

## ESERCIZIO W12D4

La seconda parte del progetto di fine modulo verte sul risolvimento da 2 fino 4 vulnerabilità trovate con il Vulnerability scanner Nessus. Una volta conclusa la scansione, e esportato un report iniziale, non ci resta apportare modifiche.

### Prima vulnerabilità

#### 11356 - NFS Exported Share Information Disclosure

##### Synopsis

It is possible to access NFS shares on the remote host.

##### 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.

##### Risk Factor

Critical

Secondo lo scanner, il servizio NFS espone gran parte delle directory, rendendole di fatto montabili su sistemi client esterni. Non dovrebbero essere accessibili da tutti, ma solo dall'utenza autorizzata. Per correggere questa vulnerabilità andiamo a cambiare la configurazione del servizio NFS direttamente da dentro Metasploitable. Andremo a creare una cartella privata con mkdir e la renderemo visibile solo da localhost.

```
[ Read 12 lines ]
nsfadmin@metasploitable:/etc$ cd ..
nsfadmin@metasploitable:/$ ls
bin    dev    initrd  lost+found  nohup.out  root  sys  var
boot   etc    initrd.img  media      opt        sbin  tmp  vmlinuz
cdrom  home  lib     mnt        proc       srv   usr
nsfadmin@metasploitable:/$ sudo mkdir privateshare
nsfadmin@metasploitable:/$ ls
bin    dev    initrd  lost+found  nohup.out  proc  srv  usr
boot   etc    initrd.img  media      opt        root  sys  var
cdrom  home  lib     mnt        privateshare  sbin  tmp  vmlinuz
nsfadmin@metasploitable:/$ sudo chmod 777 privateshare
nsfadmin@metasploitable:/$
```

Creazione cartella  
privateshare

Una volta fatto, accediamo al file di testo localizzato in /etc/exports. Usiamo sudo nano per aprire il foglio e specifichiamo la cartella privateshare come usufruibile solo da Metasploitable.

```
# /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)
#
/private share localhost(rw,sync,no_root_squash,no_subtree_check)
```

Configurazione Exports

Salviamo le nuove impostazioni e riavviamo Metasploitable2. Più tardi useremo un'altra volta il Vulnerability Scanner, ma ci accertiamo lo stesso della scongiurata vulnerabilità, andando ad effettuare un tentativo di montaggio sulla macchina attaccante. Creiamo una cartella mount per la condivisione, dentro la directory temp.

```
(kali@kali)~$ sudo mkdir /tmp/mount
(kali@kali)~$ cd /tmp/
(kali@kali)~/tmp$ ls
mount
ssh-WFQlZccCy54
systemd-private-08049c2fcef4095801f31b2af9553b8-colord.service-MpxpTu
systemd-private-08049c2fcef4095801f31b2af9553b8-ha
systemd-private-08049c2fcef4095801f31b2af9553b8-Mo
systemd-private-08049c2fcef4095801f31b2af9553b8-po
(kali@kali)~/tmp$ sudo mount -t nfs 192.168.50.100:/privateshare /tmp/mount -nolock
Created symlink /run/systemd/system/remote-fs.target.wants/rpc-statd.service → /usr/lib/systemd/system/rpc-statd.service.
mount.nfs: access denied by server while mounting 192.168.50.100:/privateshare
(kali@kali)~/tmp$
```

Creazione cartella  
e tentativo di  
montaggio

`sudo mount -t nfs 192.168.50.100:/privateshare /tmp/mount -nolock` è la sintassi che ci consentirà di tentare un montaggio di rete:

- `mount` viene utilizzato per montare un file system
- `-t nfs` indica il tipo di file system da montare, in questo `nfs`
- `192.168.50.100:/privateshare` specifica l'indirizzo IP del server NFS e la directory condivisa da montare.
- `/tmp/mount` specifica il punto di mount locale in cui verrà montata la cartella condivisa.
- `-nolock` specifica che il mount deve essere effettuato senza il lock manager NFS.

Da come si evince dallo screenshot, non è stato possibile completare il montaggio di rete perché il server ha vietato l'accesso alla macchina attaccante.

## Seconda vulnerabilità

### 61708 - VNC Server 'password' Password

#### Synopsis

A VNC server running on the remote host is secured with a weak 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.

La seconda vulnerabilità consiste in una password non adeguata per il server VNC(Virtual network computing), un servizio che consente

l'accesso e il controllo remoto di un computer. Per risolverla ci basterà semplicemente configurare una password più efficace. Utilizziamo il comando `vncpasswd` sotto le vesti root.

```
msfadmin@metasploitable:~$ sudo su
root@metasploitable:/home/msfadmin# whoami
root
root@metasploitable:/home/msfadmin# vncpasswd
Using password file /root/.vnc/passwd
Password:
Verify:
Would you like to enter a view-only password (y/n)? y
Password:
Verify:
root@metasploitable:/home/msfadmin# exit
exit
```

cambio password

Una volta fatto, testiamo che la vulnerabilità sia stata corretta. Kali linux ha già installato nel Metasploit Framework, `msfconsole`. Prima di accedere alla tipologia di exploit che ci interessa, attiviamo il database postgresql.

```
(kali㉿kali)-[~]
$ service postgresql start

(kali㉿kali)-[~]
$ service postgresql status
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; disabled; preset: disabled)
   Active: active (exited) since Mon 2024-01-29 11:18:12 EST; 9s ago
     Process: 39006 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 39006 (code=exited, status=0/SUCCESS)
       CPU: 3ms

Jan 29 11:18:12 kali systemd[1]: Starting postgresql.service - PostgreSQL RDBMS ...
Jan 29 11:18:12 kali systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.
```

Apertura database

All'interno della sfiziosa schermata di `msfconsole`, cerchiamo il modulo `vnc_client`. Per confermare la nostra scelta scriviamo `use auxiliary/scanner/vnc/vnc_login`.



```
[*] 192.168.50.100:5900 - 192.168.50.100:5900 - Starting VNC login sweep
[!] 192.168.50.100:5900 - No active DB -- Credential data will not be saved!
[-] 192.168.50.100:5900 - 192.168.50.100:5900 - LOGIN FAILED: :password (Incorrect: Authentication failed)
[*] 192.168.50.100:5900 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

autenticazione fallita

Il login non è riuscito. Per avere un'ulteriore conferma, possiamo passare anche per vncviewer per provare a visualizzare il desktop di un computer remoto. In questo caso, la password 'password' non ha effetto.

```
(kali@kali)-[~]
$ vncviewer 192.168.50.100
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
Authentication failure
```

accesso negato

## Terza vulnerabilità

### 51988 - Bind Shell Backdoor Detection

#### Synopsis

The remote host may have been compromised.

Bin Shell  
backdoor  
Vulnerability

#### 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.

#### Risk Factor

Critical

All'interno del di metasploitable sembra esserci una shell che apre una porta vulnerabile verso l'esterno. In base a ciò che ci ha detto Nessus, sappiamo che questo servizio è aperto sulla porta tcp 1524. Il comando fuser ci permetterà di rintracciare il file.

```
msfadmin@metasploitable:~$ sudo fuser 1524/tcp
lsudol password for msfadmin:
1524/tcp: 4538
msfadmin@metasploitable:~$
```

PID servizio

Scopriamo che il PID (process ID) assegnato alla porta corrisponde a 4538. A questo punto adoperiamo il comando `sudo readlink -f <percorso .exe>`:  
-f: segue il collegamento simbolico. Tutti i componenti, tranne l'ultimo, devono esistere.

```
msfadmin@metasploitable:~$ sudo readlink -f /proc/4538/exe
/usr/sbin/xinetd
msfadmin@metasploitable:~$
```

percorso trovato

Abbiamo dunque il percorso che ci conduce al programma stesso. Non ci resta che eliminarlo.

```
msfadmin@metasploitable:/usr/sbin$ file xinetd
xinetd: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.8, dynamically linked (uses shared libs), stripped
```

```
msfadmin@metasploitable:/usr/sbin$ sudo rm xinetd
```

Riavviamo Metasploitable e controlliamo se la porta tcp 1524 è ancora in funzione. Dalla scansione nmap stealth non visualizziamo più la backdoor.

```
(kali㉿kali)-[~]  
$ sudo nmap -sS 192.168.50.100  
[sudo] password for kali:  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-27 18:30 EST  
Nmap scan report for 192.168.50.100  
Host is up (0.0012s latency).  
Not shown: 983 closed tcp ports (reset)  
PORT      STATE SERVICE  
22/tcp    open  ssh  
25/tcp    open  smtp  
53/tcp    open  domain  
80/tcp    open  http  
111/tcp   open  rpcbind  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
1099/tcp  open  rmiregistry  
2049/tcp  open  nfs  
2121/tcp  open  ccproxy-ftp  
3306/tcp  open  mysql  
5432/tcp  open  postgresql  
5900/tcp  open  vnc  
6000/tcp  open  X11  
6667/tcp  open  irc  
8009/tcp  open  ajp13  
8180/tcp  open  unknown  
  
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Porta  
scomparsa