



Al-Najah National University
Intrusion Detection System

Exploit JRMI & Analysis

Instructor: Dr. Othman Othman

Mohammed Adnan
Dima Bshara
Sineen Bazyan

1 Producer

1.1 Capture The Packet

1. This module takes advantage of the default configuration of the RMI Registry and RMI Activation services, which allow loading classes from any remote (HTTP) URL. As it invokes a method in the RMI Distributed Garbage Collector which is available via every RMI endpoint, it can be used against both rmiregistry and RMIID, and most other (custom) RMI endpoints as well.
2. So, we Capture the first packet from the attacker server which does RMIloader using **RMI** protocol on the client-side, then it will send another RMIloader to trigger the garbage collector vulnerability to send another RMIPayload, those two packets using the **HTTP** protocol.
3. Figures 1 and 2 shows those captured packet.

rmi							Expressi
No.	Time	Source	Destination	Protocol	Length	Info	
41	16.905202	10.0.2.4	10.0.2.99	RMI	82	JRMI, ProtocolAck	
66	20.038557	10.0.2.4	10.0.2.99	RMI	88	JRMI, ReturnData	
39	16.905107	10.0.2.99	10.0.2.4	RMI	79	JRMI, Version: 2, StreamProtocol	
45	17.970571	10.0.2.99	10.0.2.4	RMI	242	JRMI, Call	

▶ Frame 45: 242 bytes on wire (1936 bits), 242 bytes captured (1936 bits)						
▶ Ethernet II, Src: PcsCompu_3b:0c:7d (08:00:27:3b:0c:7d), Dst: PcsCompu_f0:bf:b7 (08:00:27:f0:bf:b7)						
▶ Internet Protocol Version 4, Src: 10.0.2.99, Dst: 10.0.2.4						
▶ Transmission Control Protocol, Src Port: 34031, Dst Port: 1099, Seq: 14, Ack: 17, Len: 176						
▶ Java RMI						
0000	08 00 27 f0 bf b7 08 00	27 3b 0c 7d 08 00 45 00	..'. ';;)..E.			
0010	00 e4 1f a2 40 00 40 06	02 0c 0a 00 02 63 0a 00@.@.C..			
0020	02 04 84 ef 04 4b ed 80	f4 f6 83 2a d9 10 80 18K.. ...*....			
0030	00 e5 4c 9d 00 00 01 01	08 0a af d4 c0 f8 ff ff	..L.....			
0040	f2 11 50 ac ed 00 05 77	22 00 00 00 00 00 00 00	..P....w ".....			
0050	02 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00			
0060	00 00 00 f6 b6 89 8d 8b	f2 86 43 75 72 00 18 5bCur..[
0070	4c 6a 61 76 61 2e 72 6d	69 2e 73 65 72 76 65 72	Ljava.rm i.server			
0080	2e 4f 62 6a 49 44 3b 87	13 00 b8 d0 2c 64 7e 02	.objID;.d~.			
0090	00 00 70 78 70 00 00 00	00 77 08 00 00 00 00 00	..pxp... .w.....			
00a0	00 00 00 73 72 00 14 6d	65 74 61 73 70 6c 6f 69	...sr..m etasploi			
00b0	74 2e 52 4d 49 4c 6f 61	64 65 72 a1 65 44 ba 26	t.RMI Loa der.eD.&			
00c0	f9 c2 f4 02 00 00 74 00	24 68 74 74 70 3a 2f 2ft. \$http://			
00d0	31 30 2e 30 2e 32 2e 39	39 3a 38 30 38 30 2f 4b	10.0.2.9 9:8080/K			
00e0	5a 69 6b 79 6d 71 2f 57	4b 2e 6a 61 72 78 70 77	Zikymq/W K.jarxpw			
00f0	01 00		..			

Figure 1: RMILOADER using RMI protocol

http							
No.	Time	Source	Destination	Protocol	Length	Info	
49	17.978827	10.0.2.4	10.0.2.99	HTTP	297	GET	/KZikymq/WK.jar HTTP/1.1
55	17.999653	10.0.2.99	10.0.2.4	HTTP	1209	HTTP/1.1 200 OK	(applicatio

HTTP/1.1 200 OK\r\n							
Content-Type: application/java-archive\r\n							
Connection: Keep-Alive\r\n							
Pragma: no-cache\r\n							
Server: Apache\r\n							

03d0	50 4b 01 02 14 00 14 00	00 00 08 00 aa 71 81 53	PK.....q.S
03e0	7e 9a 15 46 62 00 00 00	91 00 00 00 14 00 00 00	~...Fb....
03f0	00 00 00 00 00 00 00 00	00 00 dc 12 00 00 4d 45ME
0400	54 41 2d 49 4e 46 2f 4d	41 4e 49 46 45 53 54 2e	TA-INF/M ANIFEST.
0410	4d 46 50 4b 01 02 14 00	14 00 00 00 08 00 aa 71	MFPK.....q
0420	81 53 f4 23 8d 06 a0 03	00 00 aa 06 00 00 1a 00	.S.#.....
0430	00 00 00 00 00 00 00 00	00 00 00 00 70 13 00 00p...
0440	6d 65 74 61 73 70 6c 6f	69 74 2f 52 4d 49 4c 6f	metasploit/RMILO
0450	61 64 65 72 2e 63 6c 61	73 73 50 4b 01 02 14 00	ader.classPK...
0460	14 00 00 00 08 00 aa 71	81 53 c0 9e 31 15 30 01q .S..l.0.
0470	00 00 f2 01 00 00 1b 00	00 00 00 00 00 00 00 00
0480	00 00 00 00 48 17 00 00	6d 65 74 61 73 70 6c 6f	...H... metasploit/RMIPa yload.cl
0490	69 74 2f 52 4d 49 50 61	79 6c 6f 61 64 2e 63 6c	assPK.....
04a0	61 73 73 50 4b 05 06 00	00 00 07 00 07 00 c5
04b0	01 00 00 b1 18 00 00 00	00

Figure 2: RMILOADER & RMIPAYLOADER using RMI protocol

1.2 Write & test Rule

1. We write the rule-based in our captured, and we do that if the first packet gets loader the second rule will activate dynamically as shown in figure 3.

```
# additions here.

activate tcp any any -> any any (msg:"alert!!"; content:"metasploit";content:".jar";nocase;sid:10000001;rev:1;activates:1;)

activate tcp any any -> any any (activates:1;msg:"alert!! http is loder class";content:"RMIloader";content:"RMIPayload";nocase;sid:10000002;)
```

Figure 3: Writing the rule

2. Then, we test the rule as shown in figure 5 to make sure that our rule is working fine.

```
test@test-VirtualBox:/etc/snort/rules$ sudo snort -c local.rules -q -A cmg -r ~/rmi.pcap
12/01-21:13:11.320647  [**] [1:10000001:1] alert!! [**] [Priority: 0] {TCP} 10.0.2.99:34031 -> 10.0.2.4:1099
12/01-21:13:11.320647  08:00:27:3B:0C:7D -> 08:00:27:F0:BF:B7 type:0x800 len:0xF2
10.0.2.99:34031 -> 10.0.2.4:1099 TCP TTL:64 TOS:0x0 ID:8098 IpLen:20 DgmLen:228 DF
***AP*** Seq: 0xED80F4F6 Ack: 0x832AD910 Win: 0xE5 TcpLen: 32
TCP Options (3) => NOP NOP TS: 2949955832 4294963729
50 AC ED 00 05 77 22 00 00 00 00 00 00 00 00 02 00 P....w".....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 F6 B6 89 8D 8B F2 86 43 75 72 00 18 5B 4C 6A .....Cur..[Lj
61 76 61 2E 72 6D 69 2E 73 65 72 76 65 72 2E 4F ava.rmi.server.0
62 6A 49 44 3B 87 13 00 B8 D0 2C 64 7E 02 00 00 bjID;.....,d~...
70 78 70 00 00 00 00 77 08 00 00 00 00 00 00 00 pxp....w.....
00 73 72 00 14 6D 65 74 61 73 70 6C 6F 69 74 2E .sr...metasploit.
52 4D 49 4C 6F 61 64 65 72 A1 65 44 BA 26 F9 C2 RMIloader.eD.&...
F4 02 00 00 74 00 24 68 74 74 70 3A 2F 2F 31 30 ....t.$http://10
2E 30 2E 32 2E 39 39 3A 38 30 38 30 2F 4B 5A 69 .0.2.99:8080/KZi
6B 79 6D 71 2F 57 4B 2E 6A 61 72 78 70 77 01 00 kymq/WK.jarxpw..

=====

12/01-21:13:11.349729  [**] [1:10000002:0] alert!! http is loder class [**] [Priority: 0] {TCP} 10.0.2.99:8080 -> 10.0.2.4:4
12/01-21:13:11.349729  08:00:27:3B:0C:7D -> 08:00:27:F0:BF:B7 type:0x800 len:0x4B9
10.0.2.99:8080 -> 10.0.2.4:43337 TCP TTL:64 TOS:0x0 ID:63027 IpLen:20 DgmLen:1195 DF
***AP*** Seq: 0x1D5B314D Ack: 0x8410DABC Win: 0xEB TcpLen: 32
TCP Options (3) => NOP NOP TS: 2949955862 4294963838
A7 E0 58 56 8F 71 DD 18 33 75 B1 63 DA A2 D0 BB ..XV.q..3u.c....
18 C5 63 CA 93 E7 2A 3D 9F B5 B9 D1 9B DF 0D 45 ..c...*=.....E
F0 9D 9C D8 A2 5C B6 B8 5F B4 3D DF B0 25 9B 63 .....\._.=..%.c
BE 33 B8 C8 E3 4E 47 3E EE D8 95 EE 51 81 9C A8 3 06? 0
```

Figure 4: test the rule

1.3 Insuring that Loader is closed

Resolution The Java RMI class loader exploit is resolved in **Java 7.21**, where the RMI property *java.rmi.server.useCodebaseOnly* defaults to true by default.

This change is also applicable to **JDK 6 Update 45** and **JDK 5 Update 45 releases**.

Upgrade the current Java version used by OpenEdge to the later supported version update. Refer to Articles:

- [How to change the Java version for an OpenEdge installation on Windows](#)
- [How to change JDK/JRE version on UNIX](#)

Figure 5: Resolved

2 References

- 1- <https://knowledgebase.progress.com/articles/Article/How-to-prevent-Java-RMI-class-loader-exploit-with-AdminServer>
- 2- https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb