Project Report: Identifying and Mitigating Network Vulnerabilities using Nessus

Submitted by

Himangshu Rana

CYBER SECURITY INTERNSHIP

AT

EXTION INFOTECH

2024

LinkedIn: www.linkedin.com/in/himangshurana

Nessus Report Index

About Me

I'm a passionate cybersecurity Person with a strong interest in ethical hacking and related fields. I'm constantly seeking new knowledge and experiences to stay ahead of the ever-evolving cyber threat landscape.

Contact : Linkedin

Gmail: tohimangshurana@gmail.com

Introduction

1.1 Project Objective

The main goal of this project is to use Tenable Nessus to identify and address network vulnerabilities. This report covers the entire process, including setting up the environment, conducting vulnerability scans, analyzing the results, and implementing mitigation strategies to improve the network's overall security.

1.2 Background

Network vulnerabilities can lead to data breaches, unauthorized access, and denial of service attacks. It's essential to identify these vulnerabilities to maintain a secure and strong network infrastructure. Tenable Nessus has established itself as a cornerstone in the field of vulnerability assessment, providing organizations with the tools they need to identify and address security weaknesses effectively.

1.3 Scan Objectives

- ☑ Identify at least five Vulnerabilities in the network.
- ☑ Provide detailed reports on each vulnerability.
- ☑ Provide recommendations for mitigating 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 to 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

1. **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.

2. **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.

3. **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.

4. **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.

5. **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.

6. **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 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 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

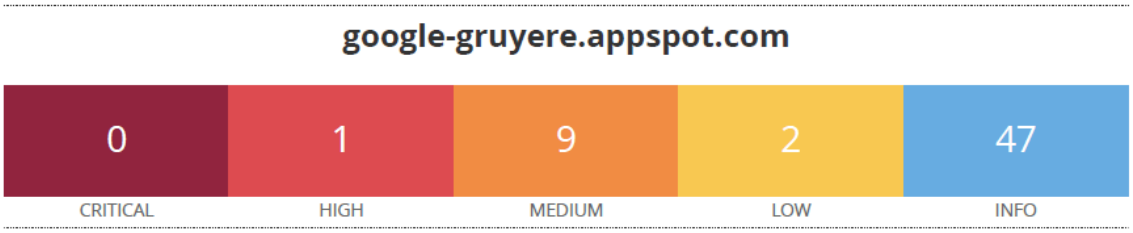
- **View Results:** Once the scan is complete, review the results in the "Reports" section. Nessus provides detailed information on identified vulnerabilities, including severity levels and potential impacts.

- **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.

Vulnerability Identification

☑Summary :

Scan Date: 2024-08-20
Target: google-gruyere.appspot.com
Scan Type: Basic Network Vulnerability Scan
Nessus Plugin Set: Standard



☑Top 5 Vulnerabilities Overview:

1. **High (7.5):**
    - **Vulnerability:** SSL Medium Strength Cipher Suites Supported (SWEET32)
    - **Plugin ID:** 42873
    - **Description:** The remote host supports the use of SSL ciphers that offer medium strength encryption. Nessus regards medium strength as any encryption that uses key lengths at least 64 bits and less than 112 bits, or else that uses the 3DES encryption suite. Note that it is considerably easier to circumvent medium strength encryption if the attacker is on the same physical network.

2. **Medium (6.5):**
    - **Vulnerability:** HSTS Missing from HTTPS Server (RFC 6797)
    - **Plugin ID:** 142960
    - **Description:** The remote web server is not enforcing HSTS, as defined by RFC 6797. HSTS is an optional response header that can be configured on the server to instruct the browser to only communicate via HTTPS. The lack of HSTS allows downgrade attacks, SSL-stripping man-in-the-middle attacks, and weakens cookie-hijacking protections.

3. **Medium (6.5):**
    - **Vulnerability:** TLS Version 1.0 Protocol Detection
    - **Plugin ID:** 104743
    - **Description:** The remote service accepts connections encrypted using TLS 1.1. TLS 1.1 lacks support for current and recommended cipher suites. Ciphers that support encryption before MAC computation, and authenticated encryption modes such as GCM cannot be used with TLS 1.1 As of March 31, 2020, Endpoints that are not enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.
4. **Medium (5.3):**
    - **Vulnerability:** SSL/TLS Protocol Initialization Vector Implementation Information Disclosure Vulnerability (BEAST)
    - **Plugin ID:** 58751
    - **Description:** The SSL/TLS protocol used by the remote server is susceptible to the BEAST attack, a known vulnerability that exploits the way SSL 3.0 and TLS 1.0 handle block cipher encryption. The vulnerability occurs due to a flaw in the implementation of the Cipher Block Chaining

(CBC) mode, which allows an attacker to decrypt portions of encrypted data (e.g., HTTP cookies or session tokens) by conducting a man-in-the-middle attack and injecting malicious JavaScript into the victim's browser.

5. **Medium (4.3):**
    - **Vulnerability:** CGI Generic HTML Injections
    - **Plugin ID:** 49067
    - **Description:** The web application running on the server contains a CGI script that is vulnerable to generic HTML injection attacks. This vulnerability allows an attacker to inject arbitrary HTML or JavaScript into the web page, which may be executed by the user's browser. This can lead to various attacks, such as defacement, redirection, or even cross-site scripting (XSS), depending on the context of the injection.

Mitigation Plan for Identified Vulnerabilities

1. SSL Medium Strength Cipher Suites Supported (SWEET32)

☑ **Steps for Remediation:**

- **Identify Affected Servers:**
  - Locate all servers using SSL/TLS that support 3DES or other medium-strength cipher suites.
- **Update SSL/TLS Configuration:**
  - Disable all 64-bit block ciphers, such as 3DES, in the server's SSL/TLS configuration.
  - Configure the server to support only strong cipher suites, such as those using AES with a minimum key size of 128 bits (e.g., AES-GCM).
- **Test the New Configuration:**
  - Use tools like SSL Labs' SSL Test or Nessus to verify that medium-strength ciphers are no longer supported.
- **Roll Out the Changes:**
  - Apply the updated configurations across all affected servers.

☑ **Estimated Timeline:** 1 week

☑ **Required Resources:**

✓ Access to SSL/TLS configuration files.

✓ SSL testing tools.

✓ Network administrator.

2. HSTS Missing from HTTPS Server (RFC 6797)

☑ **Steps for Remediation:**

1. **Review HTTPS Configuration:**
   - Ensure all content is served over HTTPS.
   - Identify all subdomains that should be included in the HSTS policy.
2. **Enable HSTS:**
   - Add the following header to the HTTPS server configuration:

     [Strict-Transport-Security: max-age=31536000;includeSubDomains; preload]
   - Test the configuration to ensure it is working correctly.
3. **Monitor and Adjust:**
   - Monitor for any issues, particularly with legacy systems that may not fully support HSTS.
4. **Submit for Preload (Optional):**
   - If the domain is ready, submit it to the HSTS preload list for browsers.

☑ **Estimated Timeline:** 2 weeks

☑ **Required Resources:**

- Web server configuration access.
- Web developers or administrators.
- Testing tools (e.g., browser developer tools).

3. TLS Version 1.0 Protocol Detection

☑ **Steps for Remediation:**

1. **Identify Servers Using TLS 1.0:**
    - Audit the network to identify servers and services that still support TLS 1.0.
2. **Update SSL/TLS Protocols:**
    - Disable support for TLS 1.0 and 1.1 in the server's SSL/TLS configuration.
    - Ensure the server supports TLS 1.2 or TLS 1.3.
3. **Test Compatibility:**
    - Test all client connections and applications to ensure compatibility with the updated protocols.
4. **Deploy Updates:**
    - Apply changes to all identified servers and services.

☑ **Estimated Timeline:** 3 weeks

☑ **Required Resources:**

- SSL/TLS configuration access.
- Compatibility testing tools.
- Network administrator.

4. SSL/TLS Protocol Initialization Vector Implementation Information Disclosure Vulnerability (BEAST)

☑ **Steps for Remediation:**

1. **Disable SSL 3.0 and TLS 1.0:**
   - Follow the steps outlined in the mitigation for TLS 1.0 to disable both SSL 3.0 and TLS 1.0.
2. **Reconfigure Cipher Suites:**
   - If TLS 1.0 must be retained for compatibility, reconfigure the server to prioritize RC4 (despite its own weaknesses) over CBC-mode ciphers for TLS 1.0. However, it's strongly recommended to upgrade to TLS 1.2 or higher.
3. **Implement TLS 1.2/1.3:**
   - Ensure TLS 1.2 or 1.3 is used, as these versions are not vulnerable to the BEAST attack.
4. **Test the Configuration:**
   - Validate that the server is no longer vulnerable using tools like SSL Labs or Nessus.

☑ **Estimated Timeline:** 2 weeks

☑ **Required Resources:**

- SSL/TLS configuration access.
- Testing tools.
- Network administrator.

5. CGI Generic HTML Injections

☑ **Steps for Remediation:**

1. **Identify Vulnerable CGI Scripts:**
   - Locate and review all CGI scripts on the server for input handling.
2. **Sanitize Input:**
   - Implement proper input validation and sanitization in CGI scripts to prevent HTML injection.
   - Use libraries or frameworks that handle input sanitization.
3. **Test for Vulnerabilities:**
   - Perform security testing using tools like Burp Suite to ensure that the vulnerability is fixed.
4. **Deploy Updated Scripts:**
   - Replace the vulnerable scripts with the updated versions.
5. **Monitor and Audit:**
   - Regularly audit the scripts and monitor logs for any signs of injection attempts.

☑ **Estimated Timeline:** 3 weeks

☑ **Required Resources:**

- Web developers familiar with CGI scripts.
- Security testing tools (e.g., Burp Suite).
- Code review tools.

Conclusion

1. Project Outcomes

This project successfully identified and mitigated numerous network vulnerabilities using Nessus. The comprehensive scanning and analysis provided valuable insights into the security posture of the network, enabling targeted and effective remediation efforts.

2 Future Work

Future work includes establishing a regular vulnerability assessment schedule, continuous monitoring for new threats, and ongoing improvements to the network security infrastructure. Adopting a proactive approach to network security will help maintain a robust and resilient network environment.