Progetto settimanale

Exploit Java-RMI code execution

1. Descrizione della vulnerabilità

La vulnerabilità del servizio Java-RMI è stata rilevata sulla nostra macchina Metasploitable attraverso una precedente fase di scansione della macchina vittima.

Il servizio Java_RMI code execution è attivo sulla porta 1099 della macchina Metasploitable.

La vulnerabilità è data da una configurazione di default errata che permette ad un potenziale attaccante di iniettare codice arbitrario per ottenere accesso amministrativo non autorizzato alla macchina target.

In questa situazione abbiamo saltato i primi due passaggi di ingaggio da parte dell'azienda e di information gathering, in quanto si tratta di un'esercitazione puramente didattica.

Per quanto riguarda la fase "0", ovvero quella di ingaggio, dove il direttore di un'azienda ci contatta per definire il nostro intervento di penetration testing all'interno della sua azienda. In questa sede si stabiliscono le regole per portare avanti il PT, la durata, gli orari, il costo dell'intervento, le clausole legali e si apprendono altre informazioni riguardanti la situazione aziendali (eventuali e-mail dipendenti se white box, configurazione della rete aziendale, ecc.).

Nella prima fase (information gathering), invece, il Pen Tester va a ricercare le informazioni più importanti e dettagliate sul target. Questa fase è davvero molto importante per la buona riuscita del Penetration Testing.

I tool più utilizzati per questa prima fase sono **Whois Lookup, Maltego e TheHarvester, Network Query Tool (NQT) e Recon-ng**, oltre ai siti OSINT.

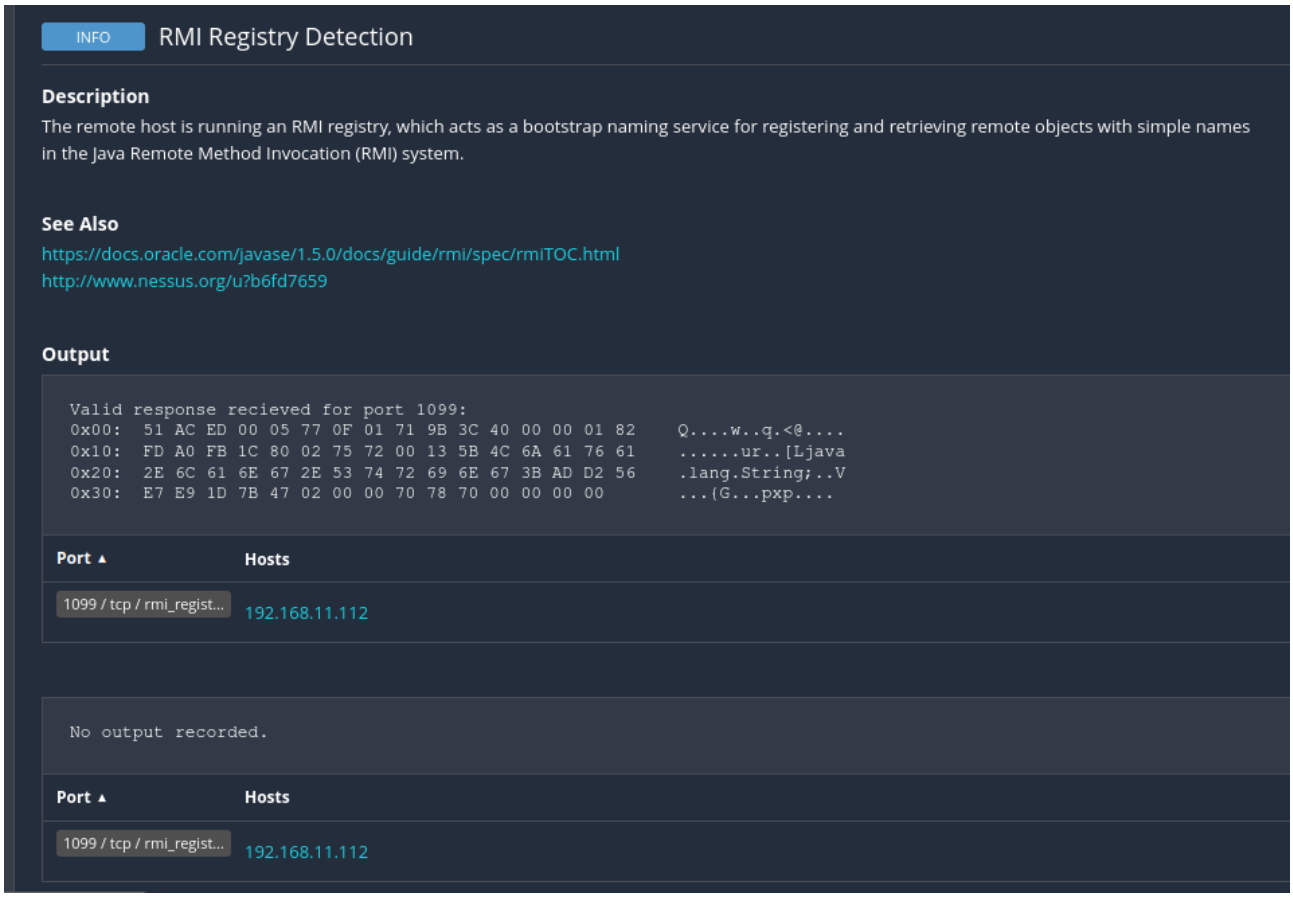
2. Fase di scansione

La fase di scansione è stata effettuata con i tool automatici **Nessus** e **nmap**. Il primo, che è un vulnerability scanner, ci restituisce delle informazioni soggettive sul target, mentre **nmap** ci fornisce informazioni oggettive.

Dalle rilevazioni di Nessus possiamo vedere come la vulnerabilità da exploitare sia riportata con un criticità informational, il che vuol dire che il vulnerability scanner, soggettivamente parlando, non ritiene che questa sia una criticità importante della macchina target.

In realtà, come si vedrà in seguito, questa vulnerabilità dà l'accesso non autorizzato alla macchina vittima, attraverso il framework **Metasploit** e attraverso i moduli adeguati.

Di seguito il risultato ottenuto dalla scansione basic network di Nessus:



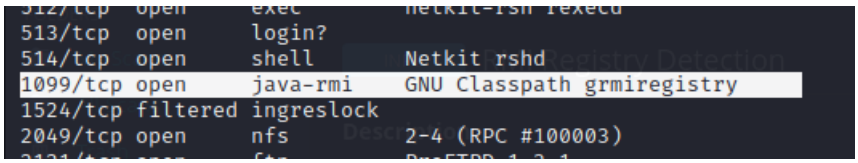
Per avere la certezza e avere un quadro oggettivo della situazione, è opportuno utilizzare anche il tool nmap.

Nmap, come già detto, restituisce la situazione oggettiva della macchina, riportando tutte le informazioni disponibili sulle porte aperte, sul sistema operativo, sulle versioni dei programmi e altro ancora.

In questo caso è stata effettuata una scansione aggressiva della macchina vittima attraverso il comando:

nmap -A 192.168.11.112

Di seguito vediamo come **nmap** abbia rilevato la stessa porta 1099 aperta con il servizio Java_Rmi.



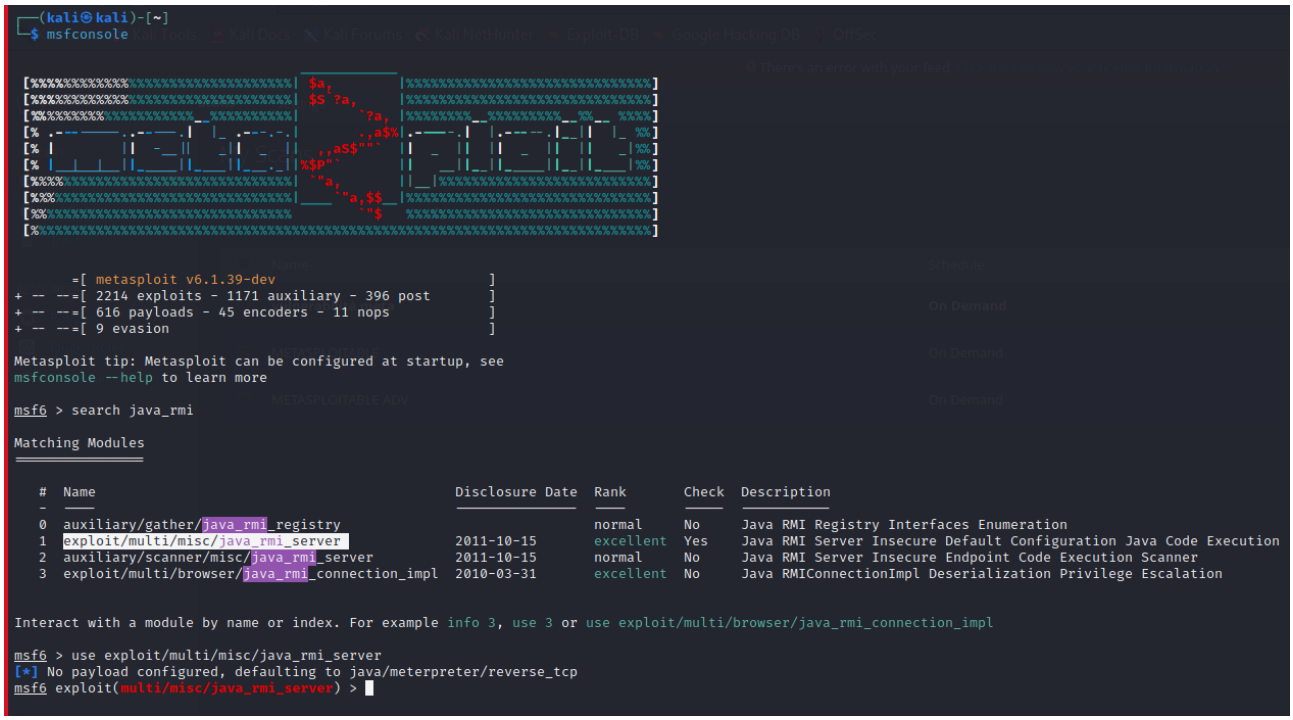
Dopo aver confermato la vulnerabilità del servizio con **nmap,** possiamo procedere con la fase di **exploit**.

3. Fase di exploit

All'interno di questa fase, andremo ad utilizzare lo strumento **Metasploit,** un framework open source utilizzato per il penetration testing e lo sviluppo di exploit.

All'interno di **Metasploit** vi sono circa 2000 exploit e 600 payloads, da utilizzare in base alle esigenze e ai vari target da exploitare. Solitamente, si consiglia di utilizzare gli exploit che sono già stati testati e che riportano la scritta "excellent".

Per exploitare il servizio Java_RMI avviamo innanzitutto il programma MSFConsole da riga di comando su Kali.



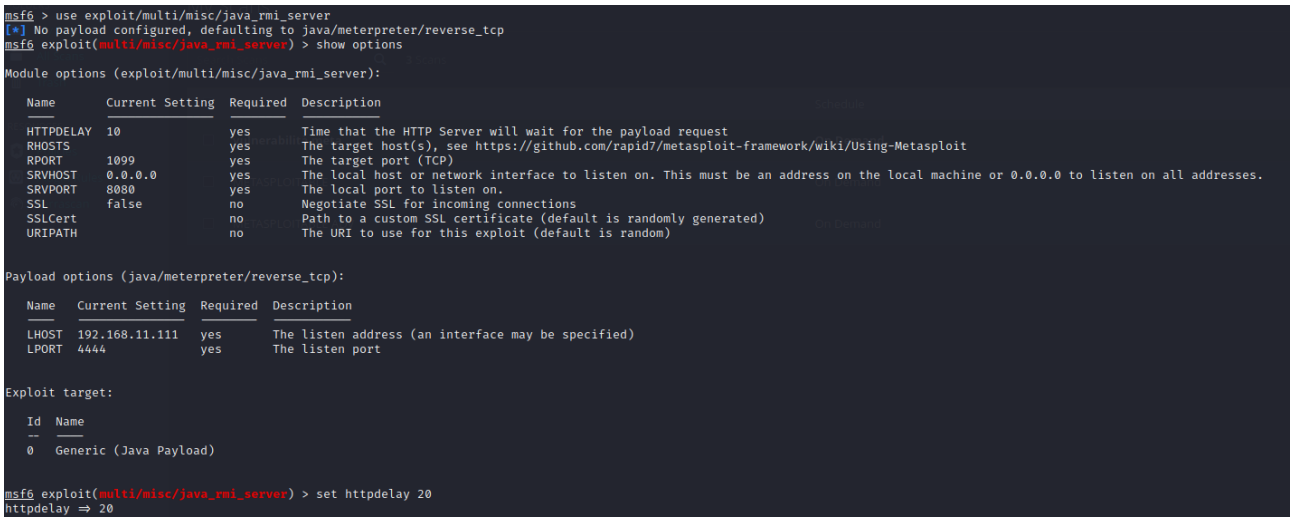
La prima cosa da fare è cercare, attraverso la keyword “search”, quale sia il modulo migliore da utilizzare per riuscire ad ottenere l’accesso alla macchina target.

Attraverso il comando “search Java_RMI” sono stati individuati 4 exploit che potrebbero essere utilizzati:

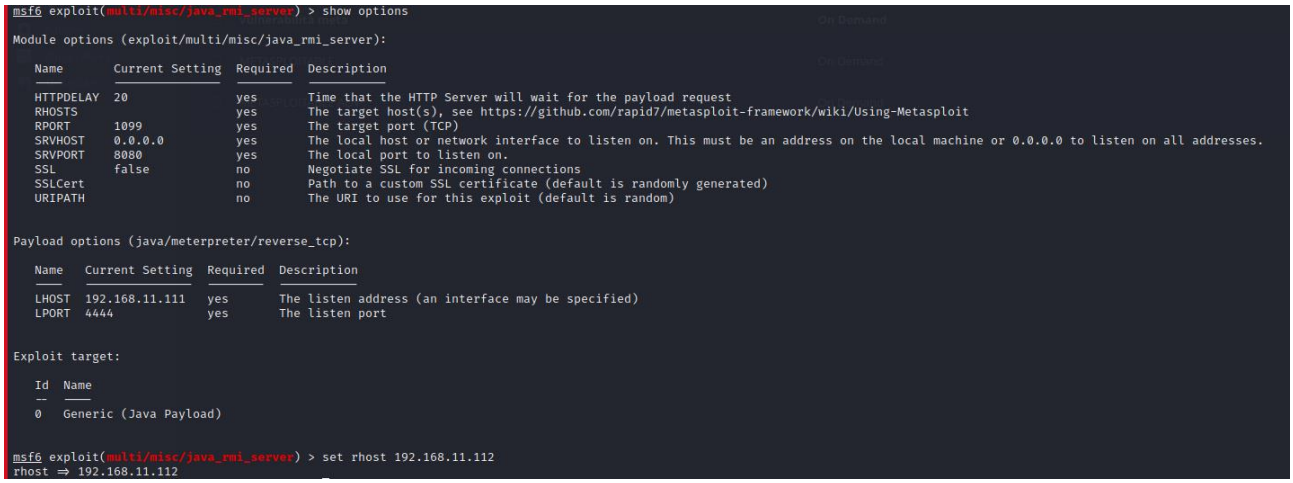
leggendo la descrizione possiamo comprendere che quello più adeguato è il primo in quanto creato appositamente per la configurazione di default errata di Java RMI. In più nella voce “rank” riporta la dicitura “**excellent**”**,** il che significa che è un exploit testato e andato a buon fine.

Attraverso il comando “use” seguito dal path dell’exploit andremo a utilizzare l’exploit.

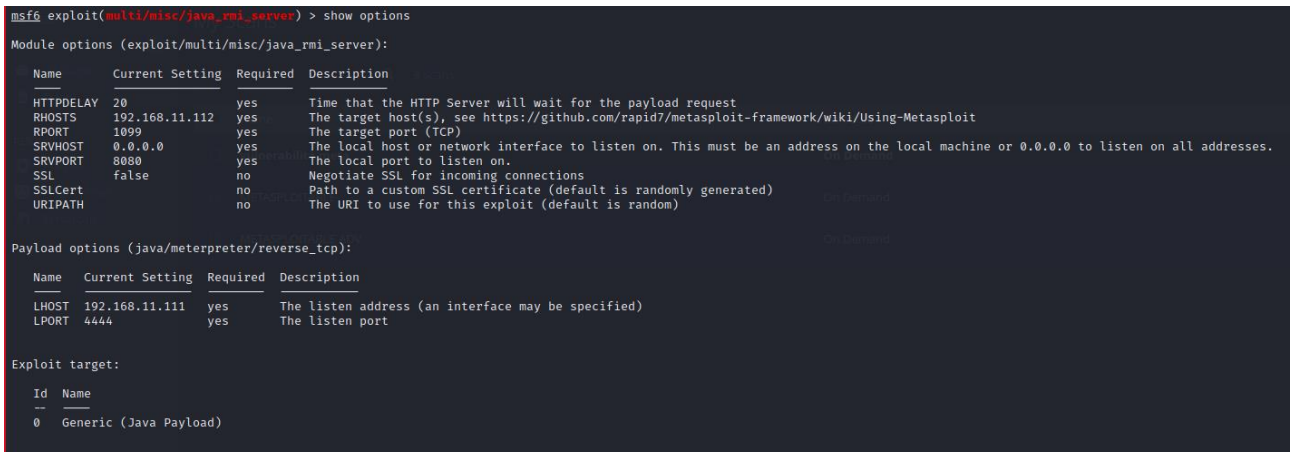
Una volta inserito l’exploit da noi scelto, vediamo che Metasploit ci assegna un payload di default, utile per creare una sessione di Meterpreter, ovvero “java/meterpreter/reverse_tcp”.



Dopodiché, per visualizzare le opzioni già inserite e quelle da configurare, dove vediamo che l’htttp delay è settato a 10. Inoltre, un’altra opzione da configurare è quella dell’RHOST che è obbligatoriamente richiesta per poter utilizzare l’exploit (REQUIRED=YES).

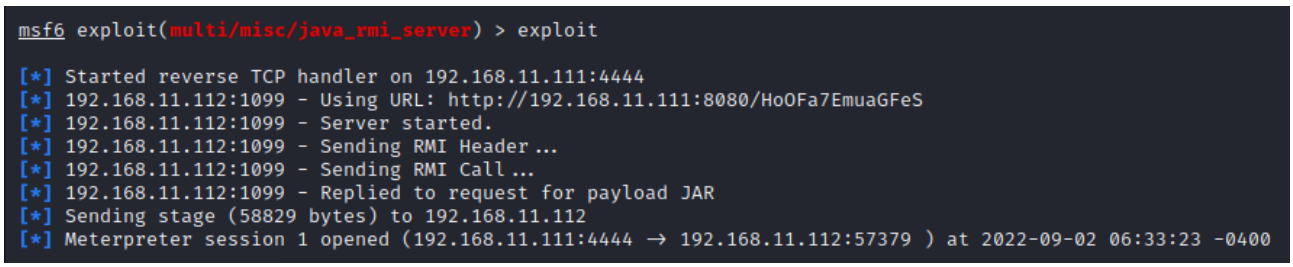


Innanzitutto, l'http delay è stato configurato a 20 attraverso il comando **"set http-delay 20"** e l'rhost (l'ip della macchina target) con indirizzo ip 192.168.11.112, attraverso il comando "**set rhost 192.168.11.112**"



Alla fine, controlliamo nuovamente attraverso il comando "**show options**" se tutto è stato inserito correttamente.

Una volta fatto ciò, possiamo far partire l'exploit e aprire una shell e attraverso il payload di default si aprirà una sessione Meterpreter.



L'exploit è andato a buon fine e la sessione di Meterpreter è stata aperta.

All'interno della sessione di Meterpreter andiamo a utilizzare il comando "**ifconfig**" che ci dà la conferma definitiva che l'accesso amministrativo non autorizzato alla macchina target è riuscito. Questo comando, infatti, ci restituisce l'ip della macchina vittima (192.168.11.112).

Di seguito il comando "ipconfig" eseguito.

```
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:fe95:11b6
IPv6 Netmask : ::

meterpreter >
```

Inoltre, eseguiamo sempre all'interno della sessione di Meterpreter il comando "**sysinfo**" per ricevere le informazioni della macchina target, come ad esempio il nome della macchina, il sistema operativo e la versione, la lingua di sistema e il tipo di sistema (x86, ovvero a 32 bit).

```
meterpreter > sysinfo
Computer        : metasploitable
OS              : Linux 2.6.24-16-server (i386)
Architecture    : x86
System Language : en_US
Meterpreter     : java/linux
meterpreter >
```

Infine, l'ultimo comando eseguito è "**route**" che ci restituisce le impostazioni della tabella di routing della macchina attaccata.

```
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:fe95:11b6  ::       ::
meterpreter >
```