

Week 5 - Esercizio Settimanale

Risoluzione delle Vulnerabilità

Consegna

Traccia:

Effettuare una scansione completa sul target Metasploitable.

Scegliete da un minimo di 2 fino ad un massimo di 4 vulnerabilità **critiche / high** 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.

Ai fini della soluzione, abbiamo scelto le vulnerabilità in giallo nella figura in slide 3.

Procedimento

Le vulnerabilità da risolvere sono le seguenti:

- **NFS Exported Share Information Disclosure**
- **rexecd Service Detection**
- **Bind Shell Backdoor Detection**
- **VNC Server 'password' Password**

Andrò ad eseguire uno scan prima della risoluzione e uno dopo di essa

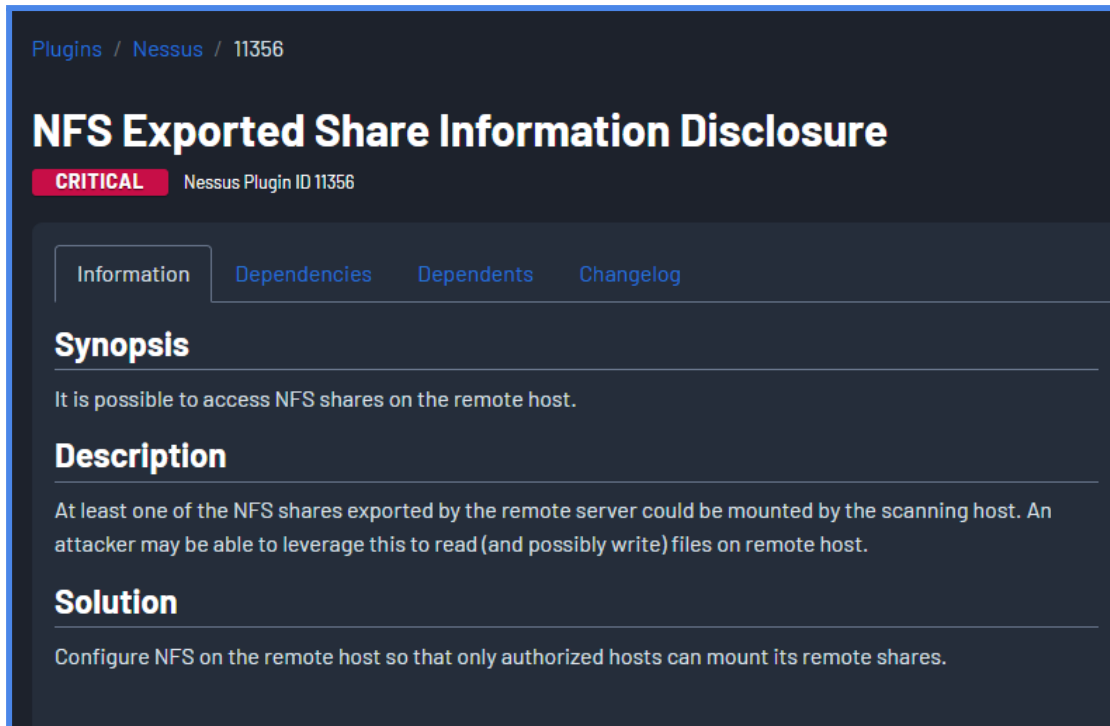
Inizialmente ho eseguito dunque un Basic Scan della macchina Metasploitable attraverso Nessus al fine di analizzarne le vulnerabilità e quindi controllare che fossero presenti quelle richieste dalla consegna.

Di seguito il risultato dello scan prima delle risoluzione delle vulnerabilità richieste:

Vulnerabilities				Total: 105
SEVERITY	CVSS V3.0	PLUGIN	NAME	
CRITICAL	9.8	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	33850	Unix Operating System Unsupported Version Detection	
CRITICAL	10.0*	32314	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness	
CRITICAL	10.0*	32321	Debian OpenSSH/OpenSSL Package Random Number Generator Weakness (SSL check)	
CRITICAL	10.0*	11356	NFS Exported Share Information Disclosure	
CRITICAL	10.0*	61708	VNC Server 'password' Password	

Da come notiamo nel report preliminare, tra le vulnerabilità critiche Metasploitable non presenta **rexecd Service Detection**

Vulnerabilità 1 >>> NFS Exported Share Information Disclosure



Plugins / Nessus / 11356

NFS Exported Share Information Disclosure

CRITICAL Nessus Plugin ID 11356

Information Dependencies Dependents Changelog

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.

NFS è un protocollo di Rete che permette di gestire come vengono salvati o recuperati i file all'interno dei dispositivi di archiviazione in una rete.

La vulnerabilità consiste nel fatto che un potenziale Attaccante potrebbe sfruttare la possibilità

Nessus ci consiglia come soluzione ad essa di Configurare il servizio NFS in modo che solo gli Host autorizzati possano accedere al Fyle System condiviso.

Per risolvere questa vulnerabilità ho cercato su Internet in base alla soluzione consigliatami da Nessus e ho trovato un articolo di IBM Cloud che mi ha portato sulla strada corretta:

<https://exchange.xforce.ibmcloud.com/vulnerabilities/79>

L'articolo proponeva di andare a ricercare nella directory **etc** e quindi nel file **exports**

Soluzione

Check the configuration of the `/etc/exports` on your host. Export file systems only to hosts that require them. Export only to fully qualified hostnames. Make sure export lists do not exceed 256 characters. Use the `showmount` utility to check that exports are correct. Wherever possible, mount file systems to be exported read only and export file systems read only.

If NFS is not needed, consider disabling it, or verify and set permissions to approved users on exported volumes or shared directories. Where possible, mount file systems to be exported read-only and export file systems read-only.

Unix: Check permissions on exported volumes using the `showmount -e` command. If the exported directories look like the listing that follows, anyone can use `mount /usr -` to possibly replace files and gain access:

Dopo di che ho continuato la ricerca sul file **etc/exports** trovando questa pagina web che mi ha permesso di capire meglio come configurare il file https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/s1-nfs-server-config-exports

The difference being "`server:/home`" and "`server:/`". To make the exports configurations compatible for all version, one needs to export (read only) the root filesystem with an `fsid=0`. The `fsid=0` signals the NFS server that this export is the root.

```
/ *(ro,fsid=0)
/home *(rw,sync,nohide)
```



```
GNU nano 2.0.7      File: 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)
#
/      192.168.50.100(rw,sync,no_root_squash,no_subtree_check)

[ Wrote 13 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
```

In questo modo tutti i file sono accessibili solo dall'host 192.168.50.100 ovvero l'ip della macchina Metasploitable

Vulnerabilità 2 >>> Bind Shell Backdoor Detection

Synopsis

The remote host may have been compromised.

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.

Ci viene segnalato da Nessus che ci potrebbe essere una Shell in ascolto sulla porta e un attaccante potrebbe utilizzarla per inviare comandi.

Per risolvere la cosa migliore in questo caso è sicuramente chiudere la porta su cui abbiamo la Shell in ascolto

Ho trovato questo articolo che mi ha permesso di arrivare più velocemente alla soluzione:

<https://security.stackexchange.com/questions/230459/what-is-bindshell-backdoor>

Ho eseguito dunque un nmap completo sull'host e ho verificato che anche in questo caso la porta sull'host aperta fosse la 1524:

```
(kali@kali)-[~]
$ sudo nmap -sV -p- -T5 192.168.50.100
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-25 10:54 EST
Nmap scan report for 192.168.50.100
Host is up (0.00007s 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       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
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         netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
1009/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  ftp          ProFTPD 1.3.1
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/drb)
33316/tcp open  nlockmgr    1-4 (RPC #100021)
38014/tcp open  java-rmi     GNU Classpath grmiregistry
44907/tcp open  status      1 (RPC #100024)
46283/tcp open  mountd      1-3 (RPC #100005)
MAC Address: 08:00:27:60:F5:95 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, 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 195.80 seconds
```

Dunque per rimediare alla vulnerabilità vado a filtrare la porta 1524 con il firewall integrato di Metasploitable:

```
msfadmin@metasploitable:~$ ufw

Usage: ufw COMMAND

Commands:
  enable           Enables the firewall
  disable          Disables the firewall
  default ARG      set default policy to ALLOW or DENY
  logging ARG      set logging to ON or OFF
  allow|deny RULE  allow or deny RULE
  delete allow|deny RULE delete the allow/deny RULE
  status           show firewall status
  version          display version information

msfadmin@metasploitable:~$ ufw enable
ERROR: You need to be root to run this script
msfadmin@metasploitable:~$ sudo ufw enable
Firewall started and enabled on system startup
msfadmin@metasploitable:~$ ufw deny 1524
ERROR: You need to be root to run this script
msfadmin@metasploitable:~$ sudo ufw deny 1524
Rules updated
msfadmin@metasploitable:~$ _
```

Vado a ricontrollare nuovamente con Nmap e notiamo che adesso l'accesso alla porta 1524 è filtrato:

```
(kali㉿kali)-[~]  
└─$ sudo nmap -sV -p1524 -T5 192.168.50.100  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-25 11:19 EST  
Nmap scan report for 192.168.50.100  
Host is up (0.00031s latency).  
  
PORT      STATE      SERVICE      VERSION  
1524/tcp  filtered  ingreslock  
MAC Address: 08:00:27:60:F5:95 (Oracle VirtualBox virtual NIC)
```