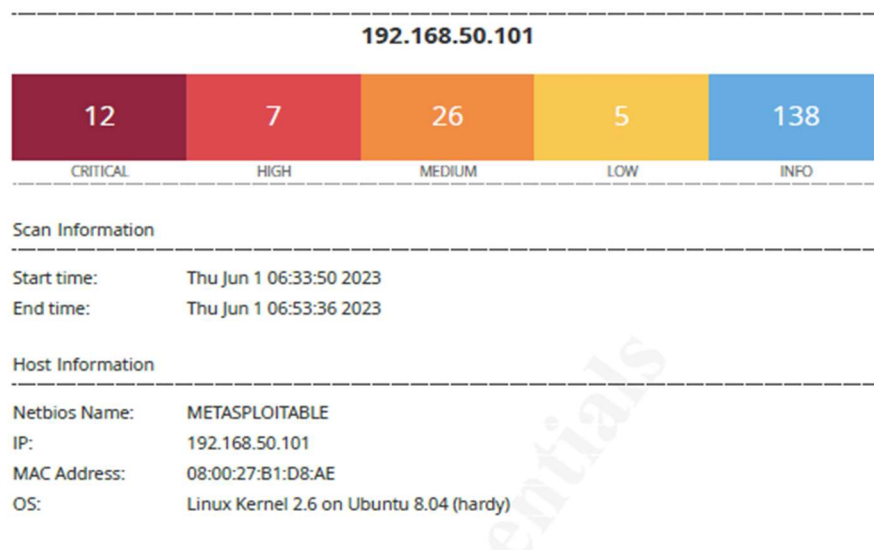


Unit 2: “Remediation actions su Metasploitable 2”

Prima scansione con Nessus:

Con la prima scansione di nessus sono state evidenziate 12 vulnerabilità di livello **critical**. In questo report andremo ad analizzare i procedimenti per il risolvimento per alcune di esse.



1. La prima vulnerabilità risolta è Debian OpenSSH/OpenSSL Package Random Number Generator Weakness.

CRITICAL Debian OpenSSH/OpenSSL Package Random Number Generator Weakness

Description
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.

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

See Also
<http://www.nessus.org/u?107f9bdc>
<http://www.nessus.org/u?f14f4224>

Output

No output recorded.

To see debug logs, please visit individual host

Port ▲	Hosts
22 / tcp / ssh	192.168.50.101

Per questo caso utilizzeremo una regola del firewall. Grazie alla scansione possiamo notare che il processo in questione si trova sulla porta 22 aperta. Per questo dalla macchina Metasploitable 2, con privilegi da superuser (comando *sudo* o come nel mio caso *sudo su*), creiamo la seguente regola per rigettare il traffico in entrata sulla suddetta porta:

iptables -I INPUT -p tcp --dport 22 -j DROP

```
Metasploitable2 [in esecuzione] - Oracle VM VirtualBox
root@metasploitable:~# iptables -I INPUT -p tcp --dport 22 -j DROP
root@metasploitable:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination          tcp dpt:ssh
DROP      tcp  --  anywhere              anywhere             tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@metasploitable:~#
```

2. Risolveremo adesso la vulnerabilità **NFS Exported Share Information Disclosure**.

Vulnerabilities 8

CRITICAL NFS Exported Share Information Disclosure

Description
At least one of the NFS shares exported by the remote server could be mounted by the scanning host. An attacker may be able to leverage this to read (and possibly write) files on remote host.

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

Output
The following NFS shares could be mounted :
+ /
+ Contents of / :
- .
- ..
- bin
- boot
- cdrom
more...
To see debug logs, please visit Individual host

Port ▲	Hosts
2049 / udp / rpc-nfs	192.168.50.101

La solution ci suggerisce di configurare la macchina in modo che solo gli host autorizzati possano accedere da remoto tramite la porta 2049 al servizio nfs, che permette di accedere alle NFS directory (che nel nostro caso sono quelle riportate nel campo Output nell'immagine) e volendo anche di leggere e scrivere dati.

Procediamo quindi modificando il file exports presente nella directory /etc commentando semplicemente l'ultima riga del file, che contiene la regola per la condivisione della cartella (e del suo contenuto).

```
Metasploitable2 [In esecuzione] - Oracle VM VirtualBox
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 hostname1(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)
```

3. e 4. In questo punto risolveremo insieme 2 vulnerabilità allo stesso momento, parliamo di **Bind Shell Backdoor Detection** e di **Rexecd Service Detection**.

CRITICAL rexecd Service Detection

Description
The rexecd service is running on the remote host. This service is design to allow users of a network to execute commands remotely. However, rexecd does not provide any good means of authentication, so it may be abused by an attacker to scan a third-party host.

Solution
Comment out the 'exec' line in /etc/inetd.conf and restart the inetd process.

Output
No output recorded.
To see debug logs, please visit individual host

Port ▲	Hosts
512 / tcp / rexecd	192.168.50.101

CRITICAL Bind Shell Backdoor Detection

Description
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.

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

Output
Nessus was able to execute the command "id" using the following request :

This produced the following truncated output (limited to 10 lines) :
----- snip -----
root@metasploitable:/# uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#
----- snip -----
To see debug logs, please visit individual host

Port ▲	Hosts
1524 / tcp / wild_shell	192.168.50.101

Bind Shell Backdoor Detection è un servizio sulla porta 1524 che permette di accedere al sistema da remoto in incognito e senza necessità di autenticazione.

Rexecd Service Detection sulla porta 512 è un servizio che permette agli utenti di una rete di eseguire comandi da remoto.

Procediamo a rimuovere entrambe le vulnerabilità aprendo il file *inetd.conf* presente nella directory */etc* e commentiamo la riga “exec” per quanto riguarda il servizio Rexecd e la riga “ingreslock” per la beckdoor.

```
GNU nano 2.0.7 File: /etc/inetd.conf
#<off># netbios-ssn      stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/in.tftpd
telnet      stream  tcp      nowait  telnetd /usr/sbin/tcpd  /usr/sbin/in.telnetd
#<off># ftp             stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/in.ftp
tftp        dgram   udp      wait    nobody   /usr/sbin/tcpd  /usr/sbin/in.tftpd
shell       stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rsh
login       stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rlogin
#exec       stream  tcp      nowait  root    /usr/sbin/tcpd  /usr/sbin/in.rexecd
#ingreslock stream  tcp      nowait  root    /bin/bash      bash -i
```

5. L'ultima vulnerabilità risolta è VNC Server 'password' Password

CRITICAL VNC Server 'password' Password

Description

The VNC server running on the remote host is secured with a weak password. Nessus was able to login using VNC authentication and a password of 'password'. A remote, unauthenticated attacker could exploit this to take control of the system.

Solution

Secure the VNC service with a strong password.

Output

Nessus logged in using a password of "password".

To see debug logs, please visit Individual host

Port ▲	Hosts
5900 / tcp / vnc	192.168.50.101

Questo servizio permette di controllare un altro dispositivo da remoto per visualizzarne il contenuto. In questo caso la password per accedere a tale servizio sulla nostra macchina è troppo debole; infatti, Nessus ci avvisa dicendoci che è riuscito ad accedere usando la password: “password”.

Dovremo quindi semplicemente cambiare la password, ne potremo inserire una con massimo 8 caratteri di lunghezza, se proveremo a inserirne una più lunga ci verrà automaticamente troncata.

```
root@metasploitable:~# vncpasswd
Using password file /root/.vnc/passwd
Password:
Warning: password truncated to the length of 8.
Verify:
Would you like to enter a view-only password (y/n)? n
root@metasploitable:~# _
```

Terminate le remediation actions abbiamo ridotto significativamente le vulnerabilità del sistema. Come possiamo vedere dalla scansione finale siamo scesi a 5 di livello critical rispetto ai 12 della scansione iniziale.

