Vulnerability Assessment

of

2 Hidden Systems

Terrell Grenyion

CYBR/ISA 4220

Fall 2024 Semester

November 23rd, 2024

1.1 TABLE OF CONTENTS

1.2 EXECUTIVE SUMMARY

The Vulnerability Assessment conducted on two hidden systems identified several critical vulnerabilities on the hidden Linux server (IP: 192.168.1.30). Unfortunately, despite extensive efforts detailed later in this report, the hidden Windows server could not be identified.

The scan on the Linux server was performed using the Greenbone Vulnerability Manager and Nmap, focusing exclusively on high and medium vulnerabilities. A total of 15 high-risk vulnerabilities were identified, including Apache Tomcat AJP Remote Code Execution (RCE), PostgreSQL Default Credentials, and vsFTPd backdoor exploits. Additionally, 28 medium-risk vulnerabilities were discovered, such as weak cipher suites and improper HTTP method configurations.

To mitigate these vulnerabilities, it is essential to apply patches and updates to the affected software and services. Stronger access controls should be enforced to restrict access to exposed services, and insecure configurations must be reviewed and hardened. It is also recommended to disable unnecessary or risky services and to follow best practices for credential management to enhance the overall security posture of the system.

1.3 TECHNICAL SUMMARY

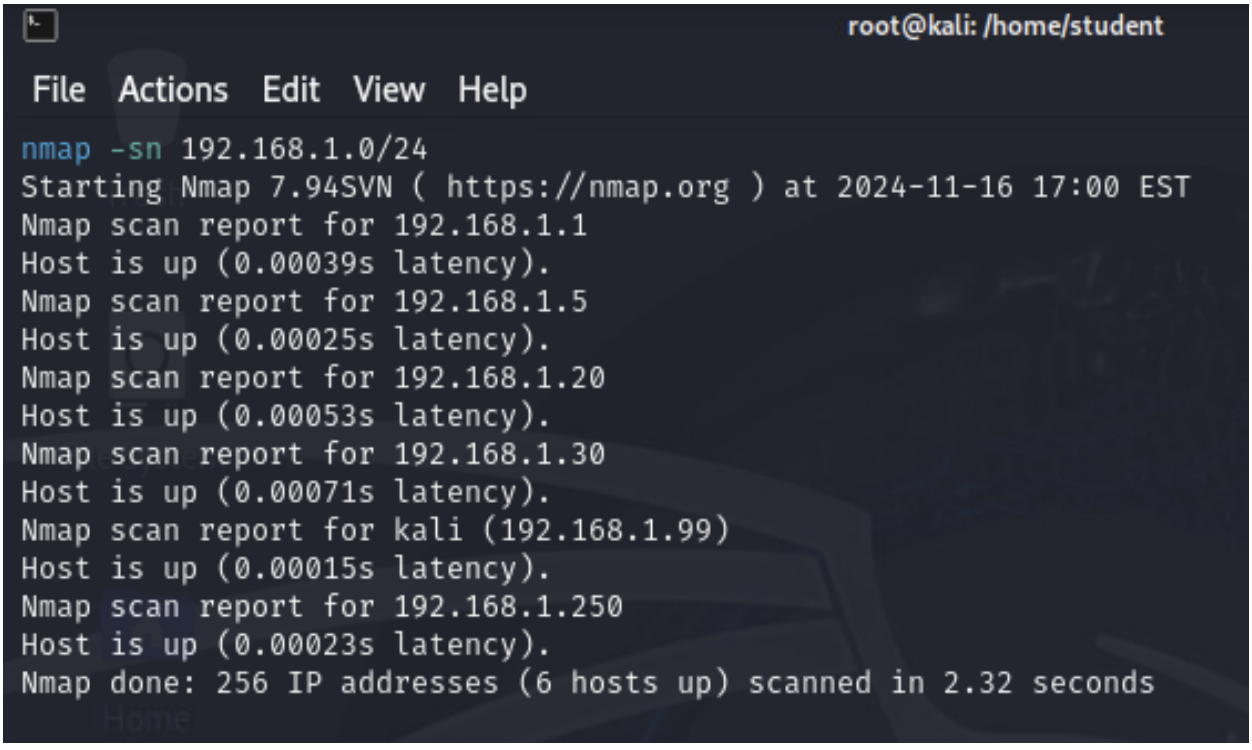
The vulnerability assessment utilized multiple tools, including Nmap and Greenbone Vulnerability Manager, to identify vulnerabilities on a hidden Linux server (IP: 192.168.1.30). Nmap was initially employed to discover active hosts on the network and verify their operational status, providing a foundational understanding of the server's open ports and services. Following this, the Greenbone Vulnerability Manager conducted a comprehensive scan using the "Full and Fast" configuration to identify vulnerabilities accurately.

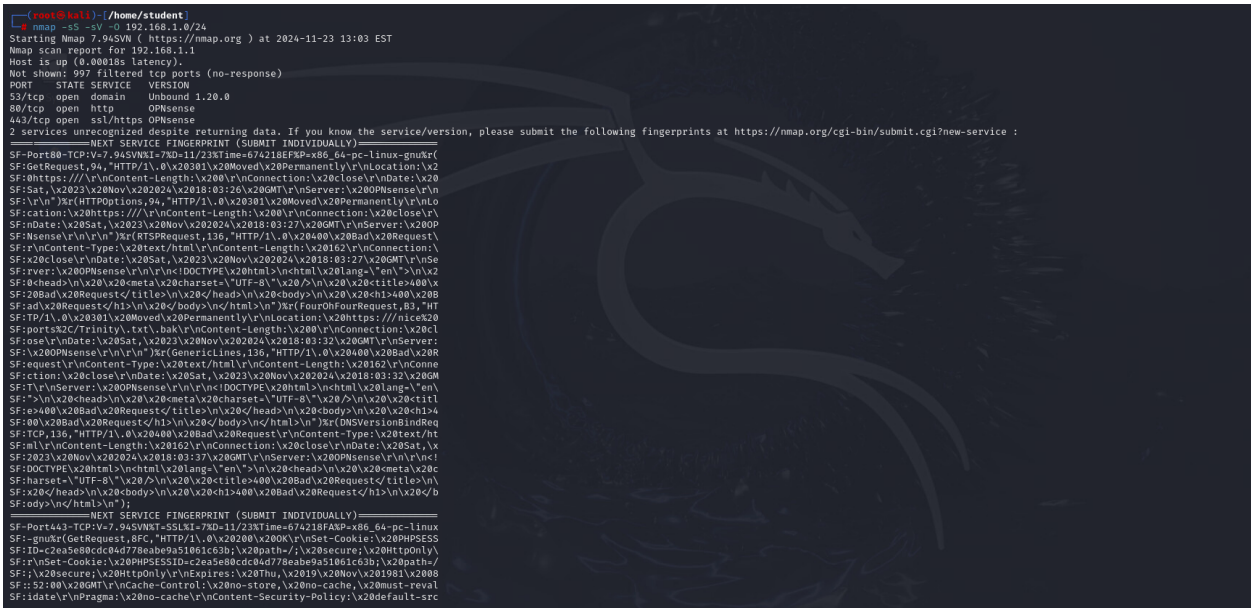
The assessment revealed 15 high-risk vulnerabilities, including critical issues such as the Apache Tomcat AJP Remote Code Execution (CVE-2020-1938), which allows unauthorized remote access, and vulnerabilities in vsFTPd that could enable attackers to execute arbitrary commands. Additionally, 28 medium-risk vulnerabilities were found, such as weak SSL/TLS cipher suites and improper HTTP methods. These findings highlight serious misconfigurations, outdated software, and insecure access protocols that could leave the server vulnerable to exploitation. Recommendations for remediation, along with further technical details, are provided in subsequent sections.

1.4 HIDDEN WINDOWS SERVER

Despite extensive efforts to locate and identify the hidden Windows server, it was ultimately concluded that the server could not be discovered. A variety of network discovery tools and methods were employed to scan the environment and detect its presence. However, none of these attempts yielded conclusive evidence of the hidden server's existence or its associated IP address.

The Nmap tool was used via the Ubuntu 24 server extensively for probing the network environment. A series of commands were executed, including "nmap -sn" for a ping sweep to discover active hosts and "nmap -sS" for a stealth scan to identify open ports and running services. Additionally, service and operating system detection scans, such as "nmap -sV" and "nmap -192.," were performed to determine each discovered host's operating system. Individual IPs were then scanned to try and spot clues to identify the hidden network. Known IPs (given and/or verified by running ipconfig/ifconfig commands on all Virtual Machines in the lab environment) were eliminated from the list of potential IPs for the hidden server. Despite these efforts, none of the detected hosts outside of the known IPs exhibited characteristics consistent with a Windows server. Below are several examples of conducted scans.





```
┌──(root㉿kali)-[/home/student]
└─# nmap -sS -sV -O 192.168.1.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-23 13:03 EST
Nmap scan report for 192.168.1.1
Host is up (0.00018s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE   VERSION
53/tcp  open  domain    Unbound 1.20.0
80/tcp  open  http      OPNsense
443/tcp open  ssl/https OPNsense
2 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port80-TCP:V=7.94SVN%I=7%D=11/23%Time=674218EF%P=x86_64-pc-linux-gnu%r(
SF:GetRequest,94,"HTTP/1\.0\x20301\x20Moved\x20Permanently\r\nLocation:\x2
SF:0https:///\r\nContent-Length:\x200\r\nConnection:\x20close\r\nDate:\x20
SF:Sat,\x2023\x20Nov\x202024\x2018:03:26\x20GMT\r\nServer:\x20OPNsense\r\n
SF:\r\n")%r(HTTPOptions,94,"HTTP/1\.0\x20301\x20Moved\x20Permanently\r\nLo
SF:cation:\x20https:///\r\nContent-Length:\x200\r\nConnection:\x20close\r\
SF:nDate:\x20Sat,\x2023\x20Nov\x202024\x2018:03:27\x20GMT\r\nServer:\x20OP
SF:Nsense\r\n\r\n")%r(RTSPRequest,136,"HTTP/1\.0\x20400\x20Bad\x20Request\
SF:r\nContent-Type:\x20text/html\r\nContent-Length:\x20162\r\nConnection:\
SF:x20close\r\nDate:\x20Sat,\x2023\x20Nov\x202024\x2018:03:27\x20GMT\r\nSe
SF:rver:\x20OPNsense\r\n\r\n<!DOCTYPE\x20html>\n<html\x20lang=\"en\">\n\x2
SF:0<head>\n\x20\x20<meta\x20charset=\"UTF-8\"\x20/>\n\x20\x20<title>400\x
SF:20Bad\x20Request</title>\n\x20</head>\n\x20<body>\n\x20\x20<h1>400\x20B
SF:ad\x20Request</h1>\n\x20</body>\n</html>\n")%r(FourOhFourRequest,B3,"HT
SF:TP/1\.0\x20301\x20Moved\x20Permanently\r\nLocation:\x20https:///nice%20
SF:ports%2C/Trinity\.txt\.bak\r\nContent-Length:\x200\r\nConnection:\x20Cl
SF:ose\r\nDate:\x20Sat,\x2023\x20Nov\x202024\x2018:03:32\x20GMT\r\nServer:
SF:\x20OPNsense\r\n\r\n")%r(GenericLines,136,"HTTP/1\.0\x20400\x20Bad\x20R
SF:equest\r\nContent-Type:\x20text/html\r\nContent-Length:\x20162\r\nConne
SF:ction:\x20close\r\nDate:\x20Sat,\x2023\x20Nov\x202024\x2018:03:32\x20GM
SF:T\r\nServer:\x20OPNsense\r\n\r\n<!DOCTYPE\x20html>\n<html\x20lang=\"en\
SF:">\n\x20<head>\n\x20\x20<meta\x20charset=\"UTF-8\"\x20/>\n\x20\x20<titl
SF:e>400\x20Bad\x20Request</title>\n\x20</head>\n\x20<body>\n\x20\x20<h1>4
SF:00\x20Bad\x20Request</h1>\n\x20</body>\n</html>\n")%r(DNSVersionBindReq
SF:TCP,136,"HTTP/1\.0\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/ht
SF:ml\r\nContent-Length:\x20162\r\nConnection:\x20close\r\nDate:\x20Sat,\x
SF:2023\x20Nov\x202024\x2018:03:37\x20GMT\r\nServer:\x20OPNsense\r\n\r\n<!
SF:DOCTYPE\x20html>\n<html\x20lang=\"en\">\n\x20<head>\n\x20\x20<meta\x20c
SF:harset=\"UTF-8\"\x20/>\n\x20\x20<title>400\x20Bad\x20Request</title>\n\
SF:x20</head>\n\x20<body>\n\x20\x20<h1>400\x20Bad\x20Request</h1>\n\x20</b
SF:ody>\n</html>\n");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port443-TCP:V=7.94SVNKT=SSLKI=7%D=11/23%Time=674218FA%P=x86_64-pc-linux
SF:-gnu%r(GetRequest,8FC,"HTTP/1\.0\x20200\x20OK\r\nSet-Cookie:\x20PHPSESS
SF:ID=c2ea5e80cdc04d778eabe9a51061c63b;\x20path=/;\x20secure;\x20HttpOnly\
SF:r\nSet-Cookie:\x20PHPSESSID=c2ea5e80cdc04d778eabe9a51061c63b;\x20path=/
SF:;\x20secure;\x20HttpOnly\r\nExpires:\x20Thu,\x2019\x20Nov\x201981\x2008
SF::52:00\x20GMT\r\nCache-Control:\x20no-store,\x20no-cache,\x20must-reval
SF:idate\r\nPragma:\x20no-cache\r\nContent-Security-Policy:\x20default-src
```

```
SF:0form\x20session\x20may\x20have\x20expired,\x20or\x20you\x20may\x20not\
SF:x20have\x20cookies\x20enabled\.</p>\n\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20</body><");
MAC Address: 00:50:56:8E:36:0E (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): FreeBSD 11.X (91%)
OS CPE: cpe:/o:freebsd:freebsd:11.2
Aggressive OS guesses: FreeBSD 11.2-RELEASE (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

Nmap scan report for 192.168.1.5
Host is up (0.00012s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
MAC Address: 00:50:56:8E:CB:17 (VMware)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.5
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.20
Host is up (0.00020s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
80/tcp open  http    Microsoft IIS httpd 10.0
MAC Address: 00:50:56:8E:12:A1 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2022|11|2016 (92%)
OS CPE: cpe:/o:microsoft:windows_server_2016
Aggressive OS guesses: Microsoft Windows Server 2022 (92%), Microsoft Windows 11 21H2 (85%), Microsoft Windows Server 2016 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.30
Host is up (0.00012s latency).
Not shown: 980 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?
25/tcp   open  smtp?
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)
```

```
512/tcp  open  exec?
513/tcp  open  login?
514/tcp  open  shell?
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Bash shell (**BACKDOOR**; root shell)
2049/tcp open  nfs         2-4 (RPC #100003)
3306/tcp open  mysql?
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:50:56:8E:9F:39 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: www, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.250
Host is up (0.00013s latency).
Not shown: 999 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
1688/tcp open  nsjtp-data?
MAC Address: 00:50:56:8E:E0:E9 (VMware)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=11/23%OT=1688%CT=1%CU=43887%PV=Y%DS=1%DC=D%G=Y%M=00
OS:5056%TM=674219A9%P=x86_64-pc-linux-gnu)SEQ(SP=F7%GCD=1%ISR=102%TI=Z%CI=Z
OS:%II=I%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11
OS:NW7%O5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE
OS:88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=4
OS:0%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O
OS:=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40
OS:%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q
OS:=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y
OS:%DFI=N%T=40%CD=S)

Network Distance: 1 hop

Nmap scan report for kali (192.168.1.99)
Host is up (0.000012s latency).
All 1000 scanned ports on kali (192.168.1.99) are in ignored states.
Not shown: 1000 closed tcp ports (reset)
Too many fingerprints match this host to give specific OS details
Network Distance: 0 hops

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (6 hosts up) scanned in 201.10 seconds
```

```
┌──(root㉿kali)-[/home/student]
└─# nmap -O 192.168.1.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-23 12:55 EST
Nmap scan report for 192.168.1.1
Host is up (0.00020s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT    STATE SERVICE
53/tcp  open  domain
80/tcp  open  http
443/tcp open  https
MAC Address: 00:50:56:8E:36:0E (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): FreeBSD 11.X (95%)
OS CPE: cpe:/o:freebsd:freebsd:11.2
Aggressive OS guesses: FreeBSD 11.2-RELEASE (95%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.57 seconds

┌──(root㉿kali)-[/home/student]
└─# nmap -O 192.168.1.5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-23 12:56 EST
Nmap scan report for 192.168.1.5
Host is up (0.00013s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:50:56:8E:CB:17 (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.51 seconds
```

1.5 HIGH FINDINGS

1.5.1 Details of findings

The assessment of the hidden Linux server revealed 11 high-risk vulnerabilities that pose significant threats to the server's integrity and security.

Firstly, the Apache Tomcat AJP connector was found to be vulnerable to the Ghostcat exploit (CVE-2020-1938). This vulnerability is related to the AJP protocol used by Apache Tomcat to communicate with other web servers and connectors. Ghostcat allows attackers to exploit insecure configurations in the AJP connector to access sensitive files, such as configuration files, or to execute arbitrary code remotely. This exploit is particularly dangerous because it enables attackers to bypass normal access controls, granting them access to restricted resources and potentially allowing them to take control of the entire server. Ghostcat is often exploited when the AJP connector is exposed to untrusted networks or lacks proper authentication, making it a significant concern for environments with public-facing services. The vulnerability has been actively exploited in real-world attacks, highlighting the urgency of its mitigation.

Additionally, PostgreSQL was discovered to be using default credentials (username: postgres, password: postgres). This oversight creates a direct pathway for unauthorized users to access the database. Default credentials are a well-known attack vector that adversaries frequently exploit to gain initial access to systems. In this case, the default username and password would allow attackers to interact with the PostgreSQL database as an administrative user, enabling them to view, modify, or delete sensitive data. This access could also allow attackers to install malicious software or escalate privileges on the underlying system, further compromising the server's security. The use of default credentials demonstrates a failure to adhere to basic security best practices, which is particularly critical in database management.

Furthermore, the vsFTPd service was found to be running a version containing a backdoor (CVE-2011-2523). This backdoor vulnerability was introduced during a specific release period when the source code for vsFTPd was compromised. The backdoor opens a shell on port 6200/TCP, allowing attackers to gain remote root-level access to the affected system. This access enables adversaries to execute any commands they wish with full administrative privileges, effectively granting them total control of the server. Such a vulnerability represents a critical threat, as it bypasses all normal security mechanisms and directly exposes the server to exploitation.

An IRC service was identified as transmitting unencrypted traffic over port 6697/tcp. While this port is typically reserved for secure IRC communications, the lack of encryption makes the service vulnerable to eavesdropping and man-in-the-middle attacks. Attackers could intercept sensitive information, such as credentials or private messages, or inject malicious data into the communication stream. This vulnerability is particularly concerning in environments where sensitive discussions or authentication details may traverse the IRC channel.

The DistCC service (CVE-2004-2687) was found to allow remote attackers to execute arbitrary commands due to insecure configurations. DistCC is a tool used to distribute compilation tasks across multiple systems, but by default, it lacks strong access controls. This misconfiguration enables attackers to exploit the service by sending malicious compilation requests that execute commands on the server. Such exploitation could lead to unauthorized data access, modification, or even a complete system compromise, depending on the commands executed.

Continuing, Distributed Ruby (dRuby/DRb) has been identified as vulnerable to remote command execution due to the insecure use of the $SAFE variable mode. dRuby is a framework for distributed programming in Ruby, and improper settings of the $SAFE variable could allow untrusted code to be executed with elevated privileges. This vulnerability enables attackers to craft malicious requests that the dRuby service may execute, potentially leading to unauthorized actions on the system. The risk is heightened when the dRuby service is exposed to untrusted networks or lacks stringent access controls.

Additionally, the Java RMI server was found to be misconfigured (CVE-2011-3556), making it susceptible to crafted packets that could execute arbitrary code. The Remote Method Invocation (RMI) protocol enables Java objects to communicate over a network, but insecure default configurations often fail to restrict access to trusted clients. This vulnerability allows attackers to exploit RMI endpoints by sending specially crafted payloads, which can lead to unauthorized code execution. The risk is amplified when the RMI server operates with elevated privileges, as successful exploitation could compromise the entire server.

The server's operating system was determined to be outdated (Ubuntu 8.04) and has reached its end-of-life (EOL) status. This means that the system no longer receives security updates or patches from the vendor, leaving it exposed to known vulnerabilities that attackers could easily exploit. Running an EOL operating system is particularly dangerous in an enterprise setting, as it creates a weak point that could be exploited to gain access to more critical systems.

Moreover, a possible backdoor was detected in the Ingreslock service (port 1524/TCP), which could provide attackers with root-level access. Backdoors are maliciously implanted vulnerabilities that allow attackers to bypass normal authentication mechanisms, granting them full control over the affected system. This backdoor was identified as responding to specific commands executed with administrative privileges, which poses a significant risk.

The TWiki application was also found to be vulnerable to cross-site scripting (XSS) and remote command execution (CVE-2008-5304) due to improper input sanitization. XSS vulnerabilities allow attackers to inject malicious scripts into web pages viewed by users, potentially stealing sensitive information such as session cookies or credentials. Furthermore, remote command execution vulnerabilities exacerbate the risk by allowing attackers to execute commands on the server hosting TWiki, leading to potential data breaches or system compromise.

Lastly, the server's SSL/TLS configuration was discovered to support outdated protocols (SSLv3, TLSv1.0, and TLSv1.1) and weak cipher suites. These configurations expose encrypted communications

to potential cryptographic attacks, such as BEAST or POODLE, which could allow attackers to decrypt sensitive data. The use of outdated encryption methods is particularly problematic in environments where secure communications are essential, such as for handling sensitive user data or authentication credentials.

1.5.2 Recommendation and fixes

To address the Apache Tomcat AJP Remote Code Execution vulnerability (CVE-2020-1938), it is essential to upgrade Apache Tomcat to a secure version, such as 7.0.100, 8.5.51, or 9.0.31 or later. You can perform the update by downloading the latest version from the official Apache Tomcat website and replacing the existing binaries while preserving the configuration files.

If the AJP connector is unnecessary for the server's operation, you should disable it in the server.xml configuration file by commenting out or removing the AJP `<Connector>` section. If the AJP connector must remain active, configure it to bind only to localhost by specifying `address="127.0.0.1"` and enforcing IP-based restrictions using a firewall. For instance, on Linux systems using iptables, you can implement the following rule: `iptables -A INPUT -p tcp --dport 8009 -s TRUSTED_IP -j ACCEPT`. This ensures that only trusted IP addresses can interact with the AJP service.

For the PostgreSQL database set with default credentials, immediate remediation involves updating the password for the 'postgres' user. You can use the PostgreSQL command-line interface or a tool like pgAdmin to execute the following command: `ALTER USER postgres WITH PASSWORD 'YourSecurePassword';`. Additionally, a strong password policy should be enforced using tools like pgcrypto to ensure the use of complex and unique passwords. Furthermore, restrict remote access to the database by editing the pg_hba.conf file to allow connections only from trusted IP ranges. For example, add the line `host all all 192.168.1.0/24 md5` to permit access exclusively from the specified subnet.

The vsFTPd service should be upgraded to a secure version from the official vendor’s repository. You can update the service using a package manager like apt by running: `sudo apt update && sudo apt install vsftpd`. It is crucial to verify the integrity of the downloaded files using a cryptographic signature or hash provided by the vendor to ensure authenticity. After upgrading, review the vsftpd.conf configuration file to disable potentially insecure options, such as anonymous access, and ensure that SSL/TLS encryption is enabled for all connections.

To secure the IRC service, enable SSL/TLS encryption to prevent eavesdropping and data manipulation. Modify the server configuration file to specify the use of an SSL certificate, ensuring that all communications are encrypted. If a trusted CA-issued certificate is unavailable, you can use OpenSSL to generate a self-signed certificate with the following command: `openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365 -nodes`. Then, configure the IRC server to reference these certificate files.

For the DistCC service, limit its accessibility to trusted IP addresses by editing the service's access control list (ACL) or using firewall rules. For example, you can add `--allow TRUSTED_IP` in the service configuration file or use iptables to enforce network restrictions. Additionally, upgrade to the latest secure version of DistCC by following the vendor’s instructions and applying security patches promptly.

The vulnerability in Distributed Ruby (dRuby/DRb) can be mitigated by updating the Ruby interpreter to the latest version and ensuring that the $SAFE level is set to 3 or higher. The $SAFE variable restricts the execution of potentially unsafe code, and you can enable it in the Ruby application code by using `Thread.current[:SAFE] = 3`. Additional precautions include binding the DRb service to localhost and configuring a firewall to limit access.

To address the misconfiguration in the Java RMI server (CVE-2011-3556), disable the class-loading feature by adding `-Djava.rmi.server.useCodebaseOnly=true` to the startup options of the Java process. Make sure to apply all relevant patches provided by the Java vendor to fix known vulnerabilities. These steps will help reduce the attack surface and limit the likelihood of attackers exploiting the RMI service.

The server's operating system, Ubuntu 8.04, needs to be upgraded to a supported version such as Ubuntu 22.04 LTS. Start by backing up all critical data, then perform a clean installation of the new version. This will ensure the server receives the latest security patches and reduces exposure to known vulnerabilities. After the upgrade, implement a regular patching schedule to maintain system security.

For the Ingreslock backdoor, a complete system reinstallation is necessary to ensure the integrity of the operating environment. Before reinstalling, use tools like chkrootkit or rkhunter to check for and remove backdoors. After the reinstallation, implement host-based intrusion detection systems (HIDS) such as OSSEC to monitor for potential threats.

To resolve vulnerabilities in the TWiki application, upgrade to version 4.2.4 or later, which addresses known cross-site scripting (XSS) and remote command execution flaws. Additionally, implement input sanitization in the TWiki configuration by enabling Encode::XS or equivalent modules to properly escape user input. Ensure all third-party plugins are up-to-date and review their configurations for potential security risks.

Finally, address the outdated SSL/TLS configuration by updating the server to support TLS 1.2 or TLS 1.3 and configuring it to use strong cipher suites. Edit the server’s SSL configuration file (e.g., /etc/ssl/openssl.cnf) to include only secure protocols and ciphers, such as TLS_AES_256_GCM_SHA384. Utilize tools like OpenSSL or SSL Labs to verify the implementation and identify weak points.

1.6 CONCLUSION

The vulnerability assessment of the hidden Linux server (IP: 192.168.1.30) revealed critical security risks, including 11 high-risk vulnerabilities that require immediate attention. These vulnerabilities, such as the Apache Tomcat AJP Remote Code Execution, use of default PostgreSQL

credentials, and backdoors in services like vsFTPd and Ingreslock, present serious threats that could lead to complete system compromise, unauthorized data access, and significant operational disruptions. The identified risks point to critical issues in configuration, outdated software, and inadequate access controls, all of which attackers could exploit to escalate privileges, execute arbitrary code, or intercept sensitive communications.

To address these vulnerabilities, it is essential to prioritize the remediation of high-risk issues through a combination of updates, reconfiguration, and removal of unnecessary or insecure services. Recommendations include upgrading outdated software versions, implementing strong access controls, enforcing strong encryption protocols, and ensuring proper credential management practices. These measures are crucial to reducing the server's exposure to potential attacks and enhancing its overall security posture.

Although the hidden Windows server could not be identified during the assessment, the efforts made to locate it illustrate a systematic approach using network discovery tools and thorough investigative methods. The findings and actionable recommendations provided in this report outline a clear pathway for mitigating risks and securing the Linux server against future threats.

1.7 ANNEXES

1.7.1.1 Greenbone scan results

Click the image below to be redirected to the full report.

Scan Report

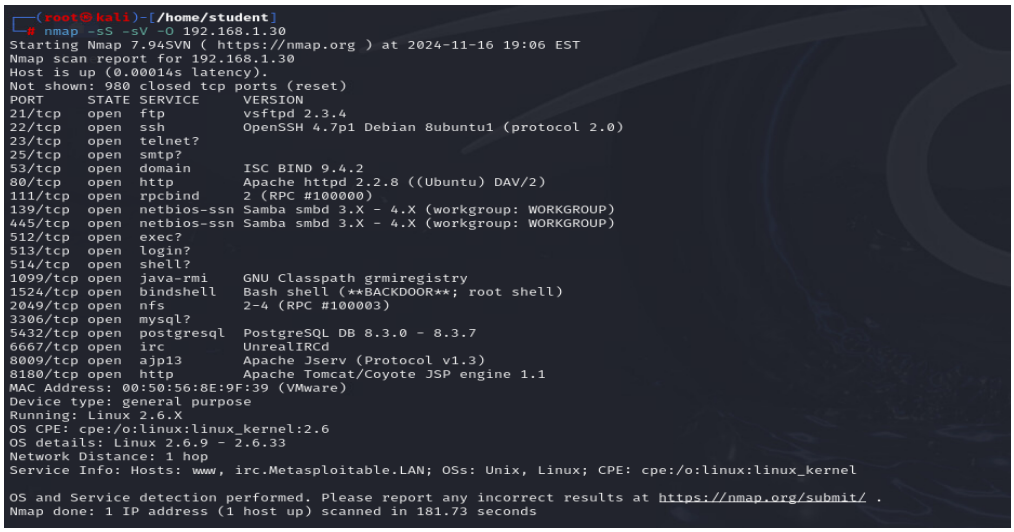
November 17, 2024

Summary

This document reports on the results of an automatic security scan. All dates are displayed using the timezone "Coordinated Universal Time", which is abbreviated "UTC". The task was "tgrenyio". The scan started at Sat Nov 16 23:41:21 2024 UTC and ended at Sun Nov 17 00:27:19 2024 UTC. The report first summarises the results found. Then, for each host, the report describes every issue found. Please consider the advice given in each description, in order to rectify the issue.

Contents

1.7.1.2 Nmap scan results – Linux server



1.7.1.3 Vulnerability ranking scale

The vulnerability ranking scale used in this assessment is based on the Common Vulnerability Scoring System (CVSS) v3.1, which evaluates vulnerabilities according to their severity and impact. The CVSS scoring system assigns a numerical value between 0.0 and 10.0 to each vulnerability, categorizing them into the following levels:

- **Critical (9.0 - 10.0):** These vulnerabilities pose the most severe threats, often leading to complete system compromise or allowing attackers to execute arbitrary code remotely. They typically require immediate remediation due to their potential for catastrophic consequences.

- **High (7.0 - 8.9:** High-risk vulnerabilities can significantly impact system confidentiality, integrity, or availability. They are often exploited in real-world scenarios and should be addressed as a priority.

- **Medium (4.0 - 6.9):** Medium-risk vulnerabilities are less likely to result in immediate compromise but can still be exploited to facilitate attacks. They may represent weaknesses that an attacker can combine with other vulnerabilities.

- **Low (0.1 - 3.9):** These vulnerabilities pose minimal risk and often require complex conditions to exploit. They are generally addressed during regular maintenance cycles.

- **Informational (0.0):** Informational findings are not vulnerabilities but may highlight areas for improved security practices or potential misconfigurations.

The severity levels in this assessment were determined based on their CVSS base scores, combined with insights from the Greenbone Vulnerability Manager. The CVSS evaluates multiple factors, including exploitability, impact, and environmental modifiers, to ensure accurate prioritization. High and

medium vulnerabilities were prioritized for remediation in this report, in alignment with the organization's risk management strategy.

1.8 REFERENCES

2011-2523 - opencve. CVE. (2019a, November 27). https://app.opencve.io/cve/CVE-2011-2523

2011-2523 : VSFTPD 2.3.4 downloaded between 20110630 and 20110703 contains a backdoor which. CVE. (2019b, November 27). https://www.cvedetails.com/cve/CVE-2011-2523

AJP file read/inclusion in Apache Tomcat (CVE-2020-1938) and Undertow (CVE-2020-1745). Red Hat Customer Portal. (2024, June 14). https://access.redhat.com/solutions/4851251

Apache tomcat: Important: AJP request injection and potential remote code execution (CVE-2020-1938). Rapid7. (n.d.). https://www.rapid7.com/db/vulnerabilities/apache-tomcat-cve-2020-1938/

Carr, A. (2020, March 4). *Ghostcat vulnerability in Apache Tomcat: What you need to know*. OpenLogic by Perforce. https://www.openlogic.com/blog/ghostcat-vulnerability

Common Vulnerability Scoring System Calculator CVE-2011-2523. NIST. (n.d.). https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?name=CVE-2011-2523&source=NIST&vector=AV%3AN%2FAC%3AL%2FPR%3AN%2FUI%3AN%2FS%3AU%2FC%3AH%2FI%3AH%2FA%3AH&version=3.1

CVE-2011-2523 Detail. NVD. (n.d.). https://nvd.nist.gov/vuln/detail/CVE-2011-2523

CVE-2011-2523 report - details, severity, & advisories. Twingate. (2024, June 6). https://www.twingate.com/blog/tips/cve-2011-2523

CVE-2011-2523. (2019, November 27). https://vulmon.com/vulnerabilitydetails?qid=CVE-2011-2523

CVE-2011-2523. CVE website. (2019, November 27). https://www.cve.org/CVERecord?id=CVE-2011-2523

CVE-2011-2523. CVE. (n.d.-a). https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523

CVE-2011-2523. INCIBE. (2019, November 27). https://www.incibe.es/en/incibe-cert/early-warning/vulnerabilities/cve-2011-2523

CVE-2020-1938 vulnerability in Apache and other products. CVE-2020-1938 vulnerability in Apache and Other Products. (2020, February 24). https://stack.watch/vuln/CVE-2020-1938/

CVE-2020-1938. CVE. (n.d.-b). https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1938

CVE-2020-1938. NVD. (2020, February 24). https://nvd.nist.gov/vuln/detail/CVE-2020-1938

HerculesRD. (2021, April 12). *VSFTPD 2.3.4 - backdoor command execution*. Exploit Database. https://www.exploit-db.com/exploits/49757

Logan, M. (2020, March 10). *Busting ghostcat: Analysis of CVE-2020-1938*. Trend Micro. https://www.trendmicro.com/en_be/research/20/c/busting-ghostcat-an-analysis-of-the-apache-tomcat-vulnerability-cve-2020-1938-and-cnvd-2020-10487.html

Metasploit. (2011, July 5). *VSFTPD 2.3.4 - backdoor command execution (Metasploit)*. Exploit Database. https://www.exploit-db.com/exploits/17491

Narang, S. (2023, October 30). *CVE-2020-1938: Ghostcat - Apache Tomcat AJP file read/inclusion vulnerability (CNVD-2020-10487)*. Tenable®. https://www.tenable.com/blog/cve-2020-1938-ghostcat-apache-tomcat-ajp-file-readinclusion-vulnerability-cnvd-2020-10487

Sethi, T. (2020, March 31). *How to fix the Ghostcat vulnerability (CVE-2020-1938): Black duck blog*. How to Fix the Ghostcat Vulnerability (CVE-2020-1938) | Black Duck Blog. https://www.blackduck.com/blog/ghostcat-vulnerability-cve-2020-1938.html

Sharma, A. (2024, August 12). *What's in a Ghostcat? CVE-2020-1938 Apache tomcat LFI and RCE risks*. What's in a Ghostcat? CVE-2020-1938 Apache Tomcat LFI and RCE Risks. https://www.sonatype.com/blog/nexus-intelligence-insights-whats-in-a-ghostcat-cve-2020-1938-apache-tomcat