5/10/2024

Progetto modulo 3

Effettuare una scansione completa sul target Metasploitable. Scegliete da un minimo di 2 fino ad un massimo di 4 vulnerabilità critiche e provate ad implementare delle azioni di rimedio.

Michele Garzoni

Traccia:

Effettuare una scansione completa sul target Metasploitable.

Scegliete da un minimo di 2 fino ad un massimo di 4 vulnerabilità **critiche** e provate ad **implementare delle azioni di rimedio**.

N.B. le azioni di rimedio, in questa fase, potrebbero anche essere delle regole firewall ben configurate in modo da limitare eventualmente le esposizioni dei servizi vulnerabili.
Vi consigliamo tuttavia di utilizzare magari questo approccio **per non più di una vulnerabilità**.

Per dimostrare l'efficacia delle azioni di rimedio, eseguite nuovamente la scansione sul target e confrontate i risultati con quelli precedentemente ottenuti.

Consegna:

1. Scansione iniziale dove si vede il grafico con tutte le vulnerabilità e le vulnerabilità da risolvere (tecnico, già riassunto) - **ScansioneInizio.pdf**

2. **Screenshot e spiegazione dei passaggi della remediation - RemediationMeta.pdf**

3. Scansione dopo le modifiche che evidenzia la risoluzione dei problemi/vulnerabilità (il grafico che mostra tutte le vulnerabilità) - **ScansioneFine.pdf**

   Oppure un report unico, a vostra scelta. Penso sia più comodo farne tre comunque.

Nota: i report possono essere lasciati in inglese, senza problemi.

Se risolvete le 4 vulnerabilità, potete risolverne una quinta (a scelta), ad esempio con una regola di firewall

Michele G. (CyberSecurity expert)

Kaelyn Security INC

www.kaelynsecinc.com Gkae@kaesec.net

Security Assessment Report for target machine “Metasploitable 2”

Table of Contents

Introduction

Metasploitable2 is a purposely vulnerable virtual machine designed for security testing, specifically for penetration testing and vulnerability assessment. It's a freely available open-source project that provides a simulated environment with intentionally exploitable vulnerabilities. This environment allows cybersecurity professionals, researchers, and students to practice and hone their skills in identifying, exploiting, and securing systems against various common security threats. Metasploitable2 includes a range of vulnerabilities found in real-world systems, making it a valuable tool for learning about cybersecurity principles and techniques in a safe and controlled setting.

After setting up Metasploitable2, conducting the security test took approximately two hours. During this time, various scanning and exploitation tools were employed to assess the vulnerabilities present in the system. The testing process involved identifying weaknesses such as outdated software versions, misconfigured services, default credentials, and known exploits.

Additionally, it's essential to note that the vulnerabilities found during the security assessment of Metasploitable2 are valid as of May 8th, the day when the scan was conducted.

Software used

The software utilized for the Security Assessment (SA) included Nmap, Nessus, and Zap.

Nmap is a powerful network scanner renowned for its versatility in discovering hosts and services on a network.

Nessus is a comprehensive vulnerability scanner, capable of identifying security flaws in systems and applications.

Zap, short for OWASP Zed Attack Proxy, is a tool specifically designed for finding security vulnerabilities in web applications.

Together, these tools provided a comprehensive approach to assessing and enhancing the security posture of the system under review.

Note: in our Security Assessment we only cover **“Critical Issues”** so Nessus report is focused only on these.

ScanMeta1

Report generated by Nessus™ Tue, 07 May 2024 16:12:40 EDT

192.168.1.100

10	6	20	8	77
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 121

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	9.0	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	9.8	-	51988	Bind Shell Backdoor Detection
CRITICAL	9.8	-	20007	SSL Version 2 and 3 Protocol Detection
CRITICAL	10.0	-	171340	Apache Tomcat SEoL (<= 5.5.x)
CRITICAL	10.0	-	33850	Unix Operating System Unsupported Version Detection
CRITICAL	10.0*	5.1	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
CRITICAL	10.0*	5.1	32321	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
CRITICAL	10.0*	5.9	11356	NFS Exported Share Information Disclosure
CRITICAL	10.0*	7.4	46882	UnrealIRCd Backdoor Detection
CRITICAL	10.0*	-	61708	VNC Server 'password' Password
HIGH	8.6	5.2	136769	ISC BIND Service Downgrade / Reflected DoS
HIGH	7.5	-	42256	NFS Shares World Readable
HIGH	7.5	5.1	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5	5.9	90509	Samba Badlock Vulnerability
HIGH	7.5*	5.9	10205	rlogin Service Detection
HIGH	7.5*	5.9	10245	rsh Service Detection
MEDIUM	6.8	6.0	33447	Multiple Vendor DNS Query ID Field Prediction Cache Poisoning
MEDIUM	6.5	3.6	139915	ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS

Identified Vulnerabilities

The vulnerabilities listed below are arranged in order of severity, with the most critical issues appearing first. This prioritization helps focus attention on addressing the most pressing security concerns promptly. By addressing high-severity vulnerabilities first, organizations can mitigate the most significant risks to their systems and data, thereby enhancing overall security posture and resilience against potential cyber threats.

VNC Server 'password' Password (Critical)

CVSS V3.0 score: 10.0

The VNC server on the remote host is secured with a weak password, specifically 'password'. This vulnerability allows a malicious actor to authenticate using this weak password. An attacker could exploit this remotely and without authentication, potentially gaining control of the system. To mitigate this risk, it's crucial to secure the VNC service with a strong, hard-to-guess password.

Suggested solution:

Change the password and secure the VNC service with a stronger password.

UnrealIRCd Backdoor Detection (Critical)

CVSS V3.0 score: 10.0

The remote IRC server is a version of UnrealIRCd with a malicious backdoor that allows an attacker to execute arbitrary code on the affected host.

Suggested solution:

Re-download the software, verify it using the published MD5 / SHA1 checksums, and re-install it.

Useful links:

https://www.rapid7.com/db/modules/exploit/unix/irc/unreal_ircd_3281_backdoor/

https://seclists.org/fulldisclosure/2010/Jun/277

NFS Exported Share Information Disclosure (Critical)

CVSS V3.0 score: 10.0

It is possible to access NFS shares on the remote host, an attacker may be able to leverage this to read (and possibly write) files on remote host.

Suggested solution:

Configure NFS on the remote host so that only authorized hosts can mount its remote shares.

Debian OpenSSH/OpenSSL Package Random Number Generator weakness (SSL check) (Critical)

CVSS V3.0 score: 10.0

The remote SSL certificate uses a weak key.

The remote x509 certificate on the remote SSL server has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to decipher the remote session or set up a man in the middle attack.

Suggested solution:

Consider all cryptographic material generated on the remote host to be guessable. In particuliar, all SSH, SSL and OpenVPN key material should be re-generated.

Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (Critical)

CVSS V3.0 score: 10.0

The remote SSH host keys are weak.

The remote SSH host key has been generated on a Debian or Ubuntu system which contains a bug in the random number generator of its OpenSSL library.

The problem is due to a Debian packager removing nearly all sources of entropy in the remote version of OpenSSL.

An attacker can easily obtain the private part of the remote key and use this to set up decipher the remote session or set up a man in the middle attack.

Suggested solution:

Consider all cryptographic material generated on the remote host to be guessable. In particuliar, all SSH, SSL and OpenVPN key material should be re-generated.

Useful links:

http://www.nessus.org/u?107f9bdc

http://www.nessus.org/u?f14f4224

Unix Operating System Unsupported Version Detection (Critical)

CVSS V3.0 score: 10.0

The operating system running on the remote host is no longer supported.

According to its self-reported version number, the Unix operating system running on the remote host is no longer supported.

Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it is likely to contain security vulnerabilities.

Suggested solution:

Upgrade to a version of the Unix operating system that is currently supported.

Apache Tomcat SEoL (<= 5.5.x) (Critical)

CVSS V3.0 score: 10.0

According to its version, Apache Tomcat is less than or equal to 5.5.x. It is, therefore, no longer maintained by its vendor or provider.

Lack of support implies that no new security patches for the product will be released by the vendor. As a result, it may contain security vulnerabilities.

Suggested solution:

Upgrade to a version of Apache Tomcat that is currently supported.

Useful links:

https://tomcat.apache.org/tomcat-55-eol.html

SL Version 2 and 3 Protocol Detection (Critical)

CVSS V3.0 score: 9.8

The remote service encrypts traffic using a protocol with known weaknesses.

The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:

- An insecure padding scheme with CBC ciphers.
- Insecure session renegotiation and resumption schemes.

An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.

Although SSL/TLS has a secure means for choosing the highest supported version of the protocol (so that these versions will be used only if the client or server support nothing better), many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as in POODLE). Therefore, it is recommended that these protocols be disabled entirely.

NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC's definition of 'strong cryptography'.

Suggested solution:

Consult the application's documentation to disable SSL 2.0 and 3.0.
Use TLS 1.2 (with approved cipher suites) or higher instead.

Useful links:

https://www.schneier.com/academic/paperfiles/paper-ssl.pdf

https://www.openssl.org/~bodo/ssl-poodle.pdf

https://www.imperialviolet.org/2014/10/14/poodle.html

Bind Shell Backdoor Detection (Critical)

CVSS V3.0 score: 9.8

A shell is listening on the remote port without any authentication being required. An attacker may use it by connecting to the remote port and sending commands directly.

Suggested solution:

Verify if the remote host has been compromised, and reinstall the system if necessary.

Another solution could be to kill the process and to remove it from the host machine

Apache Tomcat AJP Connector Request Injection (Ghostcat) (Critical)

CVSS V3.0 score: 9.8

There is a vulnerable AJP connector listening on the remote host.

A file read/inclusion vulnerability was found in AJP connector. A remote, unauthenticated attacker could exploit this vulnerability to read web application files from a vulnerable server. In instances where the vulnerable server allows file uploads, an attacker could upload malicious JavaServer Pages (JSP) code within a variety of file types and gain remote code execution (RCE).

Suggested solution:

Update the AJP configuration to require authorization and/or upgrade the Tomcat server to 7.0.100, 8.5.51, 9.0.31 or later.

Useful links:

https://access.redhat.com/security/cve/CVE-2020-1745

https://access.redhat.com/solutions/4851251

In conclusion, the vulnerability assessment conducted on the Metasploitable2 machine revealed several significant vulnerabilities that could be exploited by malicious attackers to compromise the system. Through in-depth analysis of the identified vulnerabilities, solutions and countermeasures have been proposed to mitigate the security risks. However, it is important to emphasize that cybersecurity is a continuous and evolving process, therefore it is advisable to implement constant monitoring and regular updates to ensure effective protection against current and future threats.

Signature

Exploitation and suggested remediation

As part of the comprehensive security assessment conducted on the Metasploitable2 machine, several vulnerabilities were identified that require immediate attention and remediation to bolster the system's security posture. To address these vulnerabilities effectively, a systematic approach will be adopted to ensure thorough mitigation measures are implemented.

Firstly, a prioritization matrix will be established based on the severity and potential impact of each vulnerability. This will enable the identification of high-risk vulnerabilities that pose the most significant threat to the system's integrity and confidentiality. Vulnerabilities will be categorized into critical, high, medium, and low risk, allowing for a structured remediation plan. **In our report we will focus on critical level vulnerabilities only.**

Following the prioritization, a detailed remediation plan will be developed for each identified vulnerability. This plan will outline specific actions required to address and mitigate the vulnerabilities effectively. Depending on the nature of the vulnerability, remediation strategies may include patching vulnerable software, disabling unnecessary services, implementing access controls, and enhancing network security configurations.

To ensure the effectiveness of the remediation efforts, a rigorous testing process will be conducted after implementing each remediation measure. This testing phase will involve vulnerability scanning tools such as Nessus and OpenVAS to validate that the identified vulnerabilities have been successfully mitigated and no new vulnerabilities have been introduced.

Furthermore, continuous monitoring and maintenance will be implemented to sustain the security improvements achieved through remediation efforts. Regular vulnerability assessments and penetration testing will be conducted to proactively identify and address any emerging security threats or weaknesses in the system.

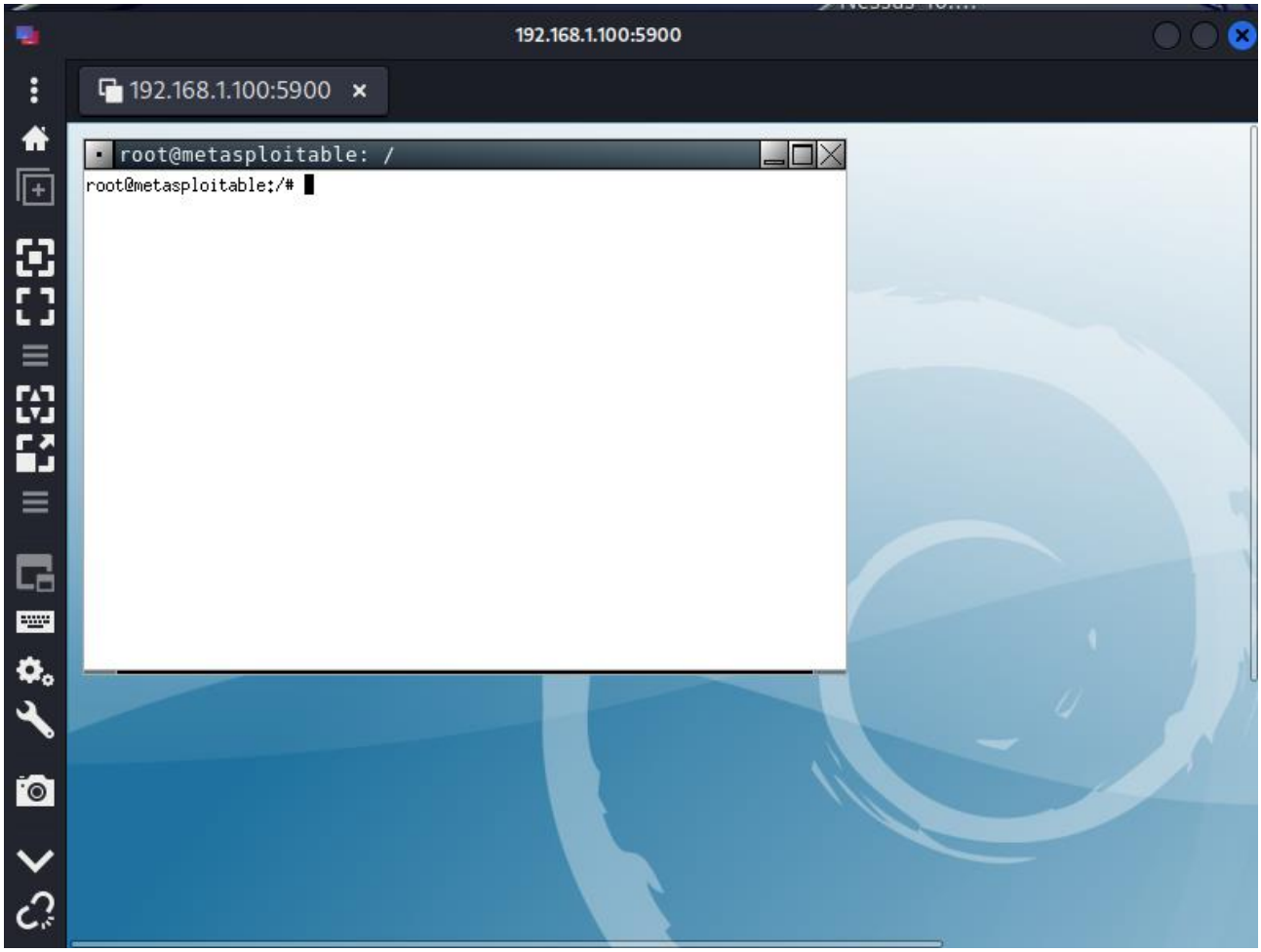
Additionally, employee training and awareness programs will be organized to educate users about best practices for maintaining a secure computing environment and recognizing potential security risks.

VNC Server 'password' Password (Critical)

CVSS V3.0 score: 10.0

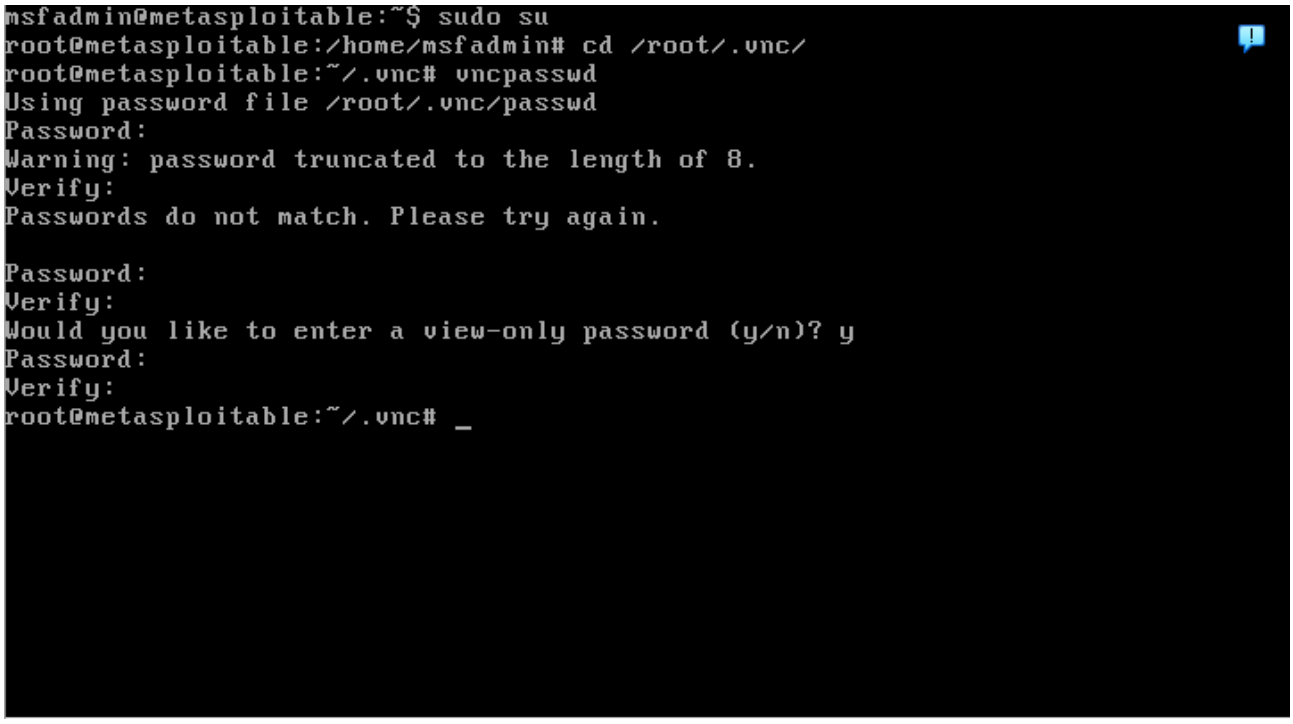
Steps to exploit the vulnerability:

1) Download and install Remmina on Kali Linux
2) Select “VNC” and connect to the server IP on port 5900
3) Enter “password” as password
4) Find a root privilege shell on the target machine



Recommended Remediation:

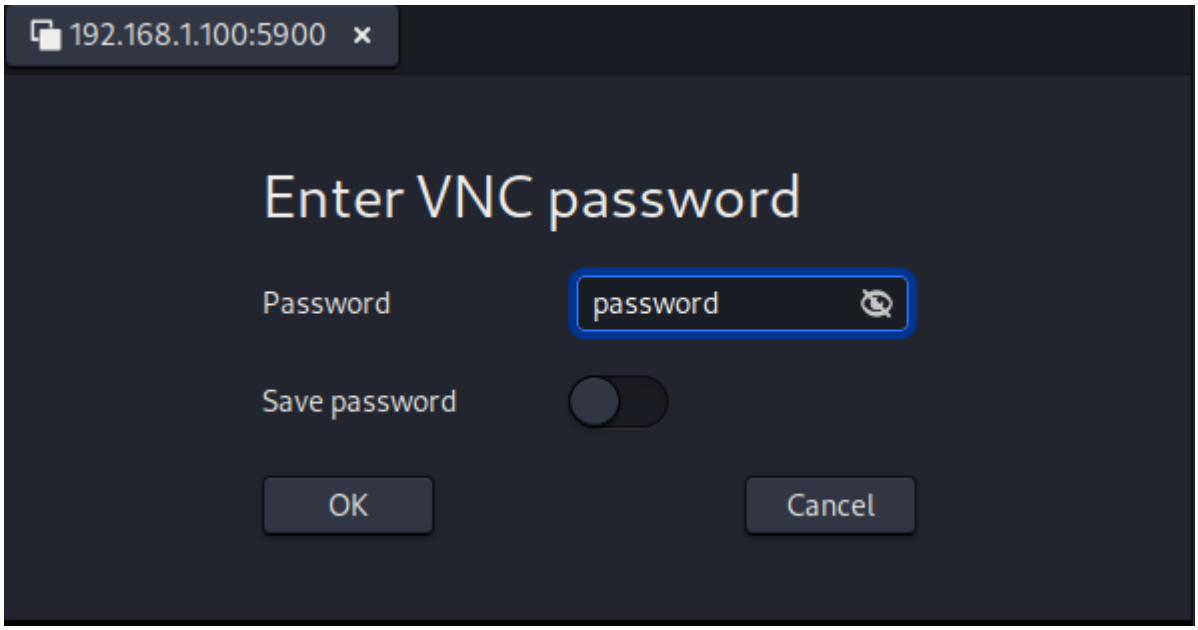
1) Enter root in the metaspoitable2 machine
2) Access folder /root/.vnc
3) Change password with vncpasswd with a secure password
4) Verify that the new password matches and protect it.

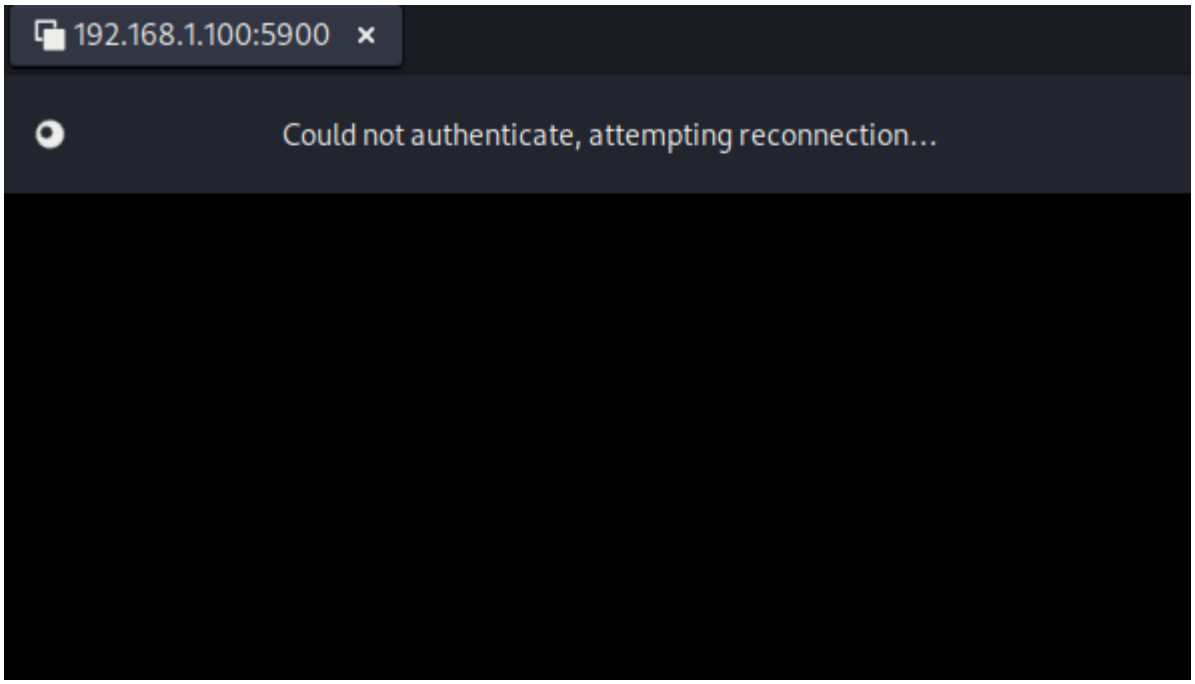


For enhanced security, I suggest considering additional measures alongside the suggested remediation steps:

Use firewall rules to restrict access to the VNC server port (on this machine port 5900) to only trusted IP addresses or networks.

After the remediation the VNC server is not accessible anymore with a weak or default password



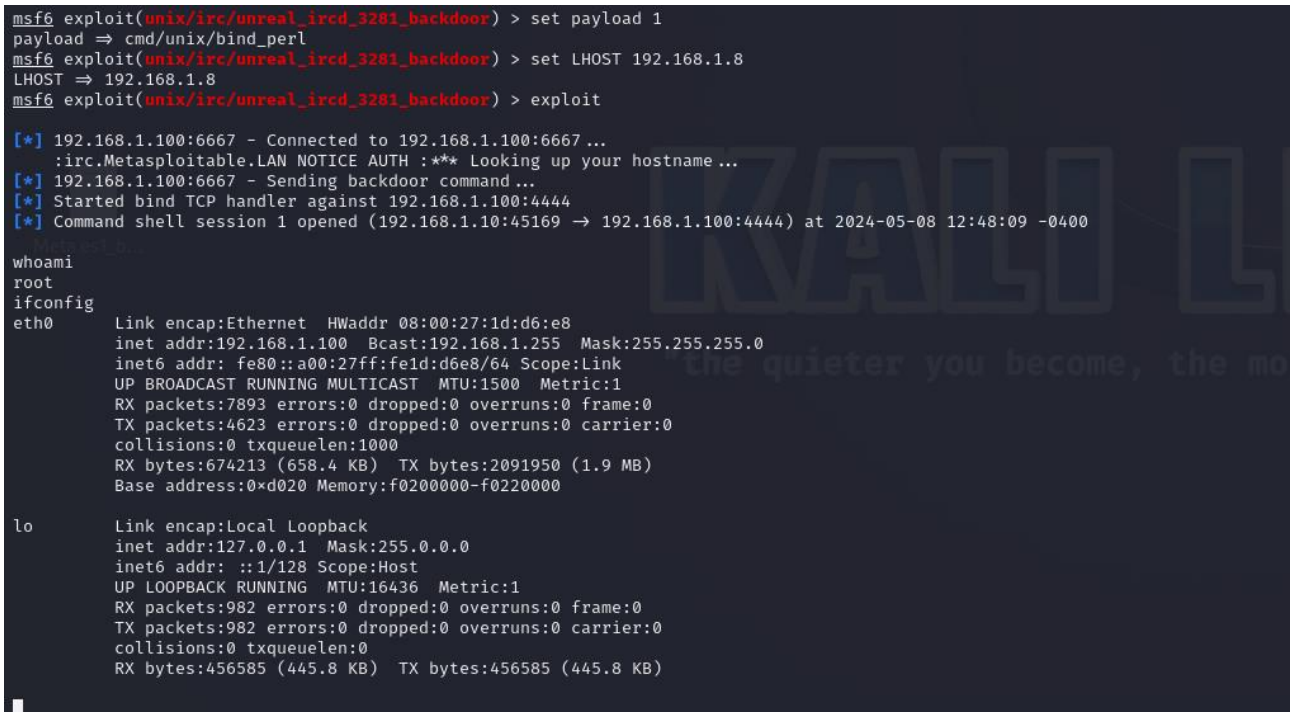


UnrealIRCd Backdoor Detection (Critical)

CVSS V3.0 score: 10.0

Steps to exploit the vulnerability:

1) Open Metasploit Framework on Kali Linux
2) Load exploit “unreal_ircd_3281_backdoor”
3) Set RHOST to the target IP machine
4) Set PAYLOAD type (bind_perl Reverse Shell)
5) Set LHOST to the IP of the attacker
6) Run the exploit
7) Gain a shell with root privilege on the target machine

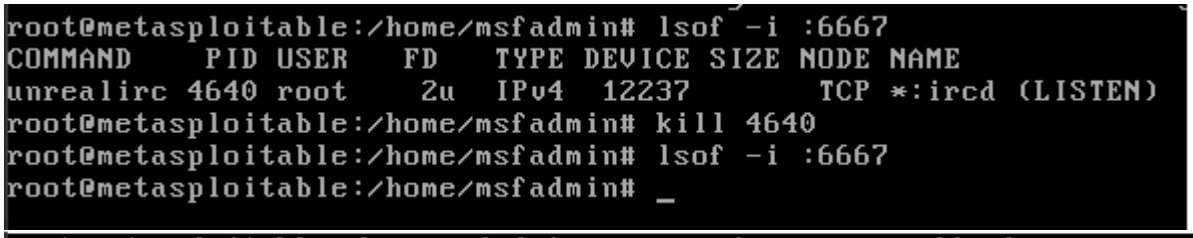


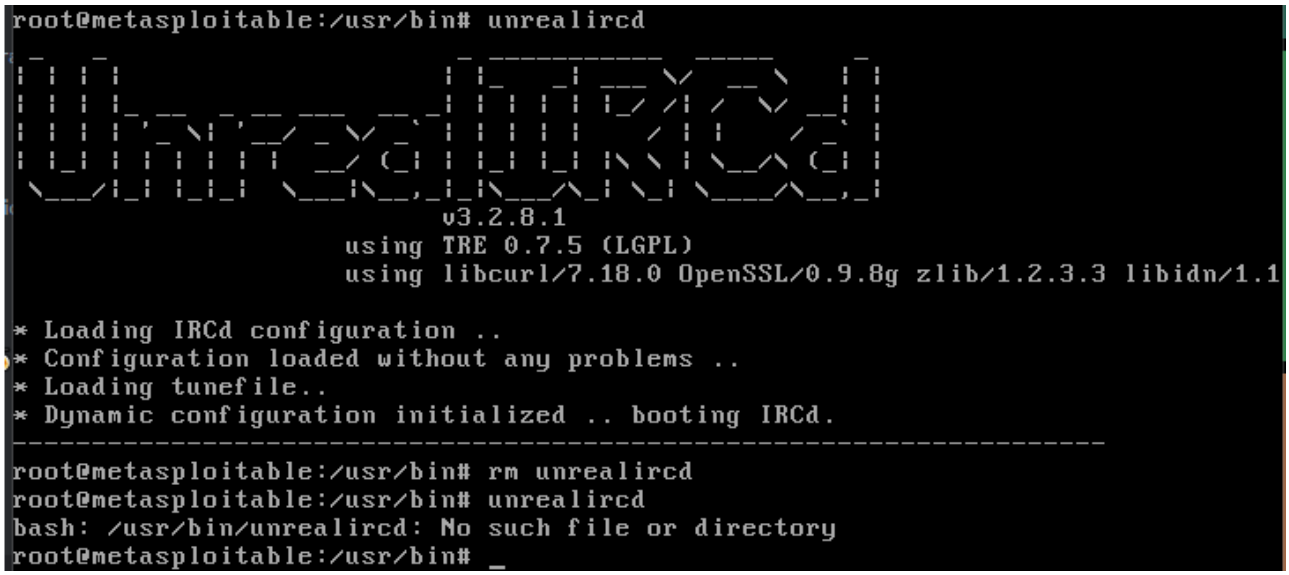
Recommended Remediation:

1) Login root on metaspolitable2
2) Isof -I: 6667 for identify the process on the port 6667
3) Kill ID (4640) for terminate the process
4) Move to /usr/bin
5) Remove unrealircd service from the machine with rm
6) Reboot the machine

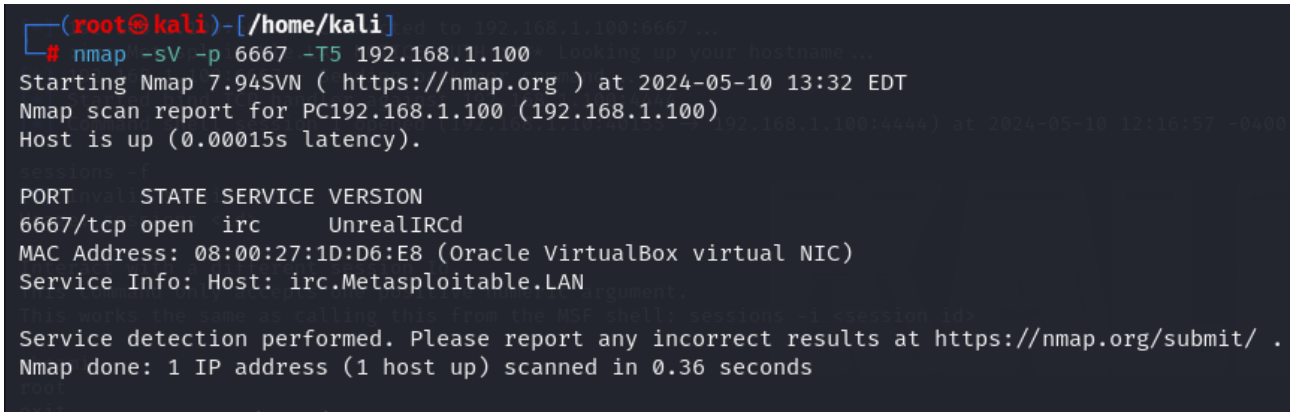
The removal of the UnrealIRCd service from the Metasploitable2 machine is needed due to the impossibility of updating the service and removing the backdoor in the service itself. Due to the service being a IRC server, there is no way to use firewall for controlling the port traffic.

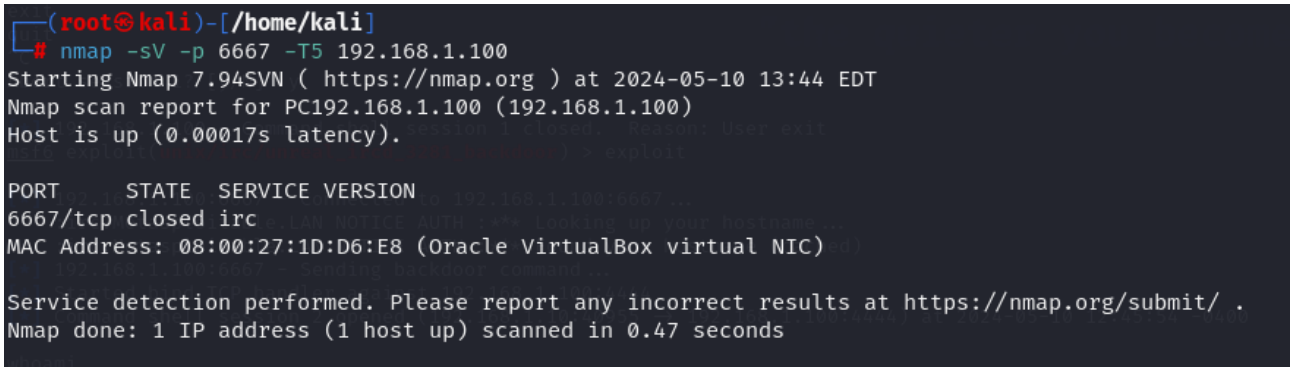
I suggest you to install a different version if an IRC server is needed





Service has been successfully terminated and the port is not anymore vulnerable to the exploit after the reboot.





NFS Exported Share Information Disclosure (Critical)

CVSS V3.0 score: 10.0

Steps to exploit the vulnerability:

1) **Ssh-keygen in terminal**
2) **mount a local folder to the NFS server of the target machine.**
3) **Add our ssh key in the authorized keys**
4) **Umount the key**
5) **Gain Root access to the machine**

```
kali@kali:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kali/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kali/.ssh/id_rsa
Your public key has been saved in /home/kali/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:hUQ+wqOi08TuljYmT0fvd8W5iqmi7wZnadnZYrY1/2I kali@kali
```

```
root@kali:~# cd ../home/kali/
root@kali:/home/kali# cat .ssh/id_rsa.pub >> /tmp/sshkey/root/.ssh/authorized_keys
root@kali:/home/kali# umount /tmp/sshkey
```

```
kali@kali:~$ ssh root@192.168.1.100
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/
You have new mail.
root@metasploitable:~#
```

Suggested solution:

Set iptables command to block any external connection as long as the NFS server is not configured and then Add authentication before sending files / folders.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -p 2049 192.168.1.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-10 17:48 EDT
Nmap scan report for PC192.168.1.100 (192.168.1.100)
Host is up (0.00021s latency).

PORT     STATE  SERVICE VERSION
2049/tcp closed nfs
```

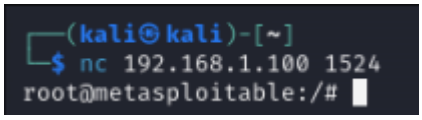
Any attempt to mount now will be frozen due to the iptables refusing the connection

Bind Shell Backdoor Detection (Critical)

CVSS V3.0 score: 9.8

Steps to exploit the vulnerability:

1) **Connect by netcat to target IP on port 1524**
2) **Obtain access to a root shell**



Recommended Remediation:

1) Login as root \ Sudo
2) Type sudo update-rc.d -f xinetd remove
3) Reboot the system
4) Sudo lsof -i :1524 for check if the service is listening on the port

Due to the fact that the shell is started by xinetd, I recommend to disable all xinetd configurations on the system. This action is going to remove the risk of a malicious shell being spawned at startup via the xinetd service.

```
msfadmin@metasploitable:/etc/rc0.d$ sudo update-rc.d -f xinetd remove
 Removing any system startup links for /etc/init.d/xinetd ...
   /etc/rc0.d/K20xinetd
   /etc/rc1.d/K20xinetd
   /etc/rc2.d/S20xinetd
   /etc/rc3.d/S20xinetd
   /etc/rc4.d/S20xinetd
   /etc/rc5.d/S20xinetd
   /etc/rc6.d/K20xinetd
```

```
msfadmin@metasploitable:~$ sudo update-rc.d -f xinetd remove
[sudo] password for msfadmin:
 Removing any system startup links for /etc/init.d/xinetd ...
msfadmin@metasploitable:~$ lsof -i :1524
msfadmin@metasploitable:~$
```

Connection is no longer possible and port is closed

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -p 1524 192.168.1.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-10 17:30 EDT
Nmap scan report for PC192.168.1.100 (192.168.1.100)
Host is up (0.00019s latency).

PORT     STATE  SERVICE   VERSION
1524/tcp closed ingreslock

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds

┌──(kali㉿kali)-[~]
└─$ nc 192.168.1.100 1524
(UNKNOWN) [192.168.1.100] 1524 (ingreslock) : Connection refused

┌──(kali㉿kali)-[~]
└─$
```

For enhanced security, I suggest considering additional measures alongside the suggested remediation steps:

Control all the inetd config files and find the file spawning the shell on that port, then remove the file. Probably check also the /etc/inetd.conf for any unauthorized changes

Set a rule on the firewall to block all connections to port 1524 until the problem is fixed.

Various FTP / Services « R » – Port 512/513/514

There is a backdoor on FTP server and the ports 512,513,514 are opened. We use iptables for drop the incoming connections and secure the ports.

Recommended Remediation:

Due to the fact that the machine comes without the possibility to update the software through console we resolve the problem with iptables string as follow:

“ **iptables -A INPUT -p tcp -m multiport --dports 21,512:514 -j DROP** ”

“

```
root@metasploitable:/etc# sudo iptables -A INPUT -p tcp --dport 21 -j DROP
root@metasploitable:/etc# sudo iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere            tcp dpt:ftp
root@metasploitable:/etc# _
```

Before the string

```
└─# nmap -sV -T5 192.168.1.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-10 12:37 EDT
Nmap scan report for PC192.168.1.100 (192.168.1.100)
Host is up (0.000057s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
```

After the string

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV -p 21 -T5 192.168.1.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-10 12:41 EDT
Nmap scan report for PC192.168.1.100 (192.168.1.100)
Host is up (0.00016s latency).

PORT   STATE    SERVICE VERSION
21/tcp filtered ftp
MAC Address: 08:00:27:1D:D6:E8 (Oracle VirtualBox virtual NIC)
```

For enhanced security, I suggest considering additional measures alongside the suggested remediation steps:

Manually update the vsftpd to its last version: **vsftpd 3.0.3**

We also add ports from 512 to 514 because these TCP ports 512, 513 and 514 are known as "r" services which can allow an attacker to enter the system if they are incorrectly configured. RSH Remote Shell services (**rsh, rexec, and rlogin**) are active.

```
kali@kali:~$ rlogin -l root 192.168.10.100
Last login: Mon Dec  7 09:53:04 EST 2020 from :0.0 on pts/0
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/
You have new mail.
root@metasploitable:~# whoami
root
```

Metapostfics

Report generated by Nessus™ Fri, 10 May 2024 18:01:36 EDT

192.168.1.100

6	4	19	8	71
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 108

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	9.0	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	9.8	-	20007	SSL Version 2 and 3 Protocol Detection
CRITICAL	10.0	-	171340	Apache Tomcat SEoL (<= 5.5.x)
CRITICAL	10.0	-	33850	Unix Operating System Unsupported Version Detection
CRITICAL	10.0*	5.1	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness
CRITICAL	10.0*	5.1	32321	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)
HIGH	8.6	5.2	136769	ISC BIND Service Downgrade / Reflected DoS
HIGH	7.5	-	42256	NFS Shares World Readable
HIGH	7.5	5.1	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5	5.9	90509	Samba Badlock Vulnerability
MEDIUM	6.8	6.0	33447	Multiple Vendor DNS Query ID Field Prediction Cache Poisoning
MEDIUM	6.5	3.6	139915	ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS
MEDIUM	6.5	-	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.5	-	57582	SSL Self-Signed Certificate
MEDIUM	6.5	-	104743	TLS Version 1.0 Protocol Detection
MEDIUM	5.9	4.4	136808	ISC BIND Denial of Service
MEDIUM	5.9	4.4	31705	SSL Anonymous Cipher Suites Supported