Maryam Khan

CR-22021

Vulnerability Scanning

Introduction:

This report documents the findings of a vulnerability scan conducted on the Metasploitable 2 machine, a deliberately vulnerable machine used for ethical hacking practice. The purpose of the scan was to identify and categorize vulnerabilities, assess their severity, and document remediation steps while following a hypothetical Vulnerability Disclosure Policy (VDP). Metasploitable2 is often used to emulate real-world vulnerabilities in a controlled setting. As the machine is set up specifically for educational purposes, testing on it poses no legal or ethical concerns, as it is intended for security research and practice.

```
┌──(dark-girl㉿paradox)-[~]
└─$ sudo nmap -sS -sV -p- 192.168.100.48
[sudo] password for dark-girl:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-14 20:46 IST
Nmap scan report for 192.168.100.48
Host is up (0.00017s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
53/tcp    open  domain      ISC BIND 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login       OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
34269/tcp open  nlockmgr    1-4 (RPC #100021)
38424/tcp open  mountd      1-3 (RPC #100005)
50422/tcp open  java-rmi    GNU Classpath grmiregistry
55343/tcp open  status      1 (RPC #100024)
MAC Address: 08:00:27:62:DB:82 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

Link: [Vulnerability Scanning Result]

Target System: Metasploitable 2

Target System IP: 192.168.100.48

Scanning Tool: Nmap 7.94

Vulnerabilities:

Critical Vulnerabilities

1. ***vsFTPd Version 2.3.4 Backdoor (CVE-2011-2523)***
   - ***Description:*** *This vulnerability is found in vsFTPd version 2.3.4, a popular FTP server software. It contains a backdoor that was maliciously introduced in a compromised distribution of the software. Once this backdoor is exploited, it allows an attacker to execute arbitrary commands as the root user. The backdoor can be triggered by connecting to the FTP service and sending specific inputs to it.*
   - ***Impact:*** *The vulnerability allows an attacker to obtain root access, effectively taking control of the entire server. With root privileges, the attacker can install malicious software, access sensitive data, modify system settings, and even control the network. This leads to a full system compromise.*
   - ***Remediation:*** *Upgrade the vsFTPd server to version 2.3.5 or a more recent secure version. Ensure that any installed software is sourced from trusted repositories and signed distribution channels.*
2. ***RMI Registry Remote Code Execution***
   - ***Description****: The Remote Method Invocation (RMI) registry is a service that allows Java programs to invoke methods across a network. If it is configured to accept class loading from remote sources (without restriction), it opens the door to malicious code execution. An attacker can load arbitrary classes, leading to remote code execution on the system.*

- ***Impact:*** *Exploiting this vulnerability can lead to the complete takeover of the target system. An attacker can remotely execute arbitrary code, compromising the confidentiality, integrity, and availability of the system. This can result in the execution of malware, data breaches, or network infiltration.*
- ***Remediation:*** *Restrict access to the RMI registry to only trusted hosts, and disable class loading from untrusted sources. Additionally, apply security measures such as authentication and authorization for any RMI communications.*

High Vulnerabilities

1. ***SSL POODLE (CVE-2014-3566)***
   - ***Description:*** *The POODLE (Padding Oracle On Downgraded Legacy Encryption) attack is a vulnerability in the SSLv3 protocol. SSLv3 uses CBC (Cipher Block Chaining) mode for encryption, which is prone to a padding oracle attack. A man-in-the-middle attacker can exploit this flaw to decrypt encrypted information exchanged between a client and a server.*
   - ***Impact:*** *The attack allows an attacker to decrypt SSL-encrypted communications between a user and a vulnerable server. Sensitive data such as login credentials, personal information, and session tokens can be exposed to the attacker.*
   - ***Remediation****: Disable SSLv3 on both the client and server-side, and ensure that only modern versions of the TLS protocol (TLS 1.2 or higher) are used. Update the server's SSL/TLS libraries to the latest versions that are not vulnerable to the POODLE attack.*
2. ***Anonymous Diffie-Hellman Key Exchange (CVE-2015-4000)***
   - ***Description:*** *This vulnerability arises from the use of weak Diffie-Hellman key exchange parameters. If an anonymous key exchange is used or weak groups (e.g., 512-bit) are employed, an attacker can perform a man-in-the-middle attack. This allows the attacker to intercept and decrypt communications, compromising the confidentiality of the transmitted data.*
   - ***Impact:*** *Using weak Diffie-Hellman groups makes the server vulnerable to eavesdropping. Attackers can decrypt data in transit, exposing sensitive information such as login credentials, financial information, or other private communications.*
   - ***Remediation****: Reconfigure the server to use strong Diffie-Hellman groups of at least 2048-bit key lengths. Update the server software and disable weak ciphers in the server's SSL/TLS configuration.*

Medium Vulnerabilities

1. ***Slowloris DOS Attack (CVE-2007-6750)***
   - ***Description:*** *Slowloris is a type of Denial-of-Service (DoS) attack that attempts to exhaust a web server's resources by opening multiple partial connections. The attack sends HTTP headers very slowly, without completing the request, which keeps the connection open and uses up server resources.*
   - ***Impact:*** *The attack can render a web server unresponsive by keeping its connection pool saturated with incomplete requests. This can cause legitimate users to be unable to access the server or services, resulting in a denial of service.*
   - ***Remediation:*** *Mitigate Slowloris attacks by configuring server parameters such as connection timeout settings and limiting the number of concurrent connections per client. You can also use load balancers or reverse proxies that detect and block slow connections.*

2. ***JSESSIONID HttpOnly Flag Not Set***
    - ***Description****: The JSESSIONID cookie is used to track user sessions on web servers. If the HttpOnly flag is not set, the session cookie is accessible via client-side scripts. This increases the risk of session hijacking through Cross-Site Scripting (XSS) attacks, where an attacker can steal the session ID and impersonate the user.*
    - ***Impact:*** *An attacker could steal a user's session cookie and hijack their session, allowing the attacker to impersonate the user. This could lead to unauthorized access to user accounts or sensitive data.*
    - ***Remediation:*** *Set the HttpOnly flag for session cookies, including JSESSIONID. This prevents client-side scripts from accessing the cookie, mitigating the risk of session hijacking through XSS.*

Low Vulnerabilities

1. ***SQL Injection (Multiple Instances)***
    - ***Description:*** *SQL injection occurs when user inputs are not properly sanitized and are passed directly to SQL queries. Attackers can exploit this flaw to manipulate database queries by injecting malicious SQL code, allowing them to retrieve, modify, or delete data in the database.*
    - ***Impact****: Exploiting SQL injection vulnerabilities can allow attackers to read sensitive data, modify the database, delete records, or even execute administrative operations on the database. In some cases, it may also lead to remote code execution.*
    - ***Remediation:*** *Sanitize all user inputs to prevent malicious SQL code from being executed. Use prepared statements and parameterized queries in your database queries to ensure that user inputs are treated as data, not executable SQL code.*
2. ***HTTP TRACE Method Enabled***
    - ***Description:*** *The HTTP TRACE method is a diagnostic tool that echoes the contents of HTTP requests to the client. This method can be abused in Cross-Site Tracing (XST) attacks, where an attacker can steal sensitive information, such as cookies or authentication tokens, by tricking the browser into sending a TRACE request.*
    - ***Impact:*** *An attacker can capture sensitive data, including session tokens or HTTP headers, and use it for session hijacking or other malicious purposes.*
    - ***Remediation****: Disable the TRACE method on the web server. Most web servers provide an option to disable TRACE in their configuration settings to mitigate this vulnerability.*

Legal and Ethical Considerations:.

- ***Legal Compliance:*** *Scanning Metasploitable2 does not violate any laws as explicit permission is granted for this kind of security testing*
- ***Computer Fraud and Abuse Act (CFAA):*** *The Computer Fraud and Abuse Act (CFAA) in the United States makes it illegal to access computer systems without authorization. Even vulnerability scanning without explicit permission could be interpreted as unauthorized access under the CFAA. Violations of this law can result in severe legal consequences, including fines and imprisonment. In my case, since Metasploitable 2 is hosted in my VirtualBox environment and I am the system owner, there are no concerns about violating the CFAA, as the system is under my control.*
- ***General Data Protection Regulation (GDPR):*** *The General Data Protection Regulation (GDPR) applies to entities that handle the personal data of EU citizens and imposes strict obligations on*

how data is collected, stored, and processed. Scanning systems that process personal data without consent could violate the GDPR, as it could lead to unauthorized access or data breaches. Under the GDPR, any vulnerabilities that might lead to a data breach must be reported within 72 hours. However, since Metasploitable 2 does not process personal data, the GDPR does not apply in this case.

- ***Health Insurance Portability and Accountability Act (HIPAA)****: The Health Insurance Portability and Accountability Act (HIPAA) is a U.S. law designed to protect sensitive patient health information. Vulnerability scanning of systems that store or process protected health information (PHI) without proper authorization could violate HIPAA regulations, which mandate strict controls on access to health data. Since Metasploitable 2 is a test environment with no connection to PHI or healthcare data, HIPAA does not apply to my testing scenario.*
- ***Other Jurisdictional Laws****: Different countries have varying laws concerning cybersecurity and vulnerability testing. Unauthorized scans may violate not only the CFAA but also local laws that protect systems from unauthorized access, even if the scan is non-malicious. Some countries impose criminal charges for unauthorized scanning of public-facing systems or systems not owned by the scanner. While I am safe performing tests on Metasploitable 2 in my local environment, conducting scans on systems that are not under my ownership or control could result in legal repercussions in other jurisdictions.*
- ***Ethical Responsibility:*** *Always conduct vulnerability scans with the explicit permission of the system owner. If you discover vulnerabilities, responsibly disclose them according to the system owner's Vulnerability Disclosure Policy (VDP) or provide them with adequate time to patch the vulnerabilities before making any disclosures public.*
- ***Ethical Implications****: No sensitive data or real-world systems were impacted during this exercise. The testing remained confined to the controlled environment, ensuring there was no harm to third-party systems, user data, or other unintended consequences.*

Key Components of a Vulnerability Disclosure Policy (VDP):

A Vulnerability Disclosure Policy (VDP) serves as a framework to ensure vulnerabilities are identified, reported, and remediated responsibly. The key components of a VDP are:

1. ***Scope****:*
   The scope clearly defines which systems and services are included for vulnerability discovery. In my VDP, it applies to the Metasploitable system and its services, excluding any real-world production systems or third-party services that are not under control of Metasploitable.
2. ***Vulnerability Identification and Categorization****:*
   Vulnerabilities must be documented along with their risk factors and severity levels. In my documentation, I identified several vulnerabilities, including the vsFTPd backdoor (critical) and outdated versions of Apache, with proper remediation steps detailed for each one.
3. ***Reporting Guidelines****:*
   Reporting guidelines outline the process to report vulnerabilities. My documentation includes detailed instructions for reporting vulnerabilities, such as submitting a Proof of Concept (PoC), providing evidence like HTTP requests or code snippets, and ensuring the vulnerability is reproducible.
4. ***Responsible Disclosure****:*
   Responsible disclosure emphasizes the need for private and secure reporting of vulnerabilities before any public announcement. I have ensured that my VDP adheres to this principle by requiring private disclosure of any vulnerabilities and committing to transparent communication about the remediation process.

How My Documentation Aligns with VDP Components

My vulnerability disclosure documentation is aligned with these key components in the following ways:

- ***Scope****: I clearly defined the target system Metasploitable and ensured that no unauthorized scanning of real-world systems or third-party services occurred.*
- ***Vulnerability Identification****: The scan results provided a detailed classification of vulnerabilities based on their severity, including descriptions, impacts, and remediation suggestions.*
- ***Reporting Guidelines****: I have detailed guidelines for reporting vulnerabilities, emphasizing the need for PoCs and supporting evidence to ensure reproducibility.*
- ***Responsible Disclosure****: I encouraged private and secure reporting before public disclosure, offering timely updates on the remediation process.*

Steps to Ensure Responsible Disclosure

To ensure responsible disclosure, I would:

1. ***Seek Explicit Permission****: Before testing any production or real-world systems, I would ensure I have explicit permission from the system owner to conduct vulnerability scanning.*
2. ***Provide Detailed Reports****: I would submit detailed reports with evidence and descriptions to the system owner, allowing them to prioritize and fix vulnerabilities.*
3. ***Allow Time for Fixes****: I would refrain from making any public disclosures until the system owner has had enough time to patch the vulnerabilities, adhering to the VDP.*

Conclusion:

The vulnerability scan of Metasploitable 2 identified several critical, high, and medium-severity vulnerabilities. The findings were documented in alignment with a hypothetical Vulnerability Disclosure Policy (VDP), ensuring compliance with ethical and legal standards. Remediation steps for all identified vulnerabilities were provided, helping secure the system against potential attacks. Legal and ethical considerations, including those under the CFAA, GDPR, and other relevant regulations, were adhered to throughout the testing process, mitigating the risk of unintended legal consequences.