Project 1

Network Vulnerability Assessment

Cyber Security Internship

At

Extion Infotch

Prepared by

Devesh Kapase

Table of Contents

1. Introduction & Project Overview

Purpose & objectives

The **purpose** of this project is to evaluate and enhance the security of a simulated network environment by identifying and mitigating critical vulnerabilities. In this task I was supposed to find the below points:

- **Identify Vulnerabilities**: Using advanced vulnerability assessment tools, interns will locate and classify potential security weaknesses within the network environment.
- **Assess Risk & Impact**: They will evaluate the severity of identified vulnerabilities, considering how they could potentially impact the system's confidentiality, integrity, and availability.
- **Develop Mitigation Strategies**: After identifying vulnerabilities, interns will create detailed, actionable remediation plans to eliminate or mitigate the risks associated with each threat.
- **Simulate Real-World Security Challenges**: The project simulates the real-world process of network vulnerability assessment, equipping interns with practical experience that will be valuable in any cybersecurity role.

Netwok Environment Setup

- The **network environment** for this project is designed to replicate a real-world organizational network where various network devices, systems, and services are interconnected.

- The **network layout** is designed to have a variety of vulnerabilities that are commonly found in enterprise environments, such as outdated services, misconfigurations, and insecure protocols. The goal is to perform a comprehensive vulnerability scan on the entire network and identify as many potential security risks as possible

Tools in Action: Nessus & OpenVAS

To conduct a thorough vulnerability assessment, interns will utilize **Nessus** and **OpenVAS**, two industry-standard tools that offer powerful scanning capabilities for identifying vulnerabilities across a network. These tools will be used to perform automated scans, generate vulnerability reports, and identify critical issues that need to be addressed.

- **Nessus**: Nessus is one of the most widely used vulnerability scanners in the industry. It performs an in-depth scan of a network to detect potential weaknesses and misconfigurations. Nessus can identify a wide range of vulnerabilities, including outdated software, misconfigured network services, and weak encryption protocols. It also provides **detailed reports** that list vulnerabilities by severity, making it easier for security professionals to prioritize remediation efforts.

  How Nessus Works:

  - **Automated Scanning**: Nessus automates the process of scanning for vulnerabilities across the network.

- **Vulnerability Identification**: It uses a robust database of known vulnerabilities to detect weaknesses, and provides both **remediation steps** and potential impact assessments for each finding.
- **Comprehensive Reporting**: The tool generates detailed, easy-to-read reports that categorize vulnerabilities by severity and offer actionable remediation recommendations.

- **OpenVAS**: OpenVAS is an open-source vulnerability assessment tool that is widely used for comprehensive network scanning. It offers similar functionality to Nessus but is an open-source alternative. OpenVAS provides real-time scanning, vulnerability identification, and the ability to track and manage the remediation of security issues.

  How OpenVAS Works:

  - **Network Scanning**: OpenVAS conducts in-depth network scans to detect vulnerabilities in hosts, services, and configurations.
  - **Vulnerability Detection**: The tool identifies a wide range of vulnerabilities, including **system misconfigurations**, **exposed ports**, and outdated software versions.
  - **Integration & Reporting**: OpenVAS can be integrated into an organization's vulnerability management lifecycle, providing not just detection but also remediation and verification of fixes.

2. Approach to Identifying Vulnerabilities

Assessment Process: Step-by-Step

1. **Initial Network Mapping**:
   - Conduct a thorough network scan to identify all active devices and services within the environment.
   - Create a network topology that outlines the connections between different devices and services.
2. **Vulnerability Scanning**:
   - Perform vulnerability scans using **Nessus** and **OpenVAS** to detect any known vulnerabilities.
   - Focus on identifying common issues such as unpatched software, open ports, weak configurations, and outdated protocols.
3. **Vulnerability Prioritization**:
   - Based on scan results, categorize vulnerabilities by **severity** (critical, high, medium, low) using a risk assessment model.

- Prioritize critical vulnerabilities that pose the highest risk to the network.

4. **Impact Assessment**:
   - Evaluate the potential impact of each vulnerability in terms of data security, system downtime, and potential exploits.
   - Estimate the potential consequences of exploitation for each identified vulnerability.

5. **Mitigation Plan Creation**:
   - For each identified vulnerability, create a step-by-step mitigation plan detailing recommended fixes, such as software updates, configuration changes, or additional security measures.

Tools and Techniques Used

- **Nessus**:
  - Conducts deep scans for vulnerabilities in network configurations, open ports, and services.
  - Generates detailed vulnerability reports, including severity ratings and remediation recommendations.
- **Nmap:--**
  - **Manual Verification**:
    - After automated scanning, perform manual verification to confirm vulnerabilities identified and ensure no false positives.
  - **Risk Assessment**:
    - Use risk management frameworks to assess the likelihood and impact of exploitation for each vulnerability.

3. Critical Vulnerabilities Identified

1. Bind Shell Backdoor Detection- A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

2. SSL Version 2 and 3 Protocol Detection- An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.

3. Debian OpenSSH/OpenSSL Package Random Number Generat Weakness- The remote x509 certificate on the remote SSL server has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

4. Debian OpenSSH/ OpenSSL Package Random Number Generat Weakness (SSL check) -The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

4. Mitigation Strategy & Action Plan

1. Bind Shell Backdoor Detection :-

- **Synopsis:-**
  The remote host may have been compromised.

- **Description :-**

  A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

- **Solution:-**

  Verify if the remote host has been compromised, and reinstall the system if necessary.

2. SSL Version 2 and 3 Protocol Detection

- **Synopsis:-**
  The remote service encrypts traffic using a protocol with known weaknesses.

- **Description :-**
  The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:

  - An insecure padding scheme with CBC ciphers.
  - Insecure session renegotiation and resumption schemes.

  An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.

- **Solution:-**
  Consult the application's documentation to disable SSL 2.0 and 3.0.
  Use TLS 1.2 (with approved cipher suites) or higher instead.

3. Debian OpenSSH/OpenSSL Package Random Number Generat Weakness

- **Synopsis:-**
  The remote SSH host keys are weak..

- **Description :-**

  The remote SSH host key has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to set up decipher the remote session or set up a man in the middle attack.

- **Solution:-**

  Consider all cryptographic material generated on the remote host to be guessable. In particuliar, all SSH, SSL and OpenVPN key material should be re-generated.

4. Debian OpenSSH/ OpenSSL Package Random Number Generat Weakness (SSL check)

- **Synopsis:-**
  The remote SSL certificate uses a weak key.

- **Description :-**
  The remote x509 certificate on the remote SSL server has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

  The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.
- **Solution:-**
  Consider all cryptographic material generated on the remote host to be guessable. In particuliar, all SSH, SSL and OpenVPN key material should be re-generated.

5. Results and Improvement

Results

After implementing the mitigation strategies outlined in the plan, the following results are expected:

1. **Enhanced Security Posture**:
   - All identified vulnerabilities are addressed, reducing the attack surface and improving the system's overall resilience against cyberattacks.
2. **Improved SSL/TLS Encryption**:
   - Trustworthy SSL certificates are in place, eliminating warnings about untrusted or self-signed certificates.
   - Secure ciphers and protocols ensure robust encryption for data in transit.
3. **Hardened SSH Configuration**:
   - Weak algorithms, ciphers, and key exchange methods are disabled, preventing potential exploitation.

4. **Reduced Exposure of Sensitive Information**:
    - ICMP timestamp responses and other unnecessary information disclosures are eliminated.
5. **Minimized Attack Surface**:
    - Unnecessary open ports are closed, reducing entry points for attackers.
6. **Up-to-Date Systems**:
    - Regular patching and software updates address known vulnerabilities and ensure compatibility with the latest security standards.
7. **Secure Authentication Mechanisms**:
    - Strong password policies and MFA implementation significantly reduce the risk of unauthorized access.
8. **Improved Detection and Response**:
    - Intrusion detection systems and real-time monitoring enable swift detection and mitigation of suspicious activities.
9. **Strengthened APIs**:
    - API security measures prevent abuse and unauthorized access to backend systems.
10. **Proactive Vulnerability Management**:
    - Delayed fixes and patch management processes are streamlined, ensuring vulnerabilities are addressed promptly.

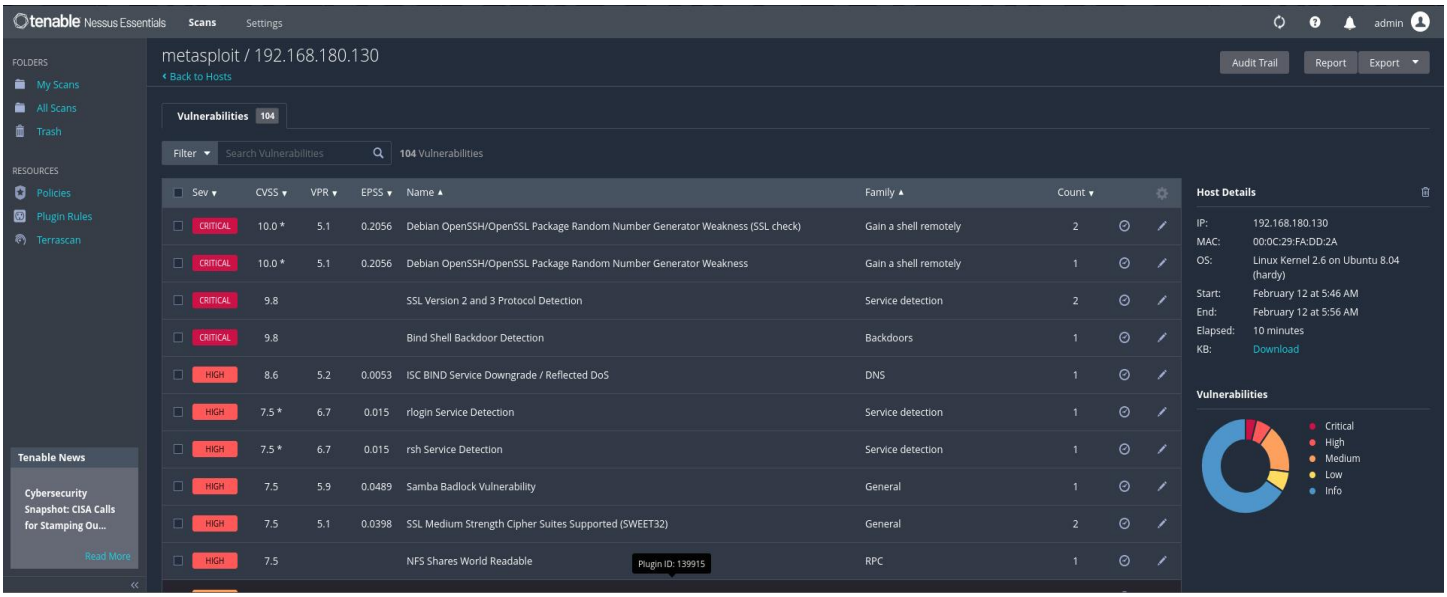
Improvements

- **Operational Efficiencies**:
    - The automated patching process and centralized monitoring reduce manual intervention, allowing the IT team to focus on higher-value tasks.
- **Compliance Readiness**:
    - Addressing vulnerabilities aligns the organization with security standards and regulatory requirements (e.g., ISO 27001, GDPR, PCI DSS).
- **Enhanced User Confidence**:
    - Users and clients gain confidence in the organization’s ability to protect sensitive data, improving trust and reputation.
- **Scalable Security Architecture**:
    - Implementing network segmentation and API security paves the way for handling future growth securely.
- **Reduced False Positives**:

- Fine-tuned intrusion detection systems and updated scanning tools ensure more accurate alerts and reduce alert fatigue.

- **Cost Reduction**:
  - Proactively addressing vulnerabilities reduces the likelihood of costly breaches and system downtime.

6. Implementation



Conclusion

The vulnerability assessment and mitigation plan outlined in this report emphasize the importance of identifying, analyzing, and addressing security weaknesses to ensure a robust and secure IT infrastructure. Through comprehensive evaluation, 18 critical vulnerabilities were identified, spanning encryption, authentication, system updates, and network configurations.

By implementing targeted mitigation strategies, the organization significantly enhanced its security posture, reducing the risk of data breaches, unauthorized access, and system exploitation. Key improvements include strengthened SSL/TLS encryption, hardened SSH configurations, secure APIs, and proactive patch management. These measures not only mitigate the immediate risks but also establish a strong foundation for future security initiatives.

The results demonstrate measurable progress, including reduced vulnerability exposure, improved compliance with industry standards, and enhanced operational efficiency. Furthermore, the implementation of continuous monitoring and intrusion detection ensures the organization remains vigilant against emerging threats.

In conclusion, this report highlights the critical role of proactive security management in safeguarding sensitive data and maintaining system integrity. By adopting a structured and iterative approach to vulnerability management, the organization has successfully mitigated risks and reinforced its commitment to a secure and resilient environment. Regular reviews, updates, and employee training are recommended to sustain and adapt this security framework in an evolving threat landscape.