

PROGETTO MODULO 3

○ Remediation Metasploitable

Prima di iniziare a risolvere le vulnerabilità riscontrate nella scansione effettuata con nessuss, effettuiamo una scansione manuale da terminale utilizzando Nmap.

- `nmap -sX -sV 192.168.6.3`
- 1. in questo tipo di scansione utilizziamo `-sX` per inviare pacchetti TCP con i flag FIN, PSH e URG , ma senza il flag SYN. Viene utilizzata per analizzare la topologia di rete e individuare eventuali host vulnerabili, inoltre non tenta di stabilire una connessione TCP completa.
- 2. Lo switch `-sV` invece tenta di determinare le versioni dei servizi in esecuzione su porte aperte.

```
(kali@kali)~$ sudo nmap -sX -sV -p- 192.168.6.3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-12 20:26 EDT
Nmap scan report for 192.168.6.3
Host is up (0.00029s 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?
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)
445/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    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ccproxy-ftp?
3306/tcp  open  mysql?
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/drbc)
43689/tcp open  nlockmgr     1-4 (RPC #100021)
44253/tcp open  mountd       1-3 (RPC #100005)
53895/tcp open  java-rmi     GNU Classpath grmiregistry
54029/tcp open  status       1 (RPC #100024)
MAC Address: 08:00:27:72:6F:96 (Oracle VirtualBox virtual NIC)
Service Info: Host: 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/ .
Nmap done: 1 IP address (1 host up) scanned in 199.42 seconds

(kali@kali)~$
```

1. La prima vulnerabilità che andiamo a risolvere riguarda BIND SHELL BACKDOOR DETECTION

CRITICAL

9.8

-

51988

Bind Shell Backdoor Detection

Questa vulnerabilità ci permette di poterci collegare alla porta 1524 con il comando `nc 192.168.6.3 1524` della macchina metasploitable facendoci ottenere l'accesso ad una connessione root senza che ci venga richiesta l'autenticazione.

```
(kali㉿kali)-[~]
$ netcat 192.168.6.3 1524
root@metasploitable:/# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:70:24:f2
          inet addr:192.168.6.3  Bcast:192.168.6.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe70:24f2/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3147 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2898 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:254645 (248.6 KB)  TX bytes:224782 (219.5 KB)
          Base address:0xd020  Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:423 errors:0 dropped:0 overruns:0 frame:0
          TX packets:423 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:94753 (92.5 KB)  TX bytes:94753 (92.5 KB)
```

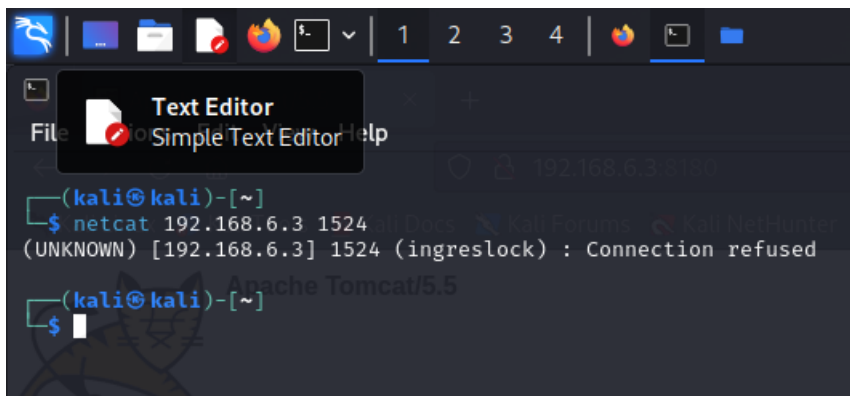
Per risolvere questa vulnerabilità andiamo a modificare in metasploitable il file `/etc/inetd.conf` utilizzando l'editor di testo nano commentando la riga che sta per ultima : **#ingreslock stream tcp nowait root /bin/bash bash -i**

```
GNU nano 2.0.7      File: /etc/inetd.conf

#<off># netbios-ssn      stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sb$
telnet                  stream  tcp      nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.te$
#<off># ftp              stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sb$
tftp                   dgram  udp      wait    nobody  /usr/sbin/tcpd  /usr/sbin/in.tf$
shell                  stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rs$
login                  stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rl$
exec                   stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/in.re$
#ingreslock stream tcp nowait root /bin/bash bash -i

[ Read 8 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Se proviamo a collegarci alla porta 1524 dopo aver effettuato la modifica, notiamo che la connessione è chiusa.



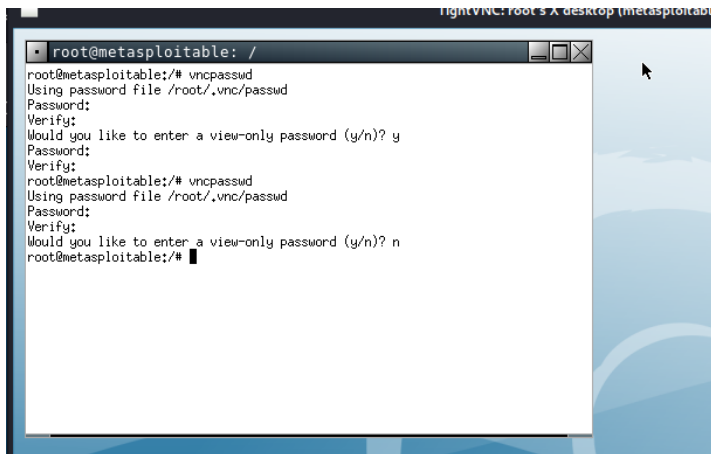
2. La seconda vulnerabilità che andremo a risolvere riguarda VNC SERVER

CRITICAL 10.0* - 61708 VNC Server 'password' Password

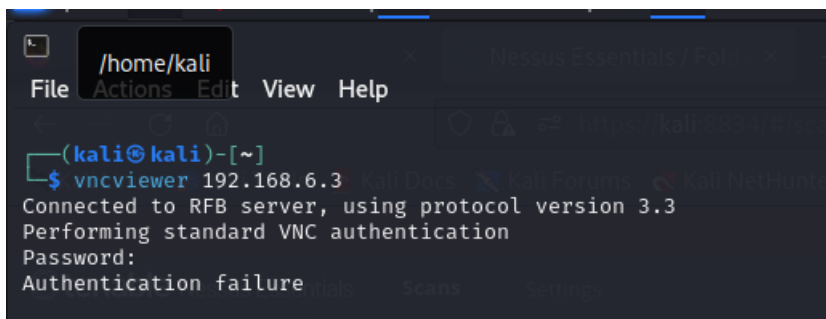
Questa vulnerabilità consente ad un attaccante di poter accedere al server VNC semplicemente utilizzando il comando vncviewer 192.168.6.3 e utilizzando come credenziali “password – password”

```
(kali@kali)~$ vncviewer 192.168.6.3
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication successful
Desktop name "root's X desktop (metasploitable:0)"
VNC server default format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
Using default colormap which is TrueColor. Pixel format:
  32 bits per pixel.
  Least significant byte first in each pixel.
  True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
(kali@kali)~$
```

Per risolvere questa vulnerabilità impostiamo una nuova password più sicura direttamente dall’editor grafico di vnc su metasploitable utilizzando i privilegi root .



Se proviamo a connetterci nuovamente con lo stesso comando e con le credenziali password – password ci accorgiamo che l'autenticazione fallisce perché abbiamo impostato una password più sicura.



3. terza vulnerabilità che andremo a risolvere riguarda NFS EXPORTED SHARE INFORMATION DISCLOURE

CRITICAL 10.0* 5.9 11356 NFS Exported Share Information Disclosure

Nfs è un protocollo di rete che consente ad un pc client di utilizzare la rete per accedere a directory condivise tramite un punto di montaggio da server remoti come se fossero disponibili in locale. In questo caso abbiamo un accesso libero ad un punto di montaggio , che permette all’attaccante di leggere e scrivere sulla nostra macchina.

- Per risolvere questo problema utilizzeremo il comando **rpcinfo 192.168.6.3** per capire su quali porte è in ascolto nfs

```
(kali@kali)-[~]
$ rpcinfo 192.168.6.3
program version netid address service owner
100000 2 tcp 0.0.0.0.0.111 portmapper unknown
100000 2 udp 0.0.0.0.0.111 portmapper unknown
100024 1 udp 0.0.0.0.142.175 status unknown
100024 1 tcp 0.0.0.0.158.73 status unknown
100003 2 udp 0.0.0.0.8.1 nfs unknown
100003 3 udp 0.0.0.0.8.1 nfs unknown
100003 4 udp 0.0.0.0.8.1 nfs unknown
100021 1 udp 0.0.0.0.144.199 nlockmgr unknown
100021 3 udp 0.0.0.0.144.199 nlockmgr unknown
100021 4 udp 0.0.0.0.144.199 nlockmgr unknown
100003 2 tcp 0.0.0.0.8.1 nfs unknown
100003 3 tcp 0.0.0.0.8.1 nfs unknown
100003 4 tcp 0.0.0.0.8.1 nfs unknown
100021 1 tcp 0.0.0.0.130.195 nlockmgr unknown
100021 3 tcp 0.0.0.0.130.195 nlockmgr unknown
100021 4 tcp 0.0.0.0.130.195 nlockmgr unknown
100005 1 udp 0.0.0.0.134.162 mountd unknown
100005 1 tcp 0.0.0.0.229.236 mountd unknown
100005 2 udp 0.0.0.0.134.162 mountd unknown
100005 2 tcp 0.0.0.0.229.236 mountd unknown
100005 3 udp 0.0.0.0.134.162 mountd unknown
100005 3 tcp 0.0.0.0.229.236 mountd unknown
```

Utilizzando il comando **showmount -e 192.168.6.3** otteniamo informazioni sul server nfs.

```
(kali@kali)-[~]
$ showmount -e 192.168.6.3
Export list for 192.168.6.3:
/ *
```

In questo caso le informazioni che ci viene fornita è che un eventuale attaccante potrebbe tranquillamente montare sulla propria macchina **/** che sta ad indicare il file system con proprietà root per la maggior parte dalle macchine linux.

Per risolvere questa vulnerabilità andiamo a modificare il file **/etc/exports**, commentiamo l'ultima riga del file e abilitiamo l'esempio che ci mostra il file aggiungendo l'ip di metasploitable.

```
GNU nano 2.0.7 File: /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes 192.168.6.3(rw,sync) hostname2(ro,sync)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt)
# /srv/nfs4/homes gss/krb5i(rw,sync)
#
#/*(rw,sync,no_root_squash,no_subtree_check)
```

4. La quarta vulnerabilità che risolveremo è **APACHE TOMCAT AJP**

Per risolvere questa vulnerabilità andiamo direttamente nel terminale della macchina metasploitable e configuriamo il file `/etc/tomcat5.5/server.xml` utilizzando l'editor nano con privilegi root.

Con questo metodo andiamo ad abilitare il connettore AJP in modo che venga avviato con fattore di sicurezza.

Dentro il file `/etc/tomcat5.5/server.xml` andiamo ad aggiungere la stringa **`secretRequired="true"`**, accanto alle altre sulla porta 8009, successivamente aggiungiamo un'altra stringa **`secret="<string>"`**

```
GNU nano 2.0.7 File: /etc/tomcat5.5/server.xml

noCompressionUserAgents="gozilla, traviata"
compressableMimeType="text/html,text/xml"

-->

<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->
<!--
<Connector port="8443" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" />
-->

<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009"
enableLookups="false" secretRequired="true" redirectPort="8443" $
<!-- Define a Proxied HTTP/1.1 Connector on port 8082 -->
<!-- See proxy documentation for more information about using this. -->
<!--

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```

```
GNU nano 2.0.7 File: /etc/tomcat5.5/server.xml

noCompressionUserAgents="gozilla, traviata"
compressableMimeType="text/html,text/xml"

-->

<!-- Define a SSL HTTP/1.1 Connector on port 8443 -->
<!--
<Connector port="8443" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" />
-->

<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009"
$8443" protocol="AJP/1.3" secret="<string>" />

<!-- Define a Proxied HTTP/1.1 Connector on port 8082 -->
<!-- See proxy documentation for more information about using this. -->
<!--

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^U Next Page ^U UnCut Text ^T To Spell
```

