

PROGETTO 23 FEBBRAIO S5L5

Oggi andremo a fare una scansione completa sul target Metasploitable, tramite Nessus. L'obiettivo è quello di implementare delle azioni di rimedio su delle vulnerabilità trovate dal nostro vulnerability scanner.

Come in Fig.1, possiamo vedere il report delle vulnerabilità più critiche trovate dallo scanner, sono riuscito a risolverne 6.

Vulnerabilities					Total: 109
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*	-	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	6.1	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)	
HIGH	7.5	6.7	90509	Samba Badlock Vulnerability	
HIGH	7.5*	5.9	10205	rlogin Service Detection	
HIGH	7.5*	5.9	10245	rsh Service Detection	
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	

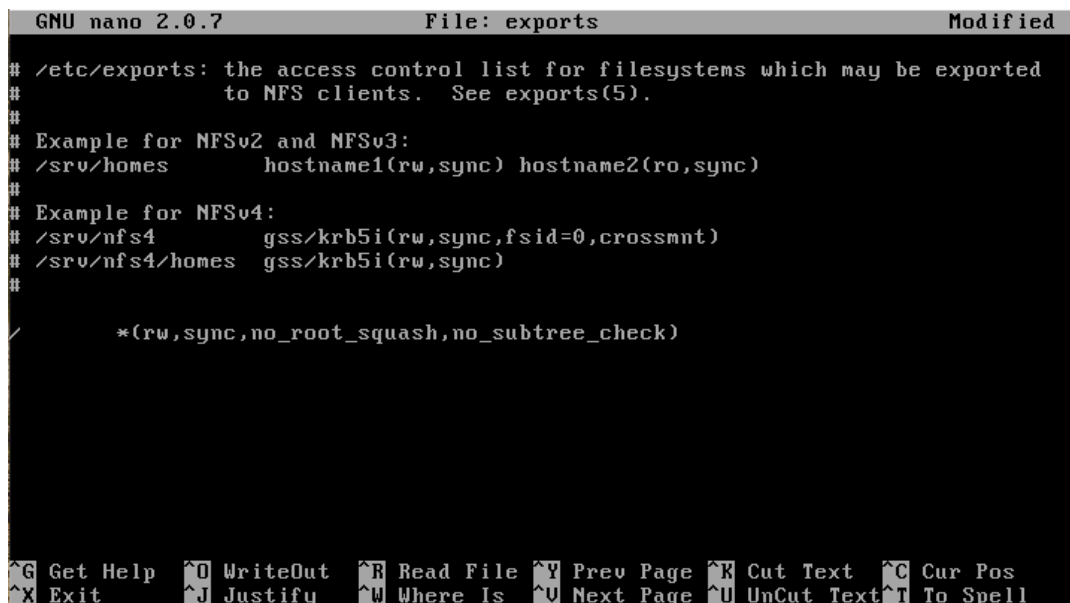
Fig.1

Le prime due vulnerabilità che sono riuscito a risolvere, riguardano il Network File System, o NFS, che è un protocollo di rete che consente a computer client di utilizzare la rete per accedere a directory condivise da server remoti come fossero disponibili in locale.

CRITICAL 10.0* - 11356 NFS Exported Share Information Disclosure

HIGH 7.5 - 42256 NFS Shares World Readable

La vulnerabilità consisteva nel fatto che era possibile accedere alle condivisioni NFS sull'host remoto, quindi un utente malintenzionato poteva essere in grado di sfruttare questa vulnerabilità per leggere ed eventualmente scrivere file sull'host remoto.



```
GNU nano 2.0.7 File: exports Modified
# /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)

^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
```

Fig.2

Come vediamo in Fig.2, all'interno del file exports, troviamo l'ultima riga di comando che specifica il percorso locale della directory, le modalità accesso ed i client autorizzati a collegarsi. La wildcard ci permette di condividere la directory con tutti gli utenti, ed è proprio questo il fulcro della nostra vulnerabilità: togliendo la wildcard ed inserendo, ad esempio, un indirizzo IP qualunque, si restringe l'accesso all'client prescelto. Tramite questa soluzione sono riuscito a risolvere anche la vulnerabilità n. 42256.

Per la terza vulnerabilità che ho risolto, c'era un problema di autenticazione: il server VNC in esecuzione sull'host remoto era protetto da una password molto debole.

CRITICAL 10.0* - 61708 VNC Server 'password' Password

Nessus è riuscito ad accedere al servizio perché protetto dalla password “password”, si trattava quindi di cambiare le credenziali del servizio VNC.

```
msfadmin@metasploitable:/$ sudo su
root@metasploitable:/# vncserver

New 'X' desktop is metasploitable:2

Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/metasploitable:2.log

root@metasploitable:/# vncpasswd
Using password file /root/.vnc/passwd
Password:
Verify:
```

Fig.3

In figura 3 vediamo come è stato possibile, entrando nel vncserver, da Meta, cambiare password. (Ho usato una password di 8 caratteri).

La 4a vulnerabilità era un backdoor in ascolto sulla porta :1524, quindi l'utente malintenzionato poteva controllare l'host da remoto collegandosi alla suddetta porta.

CRITICAL	9.8	-	51988	Bind Shell Backdoor Detection
----------	-----	---	-------	-------------------------------

Come vediamo in Fig.4, tramite terminale Kali, ho usato il comando telnet iptarget porta per vedere se riuscivo a prendere il controllo del target, e così è stato .

```
(kali@kali)-[~]
$ telnet 192.168.49.101 1524
Trying 192.168.49.101 ...
Connected to 192.168.49.101.
Escape character is '^]'.
root@metasploitable:/# whoami
root
root@metasploitable:/# root@metasploitable:/# exit
exit
Connection closed by foreign host.
```

Fig.4

Ho quindi inserito un regola firewall su PFSense che bloccasse l'accesso alla porta e quindi alla backdoor (Fig.5).

<input type="checkbox"/>	<input checked="" type="checkbox"/>	0/0 B	IPv4 TCP	*	*	192.168.49.101	1524	*	none
--------------------------	-------------------------------------	-------	----------	---	---	----------------	------	---	------

Fig.5

In Fig.6 vediamo il firewall in azione, che blocca l'accesso alla backdoor.

```
(kali@kali)-[~]
$ telnet 192.168.49.101 1524
Trying 192.168.49.101 ...
```

Fig.6

Nelle ultime due vulnerabilità che ho risolto, c'era un servizio di rlogin e rsh in esecuzione sull'host remoto.

HIGH	7.5*	5.9	10205	rlogin Service Detection
HIGH	7.5*	5.9	10245	rsh Service Detection

Un utente malintenzionato man-in-the-middle può sfruttare questa situazione per sniffare login e password. Inoltre, potrebbe consentire accessi scarsamente autenticati senza password. Se l'host è vulnerabile all'ipotesi del numero di sequenza TCP (da qualsiasi rete) o allo spoofing IP (incluso il dirottamento ARP su una rete locale), potrebbe essere possibile ignorare l'autenticazione. La soluzione è stata quella di commentare due righe di codice nel file inetd.conf, precisamente andavano commentati la riga di cell e quella di login (Fig.7).

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

Fig.7

In Fig.8, vediamo il report della scansione fatta dopo avere risolto le vulnerabilità sopracitate.

Vulnerabilities					Total: 95
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	-	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	6.1	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)	
HIGH	7.5	6.7	90509	Samba Badlock Vulnerability	
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	6.5	-	42263	Unencrypted Telnet Server	
MEDIUM	5.9	4.4	136808	ISC BIND Denial of Service	
MEDIUM	5.9	3.6	31705	SSL Anonymous Cipher Suites Supported	
MEDIUM	5.9	4.4	89058	SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolete and Weakened eNcryption)	

Fig.8

In conclusione, la scansione di Meta ha rivelato una serie di vulnerabilità critiche che, fortunatamente, sono riuscito ad affrontare con successo grazie alle misure di mitigazione e correzione adottate. Attraverso un processo di analisi approfondito, sono riuscito a ridurre significativamente il rischio di esposizione a potenziali attacchi informatici e a rafforzare la sicurezza complessiva di Metasploitable.