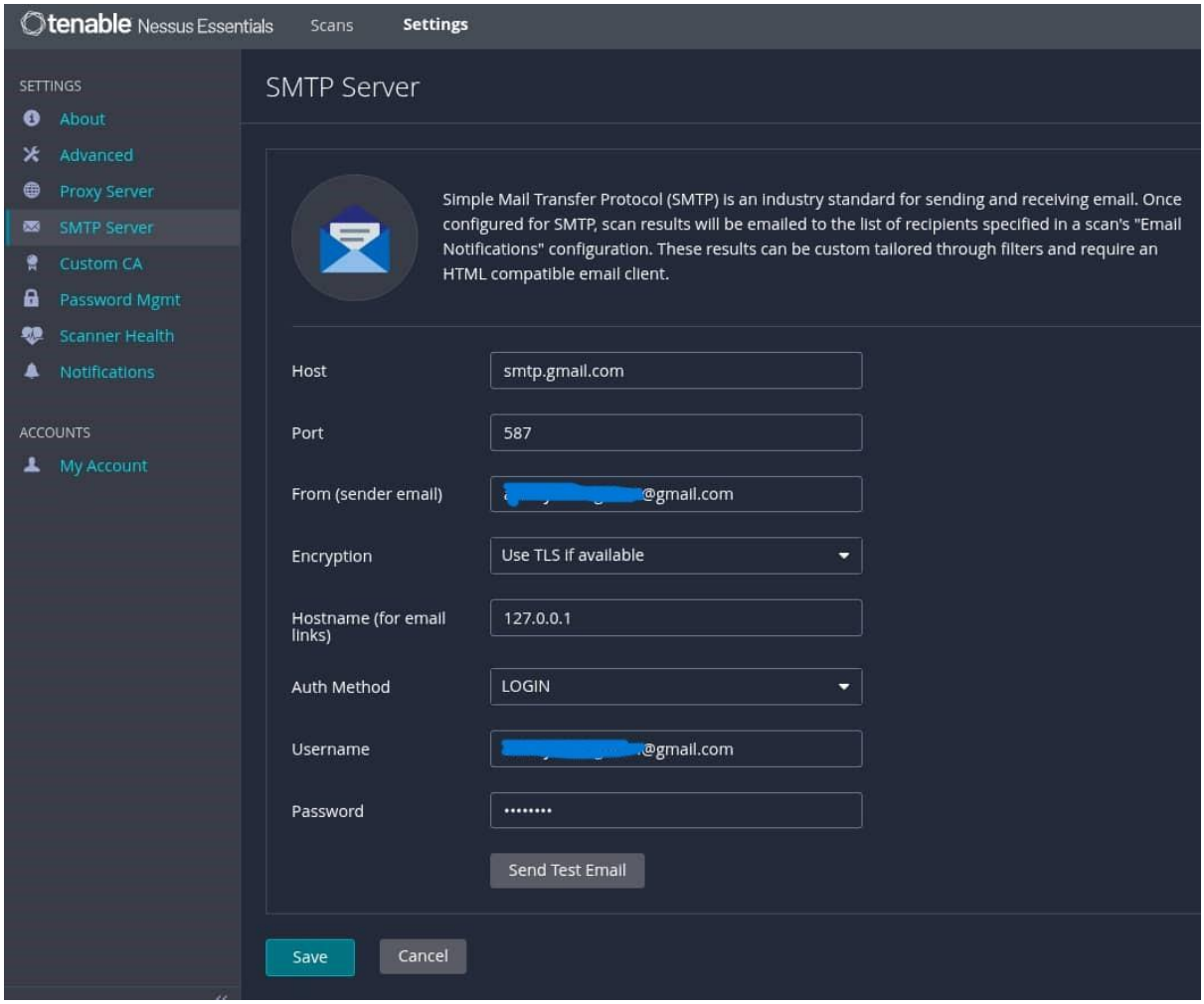
Vulnerability	Severity	Patch Status	Recommended Action
CVE-2021-23017	High	Patched	Upgrade NGINX to version 1.20.1 or later
CVE-2019-9511, CVE-2019-9513, CVE-2019-9516	High	Patched	Upgrade to NGINX version 1.16.1 or later
CVE-2018-16843, CVE-2018-16844, CVE-2018-16845	Medium	Patched	Update to NGINX version 1.14.1 or later

CVE-2019-20372	Medium	Patched	Upgrade to NGINX version 1.17.7 or later

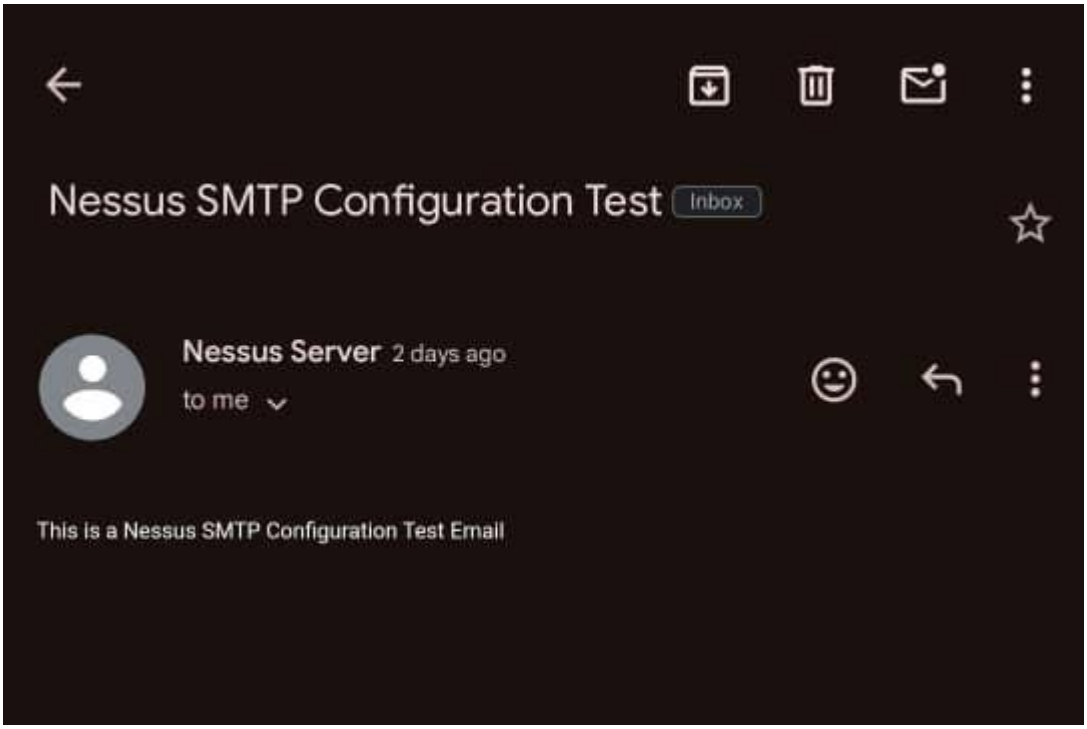
TASK 4: EMAIL REPORT AUTOMATION

Action Taken: Configured Nessus to send email reports after each scan.

- Navigated to “Settings > Notifications.
- Set recipient email to: user@email.com
- Enabled daily scan result summaries.
- Configured SMTP server credentials for internal email gateway.



(***Mail Configuration***)


Nessus SMTP Configuration Test
Inbox
Nessus Server 2 days ago
to me
This is a Nessus SMTP Configuration Test Email

(***Test Mail***)

Web Application Security Posture Summary

Overview

A Nessus Web Application Scan was conducted on the Linux-hosted web server. The scan revealed multiple critical and medium-risk vulnerabilities, primarily linked to outdated versions of NGINX and misconfigurations that expose sensitive information. Most of the issues affect the HTTP/2 implementation and version disclosure.

Tools Used

- Kali Linux
- Nessus by Tenable
- Terminal
- Code / Text Editor
- Vulnerable Web Server

Methodology

1. Target Scoping

- Identified and defined the target system: a Linux-based web server hosting the application.
- Ensured the target IP address and scope were in line with project or task instructions.

2. Scan Configuration

- Logged into Nessus and selected the **"Web Application Tests"** scan template.
- Input the **target IP address** of the server in scope.
- Used **default scan settings** with **no authentication** configured, as per assessment instructions.

3. Scan Execution

- Saved the scan and initiated it by launching the scan from the Nessus dashboard.
- Monitored the scan progress to ensure it completed successfully.

4. Result Analysis

- Reviewed the results from the **Results** tab in Nessus once the scan completed.

- Prioritized vulnerabilities based on:
    - Severity (Critical, High, Medium)
    - CVE exposure
    - Exploit availability and ease
    - Potential impact on system availability and data confidentiality.

5. Vulnerability Classification

- Focused on key findings such as:
    - Outdated NGINX versions
    - HTTP/2 DoS vulnerabilities
    - Information disclosure via headers
- Correlated Nessus plugin IDs with official CVE databases for verification.

6. Remediation Planning

- Documented recommended actions for each major finding, including:
    - Software upgrades
    - Configuration changes
    - Mitigation options where upgrades were not immediately possible

Key Vulnerabilities Identified

1. **Outdated NGINX Versions**
    - **CVE-2021-23017**: 1-byte memory overwrite via DNS resolver
    - **CVE-2019-9511, CVE-2019-9513, CVE-2019-9516**: Denial of Service via HTTP/2 frame manipulation
    - **CVE-2018-16843, CVE-2018-16844, CVE-2018-16845**: Denial of Service vulnerabilities in HTTP/2 due to memory and CPU resource misuse
    - **CVE-2019-20372**: Server version disclosure via HTTP response headers
2. **Multiple Plugin Findings from Nessus**

- Plugin IDs: 150154, 127907, 118956, 134220, among others, confirm that the current web server is running a vulnerable and unsupported NGINX release.

Recommended Improvements

1. **Upgrade NGINX Immediately**: Many of the detected vulnerabilities (including CVE-2021-23017, CVE-2019-9511, CVE-2018-16843, etc.) exist because the server is running outdated versions of NGINX. These versions are known to have flaws that can be exploited to crash the server (Denial of Service), leak memory, or be used in targeted attacks. Upgrading to the latest stable release (at least 1.20.1) ensures that all known vulnerabilities are patched.
2. **Disable or Harden HTTP/2 if Not Required**: Several vulnerabilities specifically affect the HTTP/2 protocol, which introduces additional complexity and can be exploited in ways not possible with HTTP/1.1. If your application doesn't strictly require HTTP/2, disabling it reduces the potential attack surface. If HTTP/2 must be used, ensure the server is running a version where all related vulnerabilities are patched and monitor traffic for unusual behaviour.

3. **Applying Patch Management and Monitoring**: The presence of multiple outdated packages indicates the lack of an automated or consistent patch management process. Implementing a schedule for regularly checking and applying updates (including NGINX and system libraries) will help ensure that future vulnerabilities are addressed promptly and the environment remains secure.

4. Rescan Post-Remediation: After applying all necessary fixes, rerun the Nessus scan to confirm that previously flagged vulnerabilities have been resolved.

Conclusion

The server is currently exposed to several high and medium-severity vulnerabilities due to the use of outdated NGINX versions and improper configuration. Immediate remediation actions, including software upgrades and configuration adjustments, are required to reduce the attack surface and improve the server's overall security posture.

References

BTNHD. (2024, July). *How to install Nessus Essentials in Kali Linux!* [Video].YouTube. https://youtu.be/yO3QiRMDDes

Tenable, Inc. (n.d.). *Nessus Essentials*. Tenable. Retrieved April 10, 2025, from https://www.tenable.com/tenable-for-education/nessus-essentials