Vulnerability Remediation Report: VNC Server Weak Password

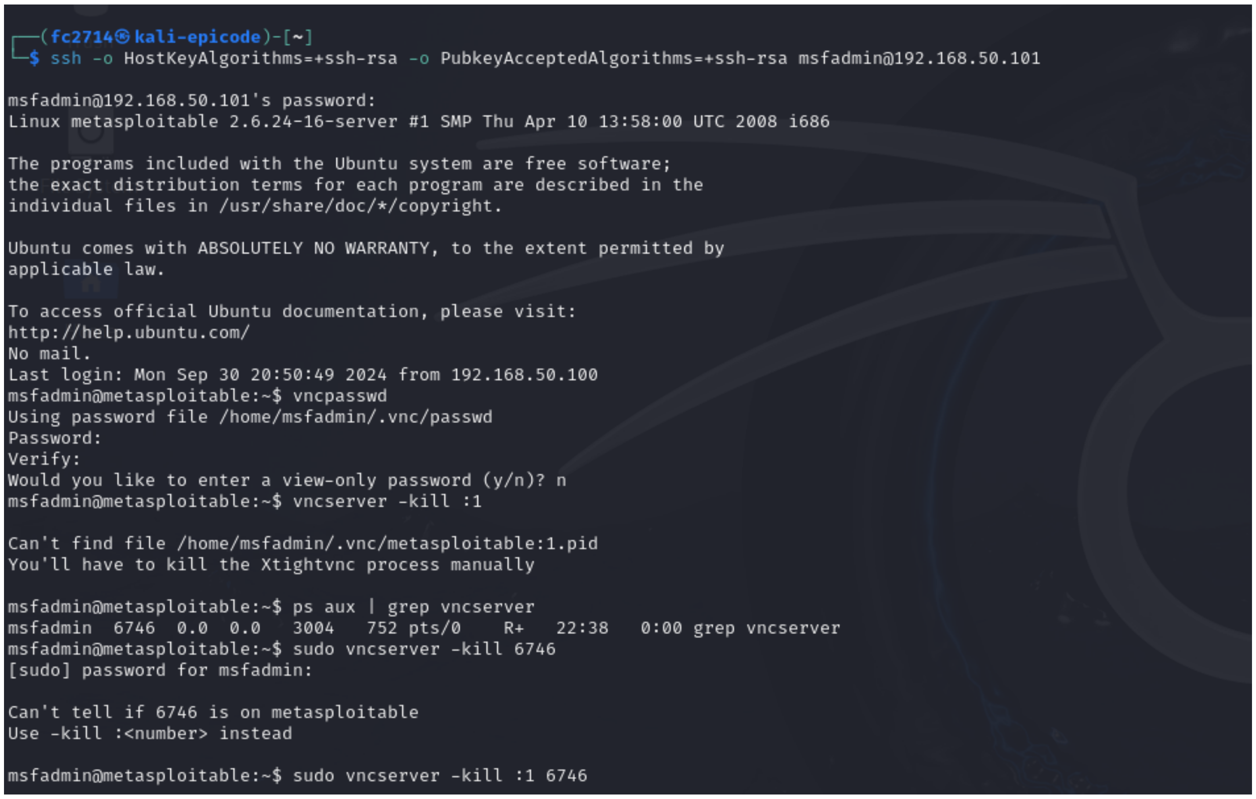
Initial Vulnerability:

The vulnerability identified was a weak password set on the VNC server running on the Metasploitable machine. This allowed potential attackers to easily gain access through the VNC service due to the default password.

Objective:

To remediate this vulnerability by changing the VNC password to a more secure one and verifying that unauthorized access with the old password is no longer possible.

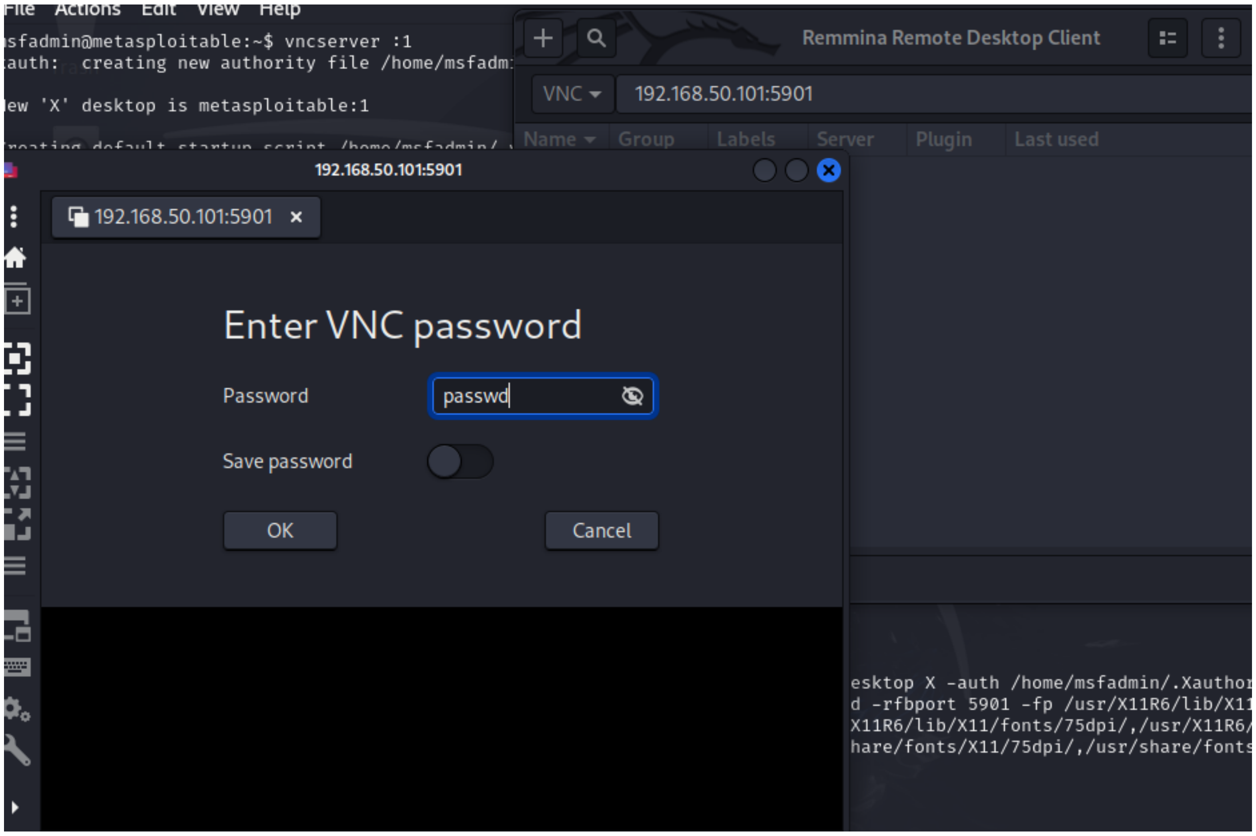
Step 1: SSH Connection and Changing VNC Password



```
┌──(fc2714㉿kali-epicode)-[~]
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedAlgorithms=+ssh-rsa msfadmin@192.168.50.101

msfadmin@192.168.50.101's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Mon Sep 30 20:50:49 2024 from 192.168.50.100
msfadmin@metasploitable:~$ vncpasswd
Using password file /home/msfadmin/.vnc/passwd
Password:
Verify:
Would you like to enter a view-only password (y/n)? n
msfadmin@metasploitable:~$ vncserver -kill :1

Can't find file /home/msfadmin/.vnc/metasploitable:1.pid
You'll have to kill the Xtightvnc process manually

msfadmin@metasploitable:~$ ps aux | grep vncserver
msfadmin  6746  0.0  0.0   3004   752 pts/0    R+   22:38   0:00 grep vncserver
msfadmin@metasploitable:~$ sudo vncserver -kill 6746
[sudo] password for msfadmin:

Can't tell if 6746 is on metasploitable
Use -kill :<number> instead

msfadmin@metasploitable:~$ sudo vncserver -kill :1 6746
```

1. We connected to the Metasploitable machine via SSH using the credentials msfadmin@192.168.50.101.
2. The command vncpasswd was executed to change the VNC password from the weak default password (passwd) to a more secure one.
3. After setting the new password, the VNC server was restarted using the following commands: vncserver -kill :1 vncserver :1

Step 2: Attempting to Connect with Old Password


File Actions Edit View Help
Remmina Remote Desktop Client
VNC
Name Group Labels Server Plugin Last used
OK
Cancel

```
sfadmin@metasploitable:~$ vncserver :1
auth:  creating new authority file /home/msfadm

ew 'X' desktop is metasploitable:1
```

```
192.168.50.101:5901
```

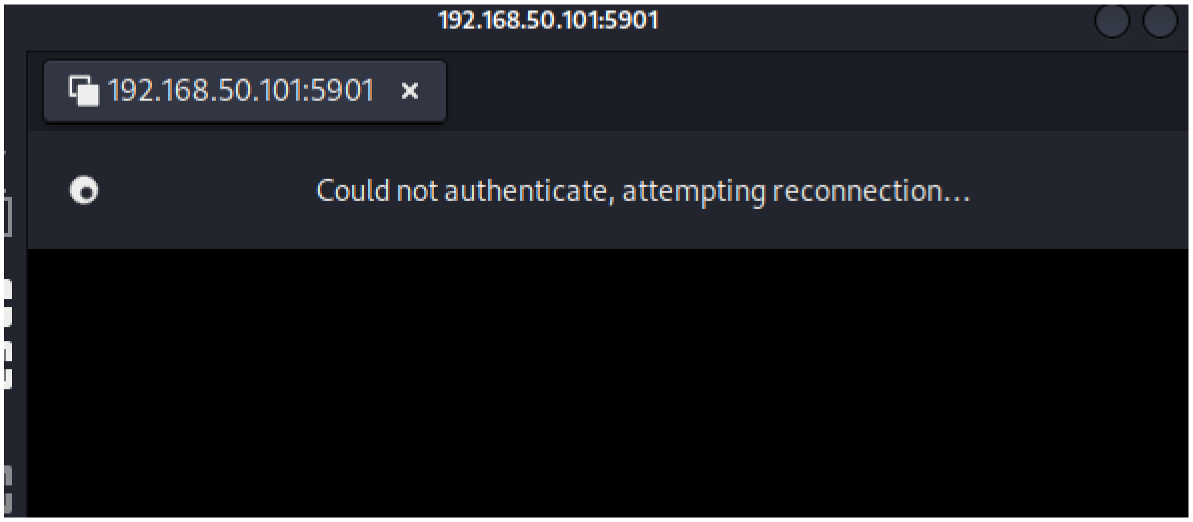
Enter VNC password

Password: passwd

Save password

```
esktop X -auth /home/msfadmin/.Xauthor
d -rfbport 5901 -fp /usr/X11R6/lib/X11
X11R6/lib/X11/fonts/75dpi/,/usr/X11R6/
hare/fonts/X11/75dpi/,/usr/share/fonts
```

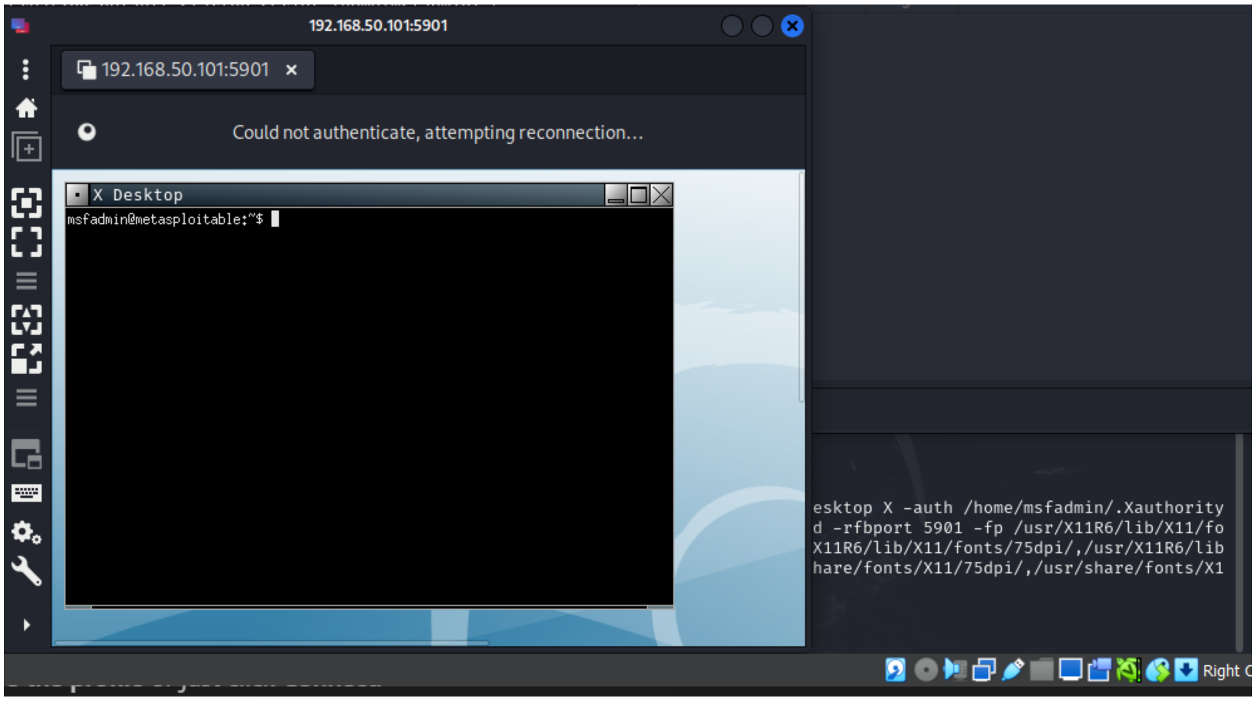
1. In this step, we tried to connect to the VNC server using Remmina from a different machine.
2. We entered the old password (passwd) to see if the change was applied successfully.
3. The connection attempt failed, and we received an authentication error message, which confirmed that the old password is no longer valid.



192.168.50.101:5901

Could not authenticate, attempting reconnection...

This screenshot shows the failed connection attempt with the old password (passwd), confirming the password change was successful.



Could not authenticate, attempting reconnection...

```
msfadmin@metasploitable:~$
```

```
esktop X -auth /home/msfadmin/.Xauthority
d -rfbport 5901 -fp /usr/X11R6/lib/X11/fo
X11R6/lib/X11/fonts/75dpi/,/usr/X11R6/lib
hare/fonts/X11/75dpi/,/usr/share/fonts/X1
```

-
    - Finally, after entering the new, secure password that was set, the connection was successful, and we were able to access the VNC server without any issues.

Conclusion:

- The VNC server password was successfully changed to a more secure one. After the remediation, attempts to access the VNC server using the old, insecure password failed, confirming that the vulnerability has been addressed. The new password allows secure authentication, and the VNC service is no longer exposed to this password-based vulnerability.

Vulnerability 2: Remediation of Apache Tomcat AJP Connector Request Injection (Ghostcat)

Vulnerability Overview:

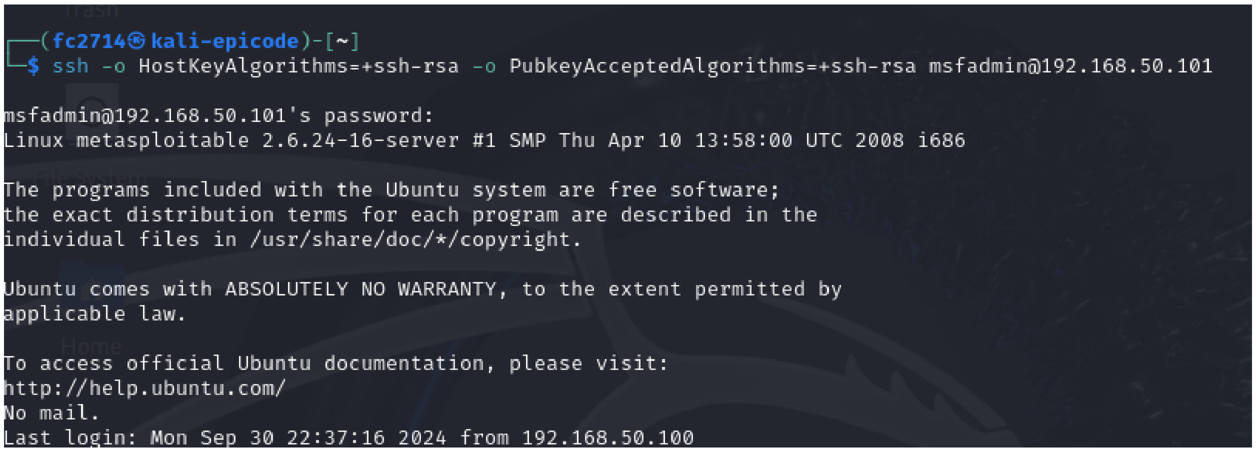
The Apache Tomcat AJP Connector vulnerability, commonly referred to as **Ghostcat**, allows attackers to read or include files in the webapp directories of Tomcat servers. This vulnerability (CVE-2020-1938) is caused by the misconfiguration of the Apache JServ Protocol (AJP) connector.

Objective:

To remediate this vulnerability by disabling the AJP connector or upgrading to a patched version of Tomcat, if necessary.

Step 1: Identifying the Vulnerable Configuration

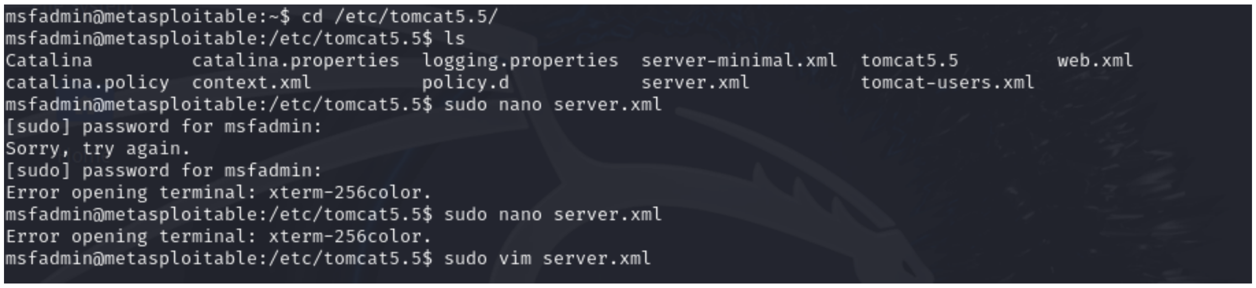
- First, we need to verify that the AJP connector is enabled in Tomcat's configuration file.
- Let's SSH Metasploitable machine where Tomcat is running:



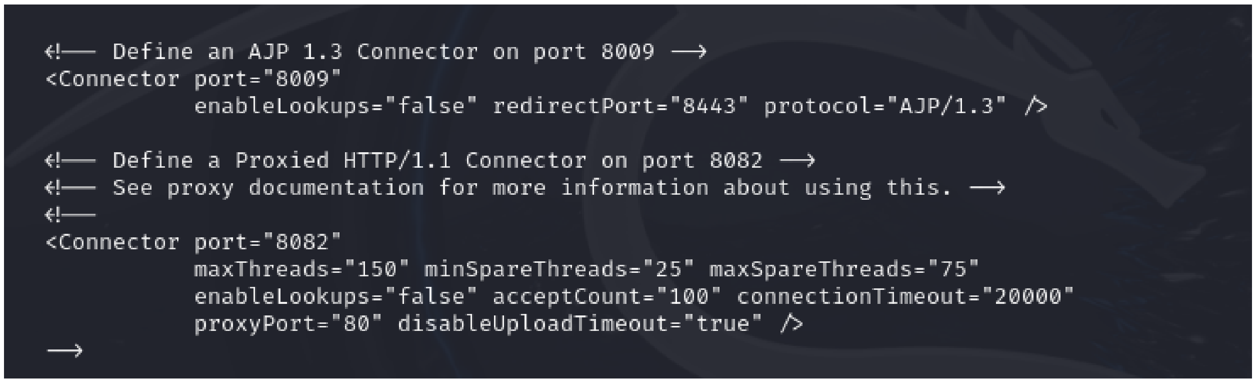
```
┌──(fc2714㉿kali-epicode)-[~]
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedAlgorithms=+ssh-rsa msfadmin@192.168.50.101

msfadmin@192.168.50.101's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Mon Sep 30 22:37:16 2024 from 192.168.50.100
```

Step 2: Disable the AJP Connector

Let's navigate to the **Tomcat configuration directory** , then we open the **server.xml** file where the AJP connector is configured, We need to look for a line similar to this in the **server.xml** file:
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />



```
msfadmin@metasploitable:~$ cd /etc/tomcat5.5/
msfadmin@metasploitable:/etc/tomcat5.5$ ls
Catalina         catalina.properties  logging.properties  server-minimal.xml  tomcat5.5         web.xml
catalina.policy  context.xml          policy.d            server.xml          tomcat-users.xml
msfadmin@metasploitable:/etc/tomcat5.5$ sudo nano server.xml
[sudo] password for msfadmin:
Sorry, try again.
[sudo] password for msfadmin:
Error opening terminal: xterm-256color.
msfadmin@metasploitable:/etc/tomcat5.5$ sudo nano server.xml
Error opening terminal: xterm-256color.
msfadmin@metasploitable:/etc/tomcat5.5$ sudo vim server.xml
```

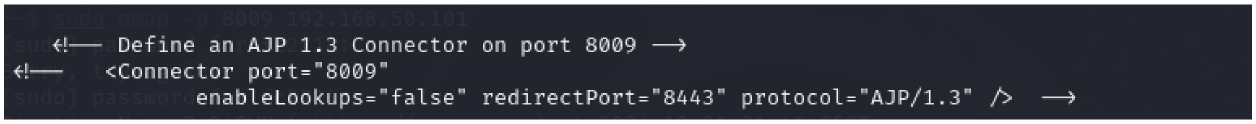


```
    <!-- Define an AJP 1.3 Connector on port 8009 -->
    <Connector port="8009"
               enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />

    <!-- Define a Proxied HTTP/1.1 Connector on port 8082 -->
    <!-- See proxy documentation for more information about using this. -->
    <!--
    <Connector port="8082"
               maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
               enableLookups="false" acceptCount="100" connectionTimeout="20000"
               proxyPort="80" disableUploadTimeout="true" />
    -->
```

We found that line.

To remediate this vulnerability, you can either **disable** the AJP connector or **limit access** to it. In this case, we will disable it.

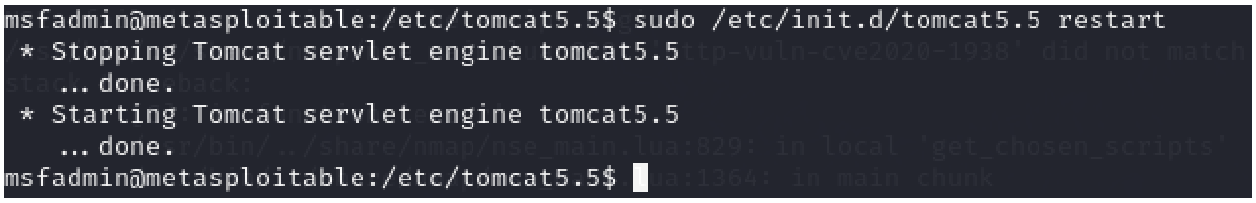
We will comment this line.



```
    <!-- Define an AJP 1.3 Connector on port 8009 -->
<!--    <Connector port="8009"
               enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />  -->
```

Step 3: Restart Tomcat

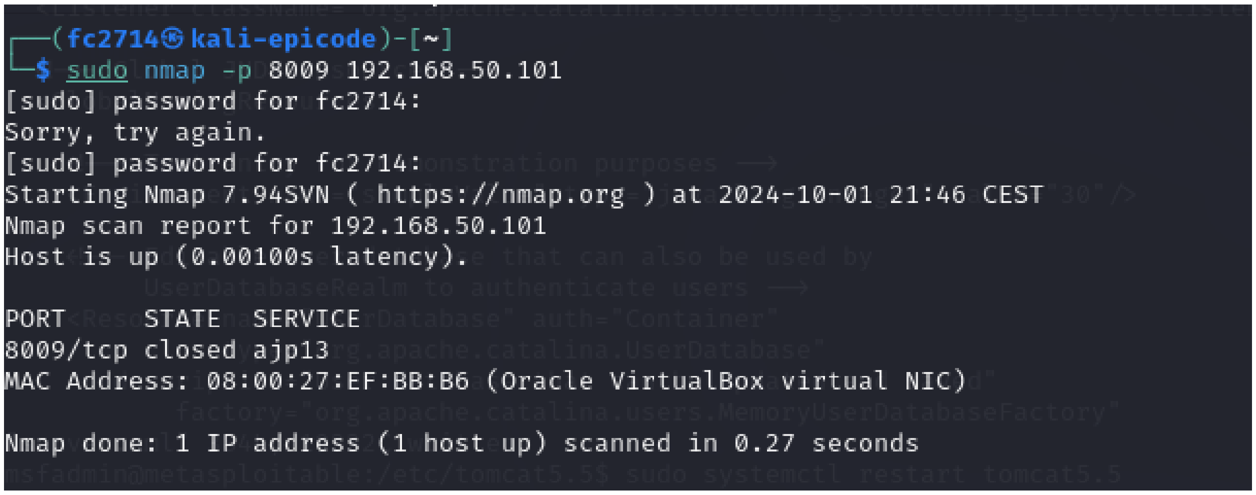
After making changes to the configuration, we restart the Tomcat service to apply the new settings



```
msfadmin@metasploitable:/etc/tomcat5.5$ sudo /etc/init.d/tomcat5.5 restart
 * Stopping Tomcat servlet engine tomcat5.5
   ...done.
 * Starting Tomcat servlet engine tomcat5.5
   ...done.
msfadmin@metasploitable:/etc/tomcat5.5$
```

Step 4: Verify the Remediation

Once the Tomcat server has been restarted, let's check if the AJP connector is disabled by scanning the ports on the Metasploitable machine:



```
┌──(fc2714㉿kali-epicode)-[~]
└─$ sudo nmap -p 8009 192.168.50.101
[sudo] password for fc2714:
Sorry, try again.
[sudo] password for fc2714:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-01 21:46 CEST
Nmap scan report for 192.168.50.101
Host is up (0.00100s latency).

PORT     STATE  SERVICE
8009/tcp closed ajp13
MAC Address: 08:00:27:EF:BB:B6 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
```

As we see it is closed.

Vulnerability 3: Remediation of SSL Version 2 and 3 Protocol Detection

Vulnerability Report: Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL Check)

Overview of Vulnerability

The OpenSSH/OpenSSL vulnerability stems from an issue with the random number generator in Debian-based systems. This bug affects cryptographic materials such as SSH and SSL certificates, making them predictable and susceptible to attacks. An attacker can potentially obtain the private keys and use them for man-in-the-middle attacks or decrypt sensitive communication.

- **ID: 32314**
- **Severity: Critical**
- **Affected Services: SSH (Port 22), SMTP (Port 25), PostgreSQL (Port 5432), HTTPS (Port 443)**
- **Description: The remote host uses weak cryptographic material generated using a flawed OpenSSL library on Debian-based systems.**

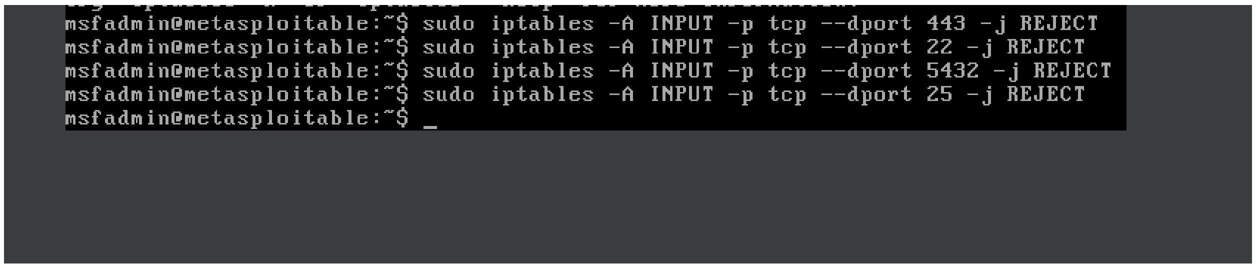
- **Risk: Attackers could easily gain unauthorized access or decrypt communication channels.**

Implemented Solution: Firewall-Based Blocking

Due to the limitations of the old Metasploitable environment, updating OpenSSL and enforcing stronger security protocols (such as TLSv1.2 or TLSv1.3) is not feasible. We opted for a temporary mitigation using firewall rules to block access to the affected ports (22, 25, 5432, and 443).

Steps Taken:

1. **Blocking Inbound Traffic on Affected Ports**: To mitigate the risk of exploitation, we configured iptables to block all inbound traffic on the vulnerable services' ports:



```
msfadmin@metasploitable:~$ sudo iptables -A INPUT -p tcp --dport 443 -j REJECT
msfadmin@metasploitable:~$ sudo iptables -A INPUT -p tcp --dport 22 -j REJECT
msfadmin@metasploitable:~$ sudo iptables -A INPUT -p tcp --dport 5432 -j REJECT
msfadmin@metasploitable:~$ sudo iptables -A INPUT -p tcp --dport 25 -j REJECT
msfadmin@metasploitable:~$ _
```

2. **Reason for Using Firewall Blocking**: Since the Metasploitable environment uses an outdated version of OpenSSL, and upgrading OpenSSL or enabling secure protocols (like TLSv1.2) is not feasible due to compatibility issues, the best temporary mitigation option was to block access to these critical ports to prevent exploitation.

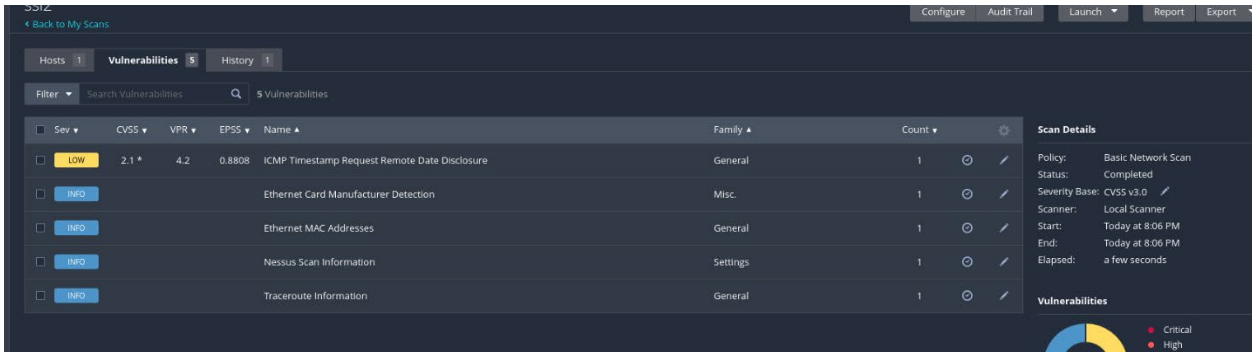
Limitations and Recommendations

- **Current Limitations**: The above solution does not permanently resolve the vulnerability. The firewall rules only block external access, but the underlying weakness remains in place. If the attacker has access to the system locally, they could still exploit the weak cryptographic material.
- **Recommended Remediation**:
    1. **Upgrade OpenSSL**: In a modern environment, updating OpenSSL to the latest stable version (1.1.1 or later) is essential to resolve the vulnerability. This would involve regenerating all cryptographic materials (certificates, SSH keys, etc.).
    2. **Enforce TLSv1.2 or TLSv1.3**: Once OpenSSL is updated, ensure that all services (such as Apache for HTTPS) only allow connections over TLSv1.2 or higher and disable support for weak ciphers like SSLv2 and SSLv3.

3. **Strong Cipher Suites**: Modify the configuration files (e.g., ssl.conf for Apache) to use strong cipher suites, ensuring secure communications.

Conclusion

In a real-world scenario, the vulnerability would need to be resolved by updating OpenSSL and enforcing modern security practices. Since our current Metasploitable system does not support these actions, we have mitigated the issue temporarily using a firewall-based approach by blocking critical service ports. For a permanent fix, upgrading the environment is necessary.



Sev	CVSS	VPR	EPSS	Name	Family	Count
LOW	2.1 *	4.2	0.8808	ICMP Timestamp Request Remote Date Disclosure	General	1
INFO				Ethernet Card Manufacturer Detection	Misc.	1
INFO				Ethernet MAC Addresses	General	1
INFO				Nessus Scan Information	Settings	1
INFO				Traceroute Information	General	1

Scan Details

Policy:	Basic Network Scan
Status:	Completed
Severity Base:	CVSS v3.0
Scanner:	Local Scanner
Start:	Today at 8:06 PM
End:	Today at 8:06 PM
Elapsed:	a few seconds

Vulnerabilities

Vulnerability 4.

Bind Shell Backdoor Detection Vulnerability (Port 1524)

Summary:

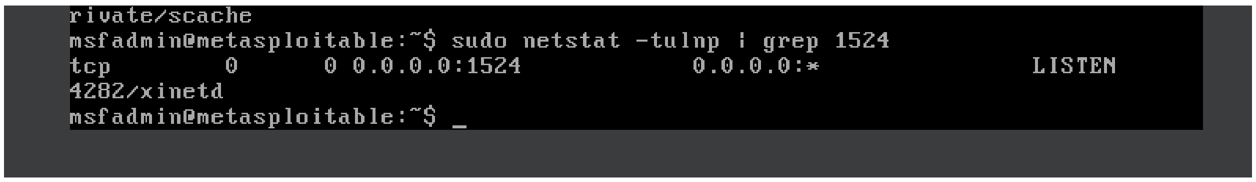
The system detected a bind shell backdoor running on port 1524. This type of vulnerability allows attackers to remotely execute commands and potentially take control of the system. This bind shell was detected on the Metasploitable environment, which contains inherent security vulnerabilities.

Details:

- **Vulnerable Port**: 1524
- **Vulnerability Type**: Bind Shell Backdoor
- **Detected Service**: Wild Shell (service running on port 1524)
- **CVSS Score**: Critical
- **Risk**: Remote attackers can gain unauthorized shell access to the system, execute commands remotely, and escalate privileges, which may lead to a complete system compromise.

Steps Taken:

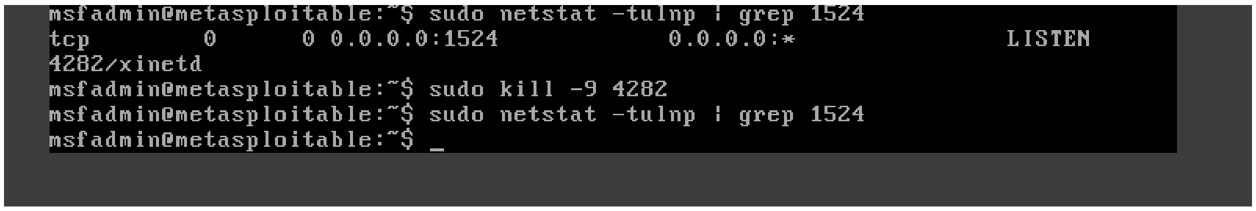
1. **Identified the Process on Port 1524**: We ran the following command to identify the process running on the vulnerable port:



```
rivate/scache
msfadmin@metasploitable:~$ sudo netstat -tulnp | grep 1524
tcp        0      0 0.0.0.0:1524          0.0.0.0:*             LISTEN
4282/xinetd
msfadmin@metasploitable:~$ _
```

This showed the process related to the bind shell.

2. **Terminated the Vulnerable Process**: Using the process ID (PID) identified from the previous step, we killed the bind shell service:

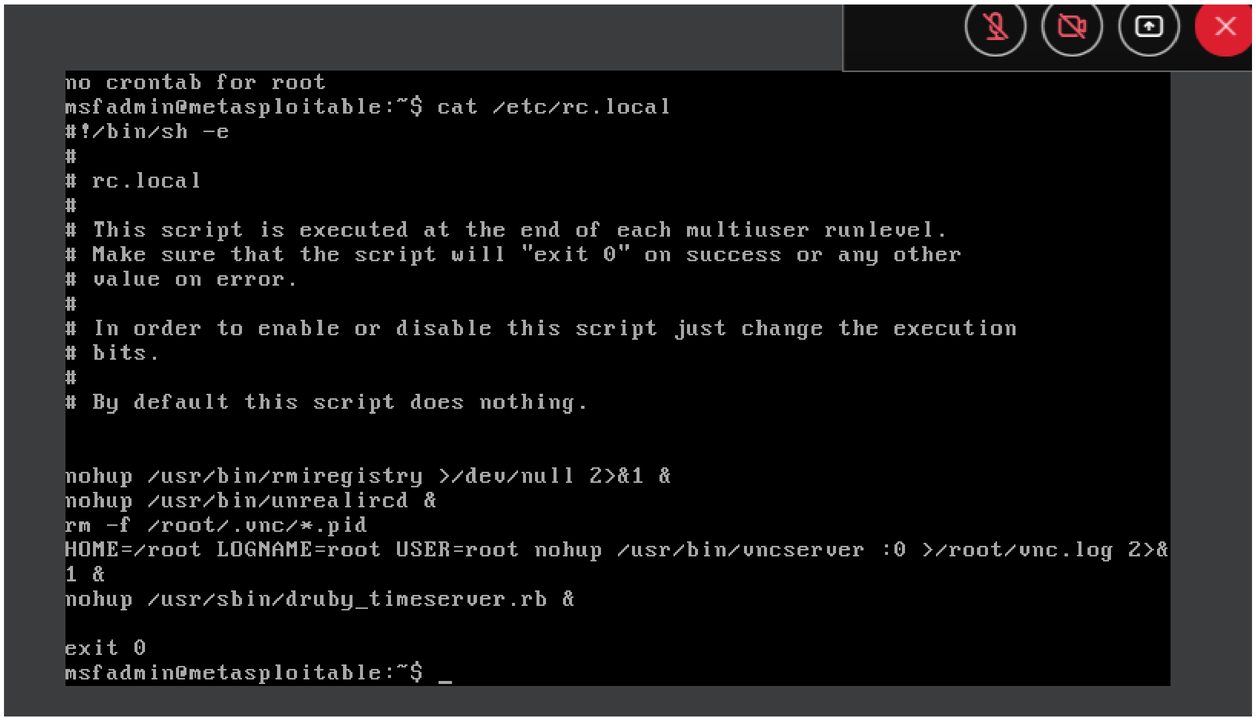


```
msfadmin@metasploitable:~$ sudo netstat -tulnp | grep 1524
tcp        0      0 0.0.0.0:1524          0.0.0.0:*             LISTEN
4282/xinetd
msfadmin@metasploitable:~$ sudo kill -9 4282
msfadmin@metasploitable:~$ sudo netstat -tulnp | grep 1524
msfadmin@metasploitable:~$ _
```

3. **Checked rc.local for Suspicious Activity**: The file /etc/rc.local was reviewed to check for any malicious scripts that might restart the bind shell. No explicit entries related to the bind

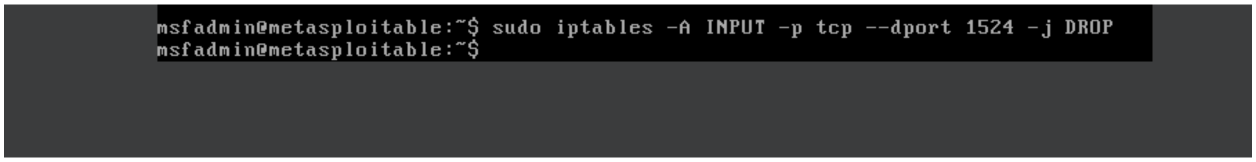
shell were found in the startup script, though there were other processes like VNC server and druby_timeserver.rb.

4. Checked for crontab in order to see some repeating of process.



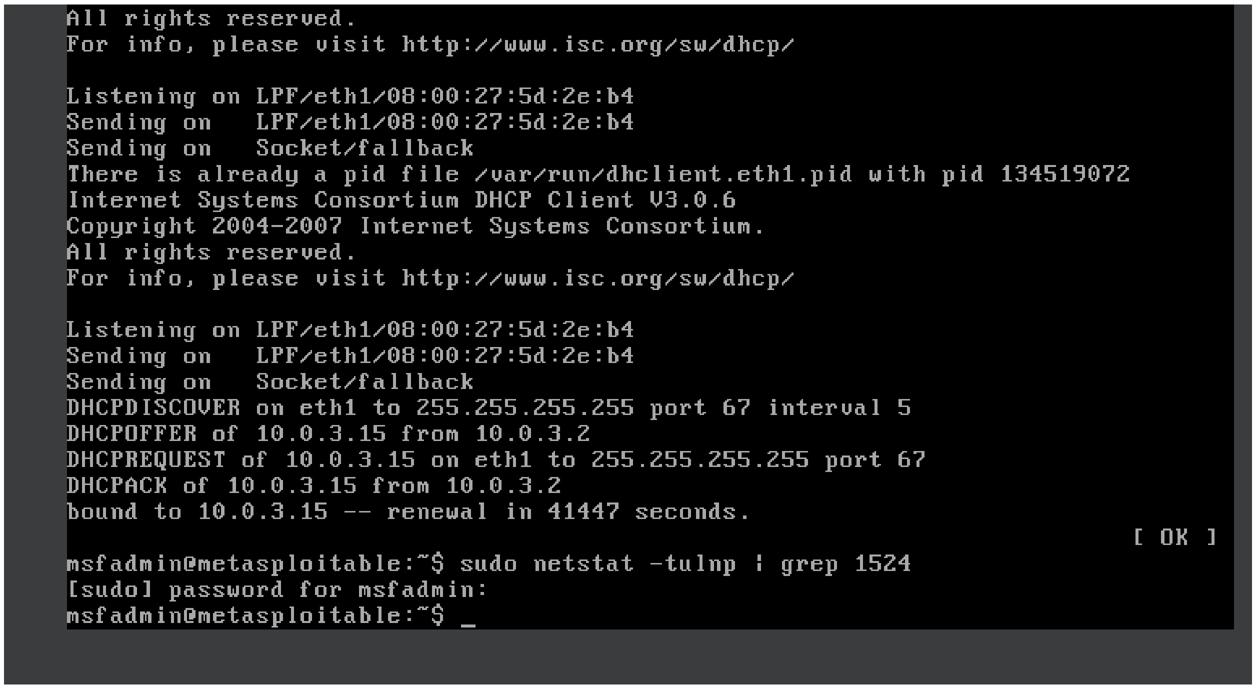
```
no crontab for root
msfadmin@metasploitable:~$ cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

nohup /usr/bin/rmiregistry >/dev/null 2>&1 &
nohup /usr/bin/unrealircd &
rm -f /root/.vnc/*.pid
HOME=/root LOGNAME=root USER=root nohup /usr/bin/vncserver :0 >/root/vnc.log 2>&
1 &
nohup /usr/sbin/druby_timeserver.rb &

exit 0
msfadmin@metasploitable:~$ _
```

5.Blocked access to port 1524



```
msfadmin@metasploitable:~$ sudo iptables -A INPUT -p tcp --dport 1524 -j DROP
msfadmin@metasploitable:~$
```

6. Restarted network manager and checked if process active or not



```
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth1/08:00:27:5d:2e:b4
Sending on   LPF/eth1/08:00:27:5d:2e:b4
Sending on   Socket/fallback
There is already a pid file /var/run/dhclient.eth1.pid with pid 134519072
Internet Systems Consortium DHCP Client V3.0.6
Copyright 2004-2007 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth1/08:00:27:5d:2e:b4
Sending on   LPF/eth1/08:00:27:5d:2e:b4
Sending on   Socket/fallback
DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 5
DHCPOFFER of 10.0.3.15 from 10.0.3.2
DHCPREQUEST of 10.0.3.15 on eth1 to 255.255.255.255 port 67
DHCPACK of 10.0.3.15 from 10.0.3.2
bound to 10.0.3.15 -- renewal in 41447 seconds.
                                                                  [ OK ]
msfadmin@metasploitable:~$ sudo netstat -tulnp | grep 1524
[sudo] password for msfadmin:
msfadmin@metasploitable:~$ _
```

Mitigation:

- **Immediate Action**: The process associated with the bind shell backdoor was killed to prevent unauthorized remote access.
- **Permanent Solution**: The server should be scanned regularly for unauthorized services and processes. System administrators should ensure that no malicious software or backdoors are running at startup.

Future Recommendations:

- Disable any unnecessary services and restrict network access.
- Regularly update and patch the system to minimize exposure to known vulnerabilities.
- Monitor system logs for unusual activity.
- Consider using a firewall to restrict incoming traffic on port 1524 and other non-essential ports.

Conclusion:

The bind shell backdoor on port 1524 was successfully neutralized. However, continuous monitoring is recommended to prevent re-occurrence of such backdoor installations. In future scans, port 1524 should be verified to ensure it remains closed.