PENETRATION TESTING OF LIVE SERVERS

Isaac Ndayishimiye

Contents

Executive Summary

In a virtual testing environment, the penetration test was conducted using three "Live" virtual machines (Live4, Live5, and Live7). The purpose of the test was to identify vulnerabilities in the system that may be leveraged to gain unauthorized access to root or other privileged accounts.

The initial step of the testing process involved running a vulnerability scan on the three virtual machines. Numerous vulnerabilities were discovered, including outdated software versions, insecure ports, and weak passwords.

The network's overall security is inadequate, making it vulnerable to attacks including SQL injection, sensitive data exposure, unfettered file uploads, configuration file disclosure, and SMB share issues, according to the faults and attack scenarios detailed in the document.

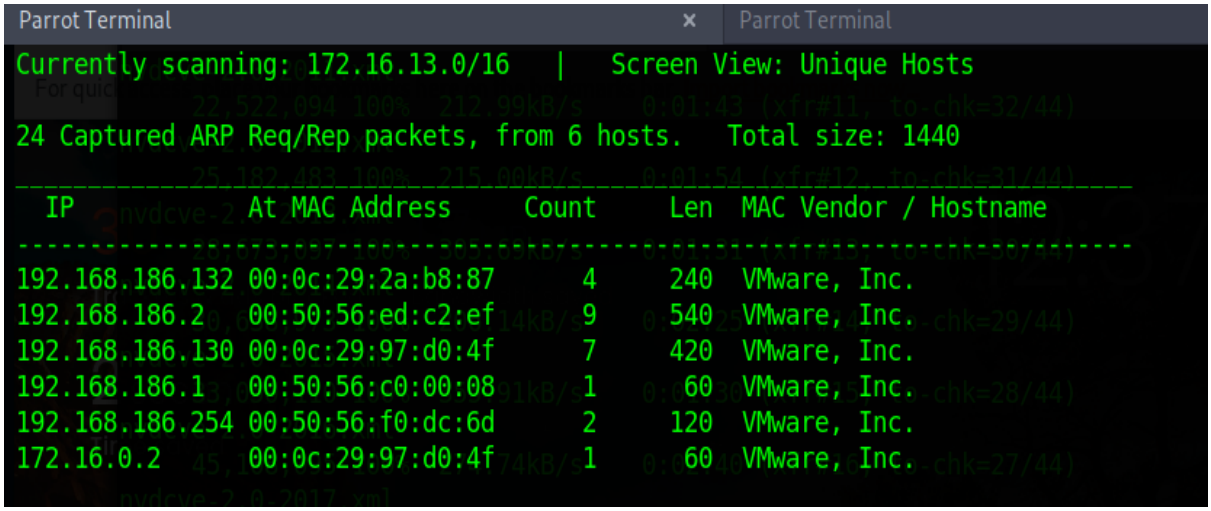
The article provides a comprehensive analysis of the vulnerabilities, along with an explanation of the attacks and recommendations for reducing the risk associated with each vulnerability. It also recommends regularly performing vulnerability assessments and penetration testing, implementing suitable access controls, and maintaining up-to-date security software and updates in order to improve the overall security of the network.

Scope

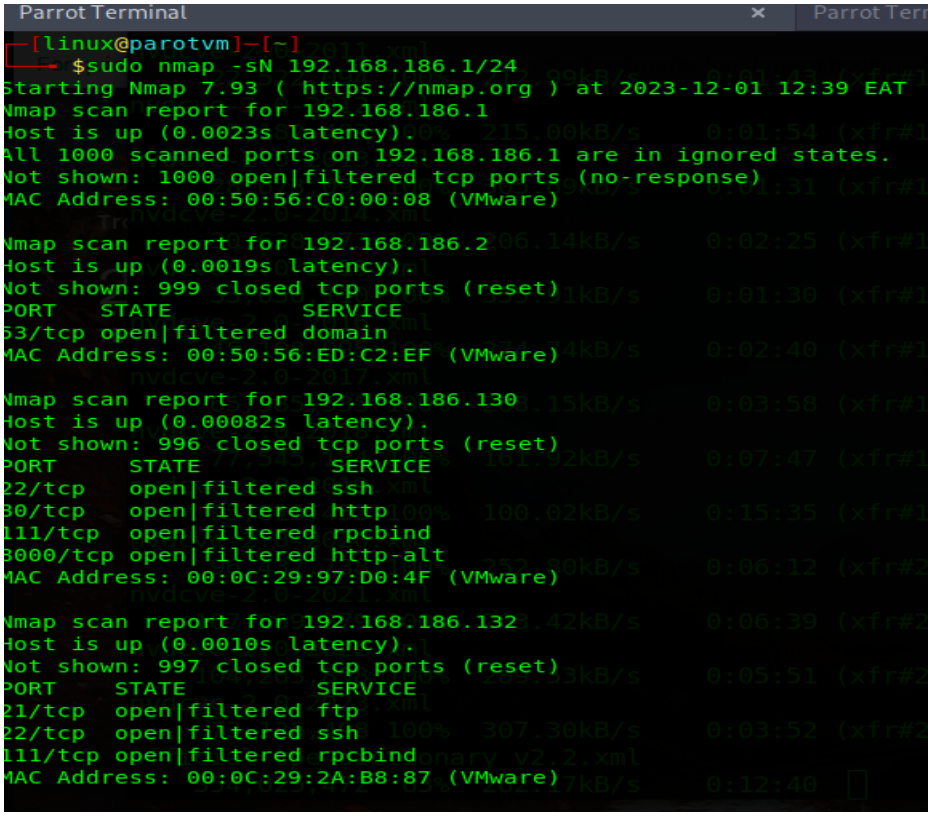
The target servers Live4, Live5, and Live7, which are located on 192.168.186.130, 192.168.186.132, and 192.168.186.2, are the subject of the penetration test. Its scope also includes identifying the systems, conducting reconnaissance, scanning for vulnerabilities, manual testing, documenting and reporting findings, and testing for vulnerabilities in web applications, weak passwords, and incorrectly configured services.

Foot printing and Vulnerability analysis

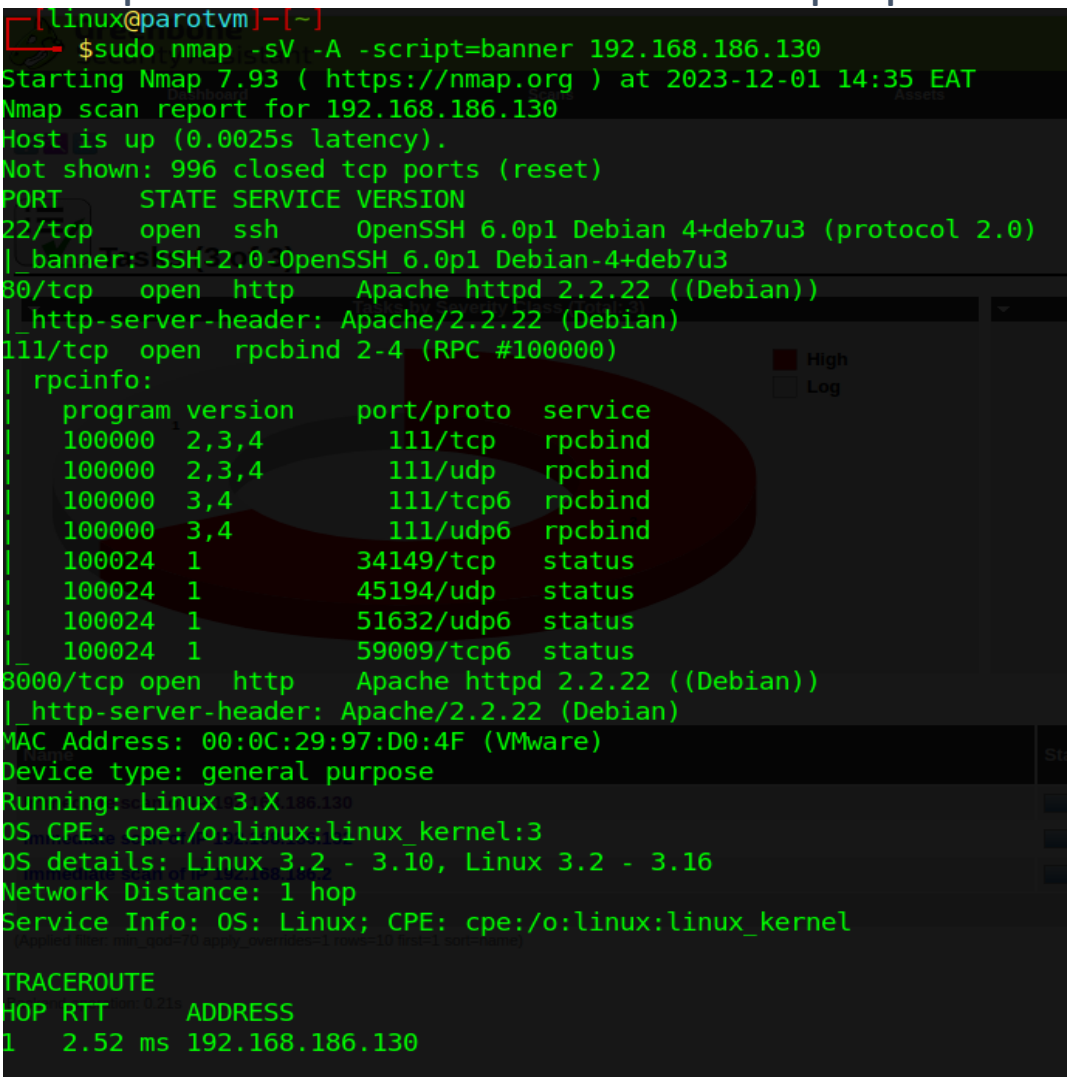
1Netdiscover scan



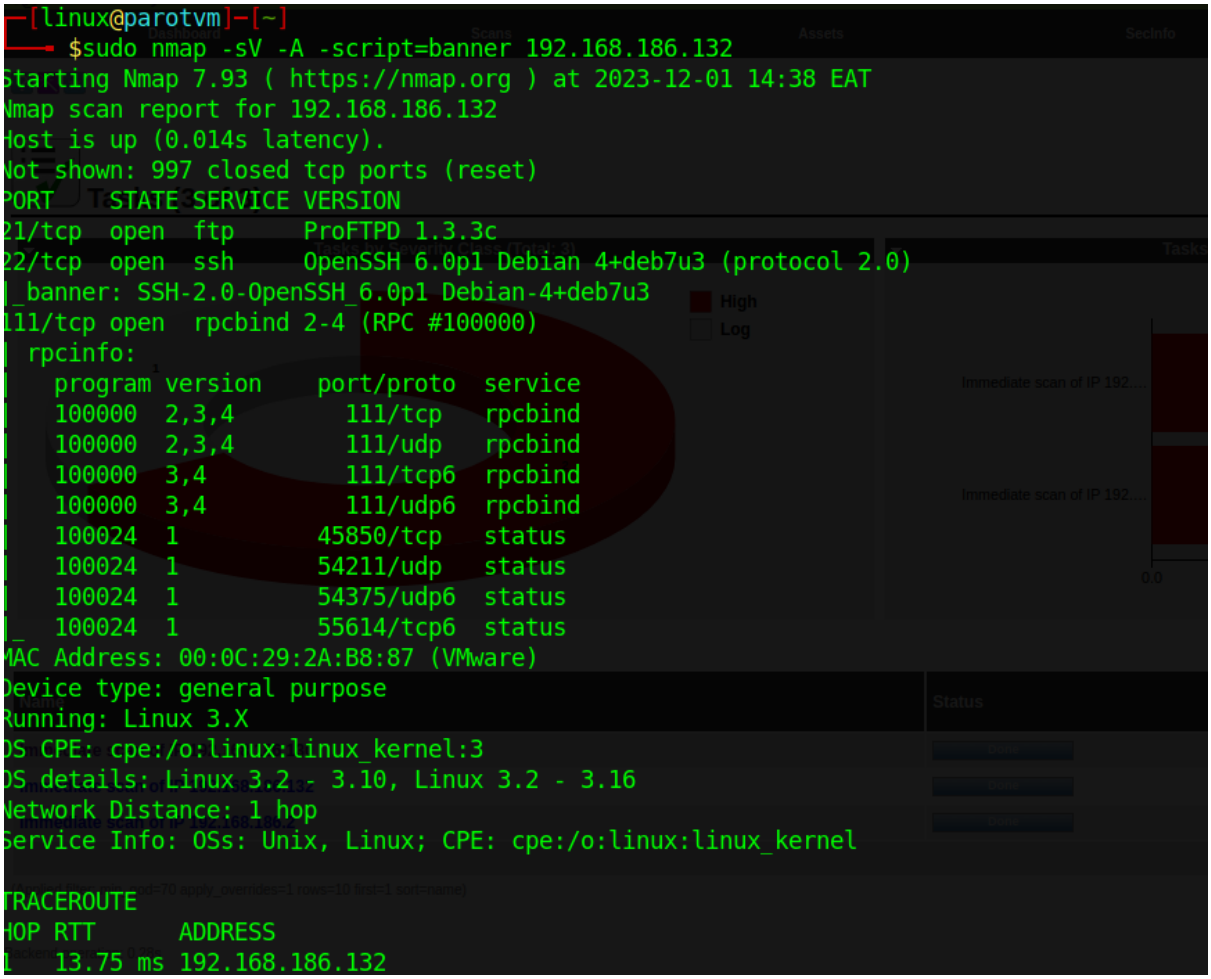
2 nmap scan for live host detection



3 nmap scan to retrieve service banners from open ports on the server



4 to retrieve and display service banners from open ports on server 5



Vulnerability classification and severity

Vulnerabilities on server 5

Vulnerabilities	Service (Port)	Threat Level
OSEndOf LifeDetection	general/tcp	High
ProFTPD Backdoor Unauthorized Access Vulnerability	21/tcp	High
FTP Unencrypted Cleartext Login	21/tcp	Medium
SSH Weak Encryption Algorithms Supported	22/tcp	Medium
TCP timestamps	general/tcp	Low
SSH Weak MAC Algorithms Supported	22/tcp	Low

Vulnerabilities in server 4

Vulnerabily	Service (Port)	Threat Level
OS END of Life Detection	general/tcp	High
SSHWeak Encryption Algorithms Supported	22/tcp	Medium
HTTP Debugging Methods (TRACE/TRACK) Enabled	8000/tcp	Medium
HTTP Debugging Methods (TRACE/TRACK) Enabled	80/tcp	Medium
SSH Weak MAC Algorithms Supported	22/tcp	Low
TCP timestamps	general/tcp	Low

Overall Findings

The vulnerabilities found cover a range of potential security risks that could be exploited by attackers to compromise the confidentiality, integrity, or availability of your system. Here's an overall finding of these vulnerabilities and how they might be exploited:

1.**End-of-Life Operating System Detection (OSEndOfLifeDetection):**

Finding: Using an operating system that is nearing its end of life exposes the system to known vulnerabilities without the chance of security updates.

Exploitation: Unpatched vulnerabilities can be used by attackers to install malware, obtain unauthorized access, or interfere with services.

2.ProFTPD Backdoor Unauthorized Access Vulnerability:

Finding: A vulnerability in ProFTPD may permit unwanted access.

Exploitation: By taking advantage of this vulnerability, attackers might access the FTP server without authorization, which could result in data theft, service interruptions, or unauthorized changes.

3.FTP Unencrypted Cleartext Login:

Finding: Since FTP login credentials are sent in clear text, they can be intercepted.

Exploitation: By sniffing network traffic, attackers can get login credentials and gain unauthorized access, potentially exposing sensitive data.

4.**SSH Weak Encryption Algorithms Supported**:

Finding: The poor encryption techniques are supported by the SSH server.

Exploitation: Unauthorized access, data interception, or session hijacking may result from attackers using poor encryption to decrypt SSH traffic.

5**.TCP Timestamps:**

Finding: The system may be vulnerable to specific attacks if TCP timestamps are used.

Exploitation: By using TCP timestamp information for tasks like sequence prediction, attackers may be able to facilitate additional attacks like session hijacking.

HTTP Debugging Methods (TRACE/TRACK) Enabled:

Finding: The server may be vulnerable to security threats since the TRACE and TRACK methods are enabled.

Exploitation: These techniques could be employed by attackers to launch Cross Site Tracing (XST) attacks, which could result in the loss of confidential data or session credentials.

Risk analysis and Mitigation

1.End-of-Life Operating System Detection (OSEndOfLifeDetection):

Risk Analysis:

•**Threat**: Taking advantage of well-known flaws in an outdated operating system.

•**Likelihood**: High since known vulnerabilities in systems are regularly targeted by attackers.

•**Impact**: High since it can result in data breaches, illegal access, or interruptions to services.

Mitigation:

•Upgrade to a supported operating system.

•Regularly check for security updates and patches.

•Implement network segmentation to isolate vulnerable systems.

2.ProFTPD Backdoor Unauthorized Access Vulnerability:

Risk Analysis:

•**Threat**: Unauthorized entry via a known vulnerability into the FTP server.

•**Likelihood**: High, depending on the existence of the vulnerability and the desire of a possible attacker.

•**Impact**: High since illegal access may result in service interruption or data compromise.

Mitigation:

•Update ProFTPD to the latest version.

•Monitor for unusual FTP server activities.

•Implement strong access controls and authentication mechanisms.

3.FTP Unencrypted Cleartext Login:

Risk Analysis:

•**Threat**: Interception of FTP login credentials.

•**Likelihood**: Medium, as it depends on network visibility and potential attacker presence.

•**Impact**: High, since data compromise could result from unauthorized access to FTP credentials.

Mitigation:

•Enable FTP over TLS/SSL or use SFTP.

•Implement network-level encryption to protect data in transit.

•Regularly monitor network traffic for unusual patterns.

4.**SSH Weak Encryption Algorithms Supported:**

Risk Analysis:

•**Threat**: Exploitation of weak encryption in SSH.

•**Likelihood**: Medium to High, as attackers may actively target weak encryption.

•**Impact**: High, as compromised SSH connections could lead to unauthorized access.

Mitigation:

•Disable weak encryption algorithms in SSH configurations.

•Regularly update the SSH server.

•Implement multi-factor authentication for SSH.

7.TCP Timestamps:

Risk Analysis:

•**Threat**: Exploitation of TCP timestamp information.

•**Likelihood**: Medium, as it requires specific knowledge and tools.

•**Impact**: Medium, as it could lead to attacks like sequence prediction.

Mitigation:

•Disable TCP timestamps if not needed.

•Employ intrusion detection and prevention systems.

•Regularly monitor network traffic for anomalies.

6.HTTP Debugging Methods (TRACE/TRACK) Enabled:

Risk Analysis:

Threat: Using web server attacks to take advantage of the TRACE and TRACK techniques.

Likelihood: Low to Medium, depending on how the web server is set up.

Impact: Medium because there's a chance of data leakage or session theft.

Mitigation:

•Disable TRACE and TRACK methods in the web server configuration.

•Implement web application firewalls (WAFs) to filter malicious requests.

•Regularly audit web server configurations.

Recommendations

• Conduct frequent security audits and assessments: These procedures and assessments can assist in locating gaps and vulnerabilities in the policies, processes, and systems of your company. Using this data, methods to reduce the risks connected to these vulnerabilities can be developed.

• Put access controls in place: These are crucial components of any security policy. Access to sensitive data and systems can be restricted by putting robust authentication measures in place, such as multi-factor authentication, and by upholding the least privilege principle.

• Conduct regular training on security awareness: Employee education regarding typical cyberthreats, like malware, phishing, and social engineering, might lower the likelihood that an attack will be effective.

• Put vulnerability management procedures in place: You may make sure that security patches and upgrades are applied to your network, systems, and apps by regularly scanning and discovering vulnerabilities in them.

are quickly installed.

• Segment networks and systems: In the event of a security issue, segmenting networks and systems can contain breaches and restrict attackers' lateral movement.

• Put security guidelines and protocols into place: Create and put into effect security policies and processes to give the organization a safe operating environment. To make sure these policies are still relevant, they should be routinely reviewed and updated.

• Use security products: To create layers of defense against cyberattacks, use security products like firewalls, intrusion detection/prevention systems, antivirus software, and security information and event management (SIEM) solutions.

It is significant to remember that the particular solutions for addressing the vulnerabilities found in the earlier scenarios would vary depending on the particular situation as well as the architecture, rules, and practices of the business.

Conclusion

To find vulnerabilities and take proactive measures to fix them, conduct regular penetration tests and security assessments.

To guarantee that only authorized users have access to critical data, implement a robust access control strategy that incorporates stringent permissions management, multi-factor authentication, and password regulations.

Update all software and systems with the most recent security patches and upgrades to reduce the possibility of vulnerabilities that are known to exist.

Install intrusion detection systems (IDS), firewalls, and other security measures to stop illegal access and find possible security lapses.

Instruct staff members and users on safe browsing techniques, phishing awareness, and password management, among other security best practices.

In order to avoid losing data in the event of a security breach or other disaster, regularly backup all important data and store backups in a secure location.

Put a calamity into action a recovery strategy to reduce downtime and respond to security incidents in a timely and efficient manner.

Through the implementation of these steps and vigilance, organizations can significantly mitigate the risk of security events and safeguard their servers and data.

Appendix

Tools used:

VMware:

Use Case: Setting up and maintaining virtualized environments for various operating systems to run on a single machine for testing and development.

Parrot OS:

Use Case: Digital forensics, security research, and penetration testing were conducted using Parrot OS. It offers an environment that is already set up with a range of security solutions.

netdiscover:

Use Case: was used to find active hosts on a networkby sending ARP requests and analyzing the responses, useful for network reconnaissance.

nmap:

Use Case: was employed to conduct a security audit and network research. scanning hosts in order to find open ports, services, and possible weaknesses.

OpenVAS (Open Vulnerability Assessment System):

Use Case: The servers were scanned and assessed for known vulnerabilities using the OpenVAS (Open Vulnerability Assessment System): Use Case, which generated reports on possible security concerns.

Metasploit:

Use Case: Exploit development and penetration testing are the use cases for Metasploit. creating, distributing, and testing exploits in addition to testing and taking advantage of security flaws.