Project Report: Identifying and Mitigating Network Vulnerabilities using Nessus

Submitted by

Archit Yadav

CYBER SECURITY INTERNSHIP AT

EXTION INFOTECH

2024

Index

About Me

My name is Archit Yadav. I am currently pursuing a Bachelor of Technology [B.Tech] in Electronics and Communication Engineering at Pranveer Singh Institute of Technology. During my academic journey, I have developed a strong foundation in programming languages such as C++ and Python and excelled in core subjects related to my field. I hold a design patent for a project based on Free Space Optics (FSO) and won a cash prize of ₹10,000 at my college's Tech-Xpo event. Additionally, I have completed research in wireless communication on the topic "Asymptotic Analysis of Kappa-Mu Shadowed Model with Application to Error Probability," verified through MATLAB simulations, and the research will be published soon.

I am also deeply passionate about Ethical Hacking and Cybersecurity. My interest began in class 10, where I started exploring tools like Kali Linux and Termux on mobile devices.

How to Reach me:-

Linkden:- https://www.linkedin.com/in/archit-yadav-42491321a

Gmail:- archityadav0007@gmail.com

Instagram:- https://www.instagram.com/yaduvanshi_archit

Leetcode:- **https://leetcode.com/u/ArchitYadav**

Introduction

Project Objective:

The primary objective of this project is to leverage Tenable Nessus to identify and remediate network vulnerabilities. This report outlines the complete process, encompassing environment setup, vulnerability scanning, result analysis, and the implementation of mitigation strategies to enhance overall network security.

Background:

Network vulnerabilities pose significant risks, including data breaches, unauthorized access, and denial-of-service attacks. Identifying these vulnerabilities is crucial for ensuring a secure and robust network infrastructure. Tenable Nessus has become a pivotal tool in vulnerability assessment, equipping organizations with powerful capabilities to detect and mitigate security weaknesses effectively.

Scan Objectives:

- Identify a minimum of five vulnerabilities within the network.
- Deliver comprehensive reports detailing each vulnerability.
- Offer actionable recommendations to mitigate the identified vulnerabilities.

Overview of Nessus

1. Historical Development and Evolution

The journey of Nessus began in the late 1990s when Renaud Deraison, motivated by the need for better vulnerability assessment tools, created the initial version of Nessus. Originally an open-source project, Nessus provided a platform for security professionals to identify vulnerabilities in their systems. Its open-source nature allowed for community contributions and rapid updates, fostering a collaborative approach to cybersecurity.
In 2005, Tenable, Inc. acquired Nessus, transitioning it from an open-source project to a commercial product. This transition introduced new features, enhanced functionality, and a more structured support model. Despite this shift, Tenable continued to maintain a free version of Nessus, known as Nessus Essentials, for personal and educational use.

2. What is Nessus?

Nessus is a widely-used vulnerability scanning tool developed by Tenable, Inc. It is designed to identify security vulnerabilities in computer systems, networks, and applications. By detecting these vulnerabilities, Nessus helps organizations mitigate potential security risks before they can be exploited by attackers. Originally developed by Renaud Deraison in 1998, Nessus has evolved into a robust tool used globally by security professionals to enhance their cybersecurity posture.

3. Key Features of Nessus

I. **Vulnerability Scanning**
Nessus performs comprehensive scans of systems and networks to identify known vulnerabilities. It uses a continually updated database of vulnerability signatures and threat intelligence to detect security issues across various components, including operating systems, applications, and network devices.

II. **Extensive Plugin Library**
Nessus utilizes a large library of plugins that are regularly updated to

address new vulnerabilities and threats. Each plugin is designed to check for specific vulnerabilities or configuration issues. This extensive library ensures that Nessus can provide thorough coverage of known security issues.

III. **Customizable Scan Policies**
Users can create and customize scan policies to tailor assessments to their specific needs. Nessus allows users to define scan parameters, select specific plugins, and configure settings to focus on particular areas of interest or compliance requirements.

IV. **Detailed Reporting**
After completing a scan, Nessus generates detailed reports that provide insights into identified vulnerabilities. These reports include information on the nature and severity of each vulnerability, along with recommendations for remediation. Reports can be customized to include specific details and formats.

V. **Integration and Automation**
Nessus integrates with a variety of security tools and platforms, such as security information and event management (SIEM) systems, ticketing systems, and network management tools. It also supports automation through its API, allowing organizations to incorporate vulnerability scanning into their continuous security monitoring and incident response processes.

VI. **User-Friendly Interface**
Nessus features a user-friendly web-based interface that simplifies the configuration and management of scans. The interface provides easy access to scan settings, results, and reports, making it accessible for users with varying levels of expertise.

4. Advantages of Nessus

- **Comprehensive Vulnerability Detection**
  Nessus provides a thorough assessment of systems and networks by identifying a wide range of vulnerabilities. Its extensive plugin library

and regular updates ensure that it can detect both common and emerging threats.

- **User-Friendly Interface**
  The web-based interface of Nessus is intuitive and easy to navigate. It simplifies the process of configuring scans, reviewing results, and generating reports, making it accessible to users with varying levels of expertise.
- **Customizability**
  Nessus offers flexible configuration options, allowing users to create customized scan policies and tailor assessments to their specific needs. This customization enhances the effectiveness of vulnerability management and ensures that scans align with organizational requirements.
- **Integration and Automation**
  Nessus integrates with other security tools and platforms, facilitating seamless integration into broader security operations. Its automation capabilities, through API support, enable organizations to incorporate vulnerability scanning into continuous monitoring and incident response workflows.
- **Cost-Effective**
  Nessus provides a cost-effective solution for vulnerability assessment compared to other tools in the market. Its scalability makes it suitable for organizations of all sizes, and its comprehensive features help avoid costly security breaches and incidents.
- **Regular Updates and Support**
  Tenable provides regular updates to Nessus, including new plugins and features, to address evolving threats. Additionally, users have access to support resources, including documentation, forums, and customer support, to assist with any issues or questions.

Setup Up Nessus

Download and Installation

To set up Nessus, follow these steps:

- **Download**: Visit the Tenable website to download the appropriate Nessus version for your operating system (Windows, Linux, macOS).
- **Install**: Follow the installation instructions provided for your operating system. This typically involves running an installer or using a package manager to complete the setup.

Initial Configuration

After installation, you need to configure Nessus:

- **Access the Web Interface**: Open a web browser and navigate to the Nessus web interface, typically accessible via https://localhost:8834 or the IP address of the server where Nessus is installed.
- **Create an Account**: The first time you access the interface, you will be prompted to create an administrator account. Provide the necessary information and create a secure password.
- **License Activation**: Enter your Nessus license key or select the free version (Nessus Essentials) if you are using it for personal or non-commercial use. Follow the prompts to activate the license.

Update Plugins

Nessus requires regular updates to its plugin database to stay current with the latest vulnerabilities. After the initial setup, Nessus will automatically download and update plugins. You can also manually trigger updates from the web interface.

Configuration

Create and Configure Scan Policies

- **Scan Policy Creation**: In the Nessus web interface, navigate to the "Policies" section and create a new scan policy. Define the policy settings, including the type of scan (e.g., basic network scan, web application scan), target systems, and specific plugins to use.
- **Customize Settings**: Configure additional settings such as scan schedules, authentication credentials, and advanced options based on your requirements.

Set Up and Launch Scans

- **Define Targets**: In the "Scans" section, create a new scan by specifying the target systems or network ranges. Enter relevant information, such as IP addresses or hostnames.
- **Apply Policies**: Select the scan policy you created and apply it to the scan.
- **Run the Scan**: Start the scan by clicking the appropriate option in the interface. Monitor the progress and results through the web interface.

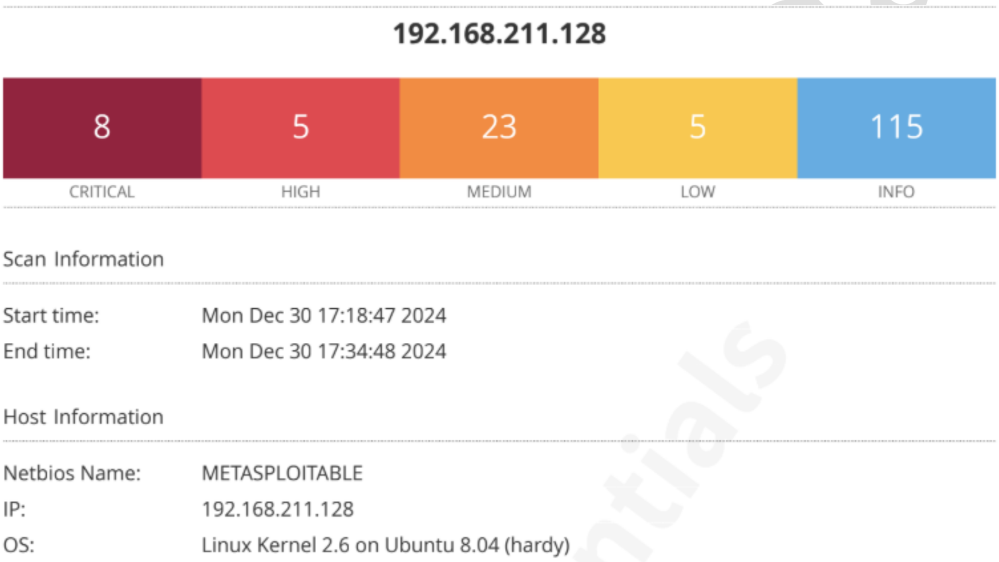
Review and Analyze Results

- **View Results**: Once the scan is complete, review the results in the "Reports" section. Nessus provides detailed information on identified vulnerabilities, including severity levels and potential impacts.
- **Remediation**: Use the recommendations provided in the reports to address and remediate identified vulnerabilities. Nessus may also offer guidance on best practices and mitigation strategies.

Mitigation Plan for Identified Vulnerabilities

Scan Summary

- **Scan Date**: 2024-12-30
- **Target**: 192.168.211.128[Metasploitable]
- **Scan Type**: Basic Network Vulnerability Scan
- **Nessus Plugin Set**: Standard



192.168.211.128

8	5	23	5	115
CRITICAL	HIGH	MEDIUM	LOW	INFO

Scan Information

Start time:	Mon Dec 30 17:18:47 2024
End time:	Mon Dec 30 17:34:48 2024

Host Information

Netbios Name:	METASPLOITABLE
IP:	192.168.211.128
OS:	Linux Kernel 2.6 on Ubuntu 8.04 (hardy)

Top 5 Vulnerabilities Overview:

134862 - Apache Tomcat AJP Connector Request Injection (Ghostcat)

Synopsis-
The remote host has a vulnerable Apache JServ Protocol (AJP) connector.

Description-
A vulnerability exists in the AJP connector that allows a remote, unauthenticated attacker to:

- Read sensitive web application files from the server.
- Exploit file upload functionalities (if enabled) to upload malicious JavaServer Pages (JSP) code, potentially achieving remote code execution (RCE).

Impact-
Exploitation of this vulnerability can compromise the confidentiality, integrity, and availability of the affected system.

References-

- Nessus Vulnerability Report - Reference 1
- Nessus Vulnerability Report - Reference 2
- Red Hat Security Advisory
- CVE-2020-1938 Details

Solution
To mitigate this vulnerability:

1. **Upgrade Apache Tomcat**
   - Update the Tomcat server to version **7.0.100**, **8.5.51**, **9.0.31**, or later.
   - Restart the server after the update to apply the changes.
2. **Secure the AJP Connector**

   - Edit the server.xml configuration file:
     - Restrict access to the AJP connector by setting address="127.0.0.1".
     - Enable authentication by setting secretRequired="true" and defining a strong secret.
   - <Connector protocol="AJP/1.3" address="127.0.0.1" secretRequired="true" secret="StrongSecret" port="8009" />

3. **Disable the AJP Connector if Unused**
   - If the AJP connector is not required, comment out or remove the connector from the server.xml file:
   - <!-- <Connector protocol="AJP/1.3" port="8009" /> -->
4. **Restrict Network Access**
   - Use a firewall or similar network control mechanism to block external access to the AJP port (default: 8009).
5. **Verify Remediation**
   - Rescan the system to ensure the vulnerability has been resolved.

Risk Factor

- **CVSS v3.0 Base Score:** 9.8 (Critical)
- **CVSS v2.0 Base Score:** 7.5 (High)

Plugin Information

- **Published:** March 24, 2020
- **Last Modified:** July 17, 2024

32314 - Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

Synopsis

The SSH host keys on the remote system are weak.

Description

The remote SSH host key was generated on a Debian or Ubuntu system affected by a critical bug in the random number generator of its OpenSSL library. This issue arose from a modification in the Debian-packaged version of OpenSSL, which removed nearly all entropy sources. As a result, any cryptographic material generated on the affected systems is highly predictable.

An attacker could exploit this vulnerability to:

- Gain access to the private portion of the SSH host key.
- Intercept and decipher encrypted communications.
- Perform a man-in-the-middle attack, compromising the integrity and confidentiality of the affected system.

References

- Nessus Plugin Details
- Additional Information
- CVE Details: CVE-2008-0166
- BID Reference: BID 29179

Solution

All cryptographic material generated on the affected system should be considered compromised and must be re-generated:

1. **Update OpenSSL**
    - Upgrade the OpenSSL library to a secure version unaffected by this bug.
2. **Re-generate Cryptographic Keys**

- Replace all SSH, SSL, and OpenVPN keys created on the system.
- Update host and user keys using the following commands:
- rm -f /etc/ssh/ssh_host_*
- dpkg-reconfigure openssh-server

3. **Distribute New Public Keys**
    - Inform any systems or users relying on the compromised keys to replace them with the newly generated ones.
4. **Verify**
    - Conduct a thorough review of the system's cryptographic configuration to ensure compliance with best practices.

Risk Factor

- **Critical**
- **CVSS v2.0 Base Score:** 10.0 (Critical)
- **CVSS v2.0 Temporal Score:** 8.3

Additional Notes

Given the critical nature of this vulnerability, it is essential to promptly update the affected system and replace compromised keys to prevent unauthorized access or data breaches.

20007 - SSL Version 2 and 3 Protocol Detection

Synopsis

The remote service encrypts traffic using a protocol with known vulnerabilities.

Description

The remote service supports connections encrypted with SSL 2.0 and/or SSL 3.0. These protocol versions have several well-documented cryptographic flaws, including:

- Insecure padding schemes when using CBC ciphers.
- Weak session renegotiation and resumption mechanisms.

These vulnerabilities allow attackers to:

- Perform man-in-the-middle (MITM) attacks.
- Decrypt sensitive communications between the service and its clients.

Although SSL/TLS has mechanisms to select the highest-supported protocol version, many web browsers implement this in a manner susceptible to downgrade attacks (e.g., the POODLE vulnerability).

NIST has declared SSL 3.0 inadequate for secure communications. Additionally, PCI DSS v3.1 mandates that SSL versions no longer meet the definition of "strong cryptography" and should not be used.

References

- Schneier SSL Paper
- Nessus Plugin Details
- OpenSSL POODLE Details
- Imperial Violet POODLE Information
- RFC 7507 - POODLE Mitigation
- RFC 7568 - Deprecating SSL 3.0

Solution

1. Disable SSL 2.0 and SSL 3.0 in the application's configuration settings.
2. Enable and enforce the use of TLS 1.2 or higher with approved and secure cipher suites.
3. Review and update the application's documentation for specific configuration steps to disable these protocols.

Risk Factor

- **Critical**
    - **CVSS v3.0 Base Score:** 9.8 (Critical)
    - **CVSS v2.0 Base Score:** 10.0 (Critical)

Plugin Information

- **Published:** 2005/10/12
- **Modified:** 2022/04/04

Addressing these vulnerabilities is critical to maintaining the integrity and confidentiality of communications and to ensure compliance with industry security standards.

46882 - UnrealIRCd Backdoor Detection

Synopsis

The remote IRC server contains a backdoor.

Description

The detected IRC server is running a version of UnrealIRCd that includes a backdoor. This vulnerability allows an attacker to execute arbitrary code on the affected host, posing a critical security risk.

References

- Full Disclosure - Advisory 1
- Full Disclosure - Advisory 2
- UnrealIRCd Security Advisory

Solution

1. Re-download the UnrealIRCd software from the official source.
2. Verify the downloaded file using the published MD5 or SHA1 checksums to ensure its integrity.
3. Re-install the software using the verified version.

Risk Factor

- **Critical**
  - **CVSS v2.0 Base Score:** 10.0 (Critical)
  - **CVSS v2.0 Temporal Score:** 8.3 (Critical)

Exploitability

- Exploitable With:
    - **CANVAS**: True
    - **Metasploit**: True

Plugin Information

- **Published:** 2010/06/14
- **Modified:** 2022/04/11

Mitigating this vulnerability promptly is essential to prevent potential exploitation and ensure the security of the affected host.

61708 - VNC Server 'password' Password

Synopsis

A VNC server running on the remote host is secured with a weak password.

Description

The VNC server on the remote host uses a weak password for authentication. Nessus successfully authenticated using the default password, "password." This vulnerability allows a remote, unauthenticated attacker to take control of the system, posing a critical security risk.

Solution

Secure the VNC service by setting a strong, complex password to prevent unauthorized access.

Risk Factor

- **Critical**
    - **CVSS v2.0 Base Score:** 10.0 (Critical)

Plugin Information

- **Published:** 2012/08/29
- **Modified:** 2015/09/24

Implementing a robust password policy is vital to mitigate this vulnerability and safeguard the system against potential exploitation.

Conclusion

1. **Project Outcomes**
   This project effectively identified and addressed several network vulnerabilities using Nessus. The detailed scanning and analysis offered crucial insights into the network's security posture, enabling focused and efficient remediation measures.
2. **Future Work**
   Future efforts should prioritize implementing a routine schedule for vulnerability assessments, continuously monitoring emerging threats, and enhancing the network's security infrastructure. A proactive approach to network security will ensure the network remains robust, secure, and resilient against potential risks.