

MÓDULO: Mod_Security

Actividad de Aprendizaje:

Esta actividad busca que los aprendices se familiaricen con el módulo de seguridad Mod_security y la aplicación de este. Para esto, la clase se dividirá en dos grupos, en una primera parte de la actividad cada grupo se encargará de desplegar la aplicación web <https://dvwa.co.uk/> y de aplicar los correctivos correspondientes, en una segunda parte de la actividad cada equipo pondrá a prueba los correctivos llevados a cabo por el equipo contrario.

ModSecurity es un cortafuegos de aplicaciones web (WAF) que proporciona una capa adicional de seguridad para las aplicaciones web. Es un módulo de código abierto para el servidor HTTP Apache, IIS y Nginx que ayuda a proteger los sitios web que tienen vulnerabilidades web (SQLi, XSS, Path Traversal, ...).

El módulo funciona analizando las peticiones HTTP entrantes y aplicando un conjunto de reglas para identificar y bloquear posibles ataques. Estas reglas pueden personalizarse para satisfacer los requisitos de seguridad específicos de un sitio web o aplicación.

Lista de funciones completas:

- Prevención de ataques a aplicaciones web (compatible con Apache)
- Código abierto.
- Registro y supervisión.
- Soporte SSL/TLS.
- Reglas personalizables.

Métodos de despliegue:

Embebido: despliegue junto con el propio Servidor Web (una sola VM).

Proxy inverso: independiente como nodo situado delante del servidor web.

Instalación de Modsecurity en DVWA.

Instalación.

Ejecutaremos los siguientes comandos:

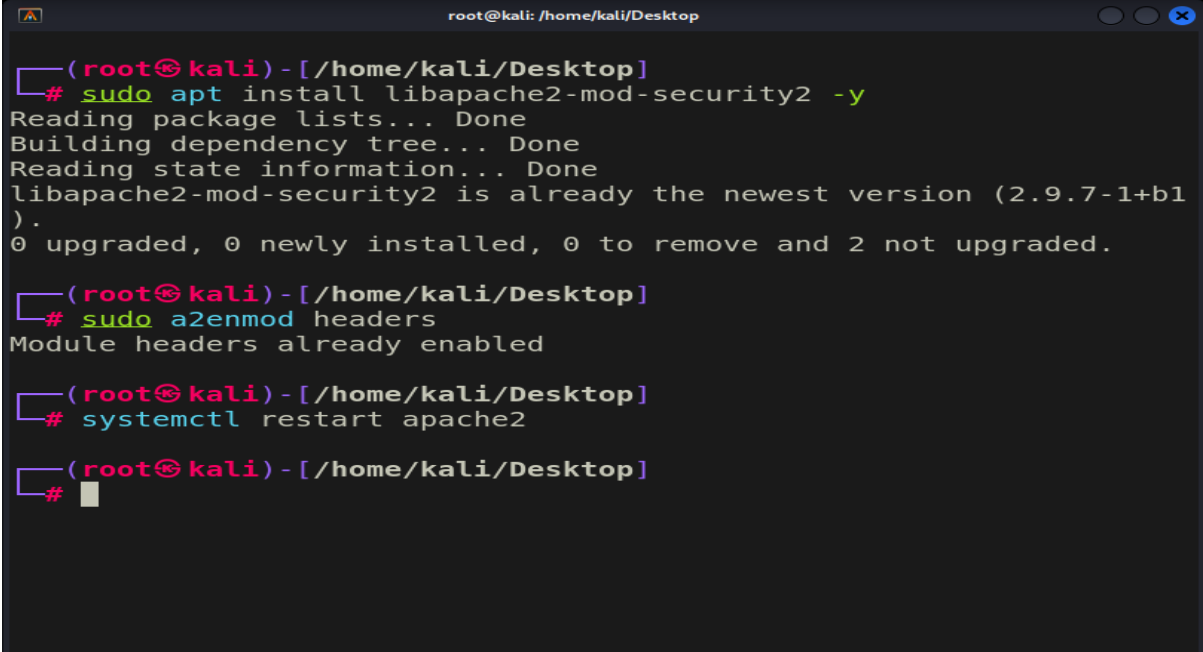
```
sudo apt install libapache2-mod-security2 -y
```

Después de instalar ModSecurity, activamos el módulo de cabeceras de Apache 2 ejecutando el siguiente comando:

```
sudo a2enmod headers
```

Reiniciaremos el servicio de apache2.

```
sudo systemctl restart apache2.
```

A terminal window with a dark background and light-colored text. The window title is 'root@kali: /home/kali/Desktop'. The terminal shows four command prompts and their outputs. The first command is 'sudo apt install libapache2-mod-security2 -y', which outputs package list and dependency tree information, stating that the package is already the newest version. The second command is 'sudo a2enmod headers', which outputs 'Module headers already enabled'. The third command is 'systemctl restart apache2'. The fourth command prompt is shown with a cursor, but no output is visible.

```
(root@kali) - [/home/kali/Desktop]
# sudo apt install libapache2-mod-security2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libapache2-mod-security2 is already the newest version (2.9.7-1+b1).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.

(root@kali) - [/home/kali/Desktop]
# sudo a2enmod headers
Module headers already enabled

(root@kali) - [/home/kali/Desktop]
# systemctl restart apache2

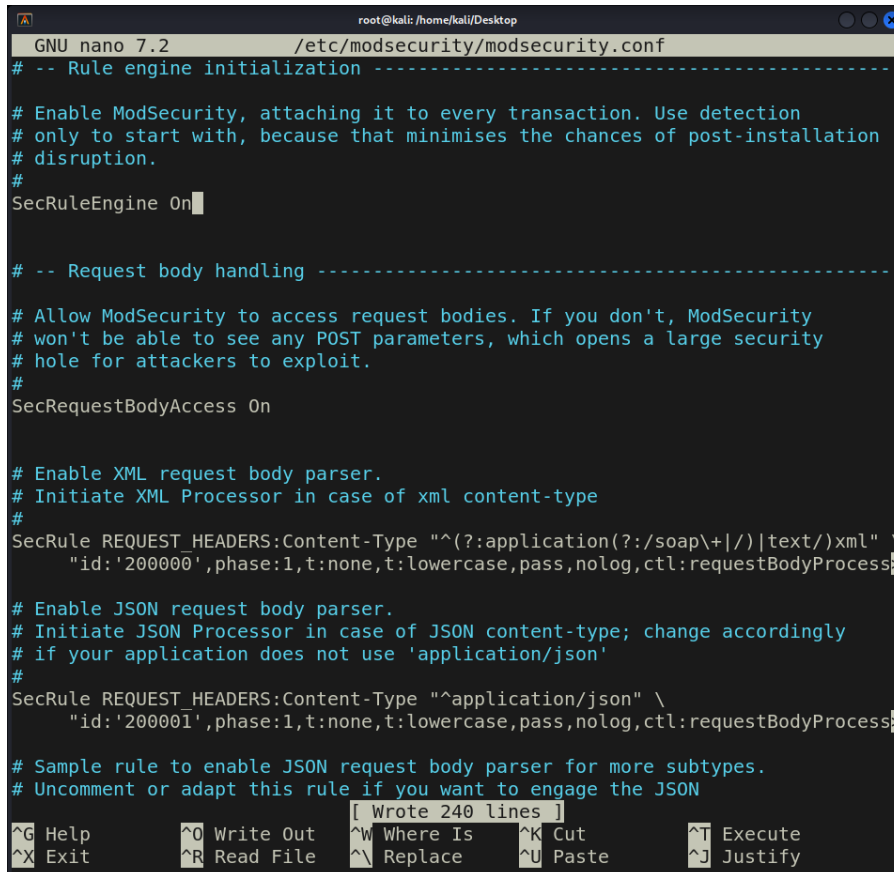
(root@kali) - [/home/kali/Desktop]
# █
```

Configuración de modsecurity.

Utilizamos el archivo de configuración por defecto.

```
sudo cp /etc/modsecurity/modsecurity.conf-recommended  
/etc/modsecurity/modsecurity.conf
```

Editamos `/etc/modsecurity/modsecurity.conf`, cambiando el valor de `SecRuleEngine` a `On`. Luego aplicamos reiniciando el servicio de `apache2`:



The screenshot shows a terminal window with the title `root@kali: /home/kali/Desktop`. The editor is GNU nano 7.2, editing `/etc/modsecurity/modsecurity.conf`. The visible configuration includes:

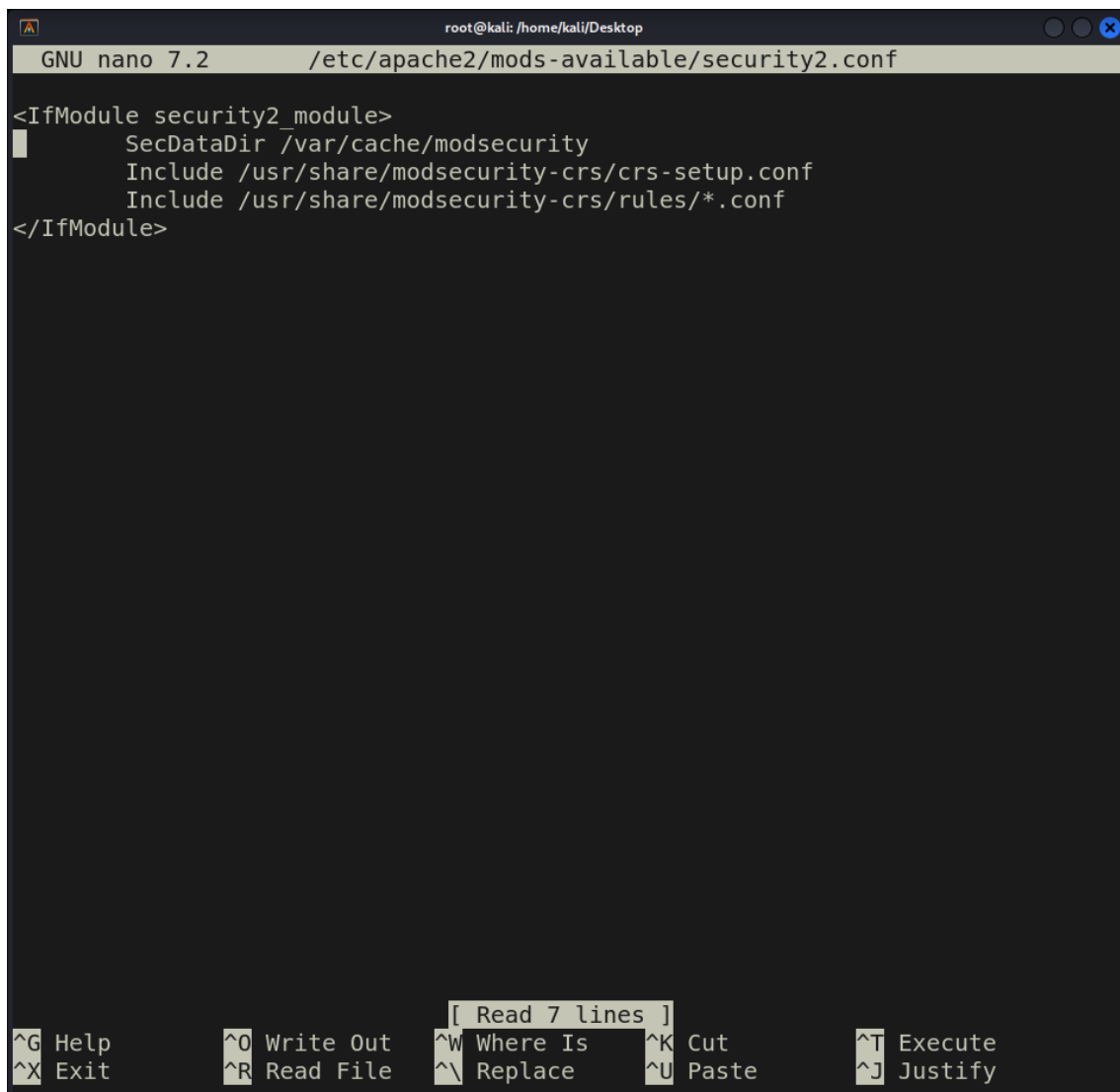
```
# -- Rule engine initialization -----  
  
# Enable ModSecurity, attaching it to every transaction. Use detection  
# only to start with, because that minimises the chances of post-installation  
# disruption.  
#  
SecRuleEngine On  
  
# -- Request body handling -----  
  
# Allow ModSecurity to access request bodies. If you don't, ModSecurity  
# won't be able to see any POST parameters, which opens a large security  
# hole for attackers to exploit.  
#  
SecRequestBodyAccess On  
  
# Enable XML request body parser.  
# Initiate XML Processor in case of xml content-type  
#  
SecRule REQUEST_HEADERS:Content-Type "^(?:application(?:/soap\+|/)|text/)xml" \  
  "id:'200000',phase:1,t:none,t:lowercase,pass,nolog,ctl:requestBodyProcess>  
  
# Enable JSON request body parser.  
# Initiate JSON Processor in case of JSON content-type; change accordingly  
# if your application does not use 'application/json'  
#  
SecRule REQUEST_HEADERS:Content-Type "application/json" \  
  "id:'200001',phase:1,t:none,t:lowercase,pass,nolog,ctl:requestBodyProcess>  
  
# Sample rule to enable JSON request body parser for more subtypes.  
# Uncomment or adapt this rule if you want to engage the JSON  
[ Wrote 240 lines ]  
  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

```
sudo systemctl restart apache2
```

Habilitar ModSecurity en Apache 2

Editamos el fichero de apache2 *sudo nano /etc/apache2/mods-available/security2.conf* para incluirlo en el fichero de configuración:

```
<IfModule security2_module>
    SecDataDir /var/cache/modsecurity
    Include /usr/share/modsecurity-crs/crs-setup.conf
    Include /usr/share/modsecurity-crs/rules/*.conf
</IfModule>
```



```
root@kali: /home/kali/Desktop
GNU nano 7.2 /etc/apache2/mods-available/security2.conf

<IfModule security2_module>
|   SecDataDir /var/cache/modsecurity
    Include /usr/share/modsecurity-crs/crs-setup.conf
    Include /usr/share/modsecurity-crs/rules/*.conf
</IfModule>

[ Read 7 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

Después de haber hecho el paso anterior tendremos que editar el archivo *000-default.conf*. Para ellos abrimos el archivo con un *sudo nano /etc/apache2/sites-enabled/000-default.conf*

```
root@kali: /home/kali/Desktop
GNU nano 7.2 /etc/apache2/sites-enabled/000-default.conf
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    ErrorLog /error.log
    CustomLog /access.log combined

    SecRuleEngine On
</VirtualHost>

[ Read 11 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Y añadiremos la directiva SecRuleEngine a On.

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

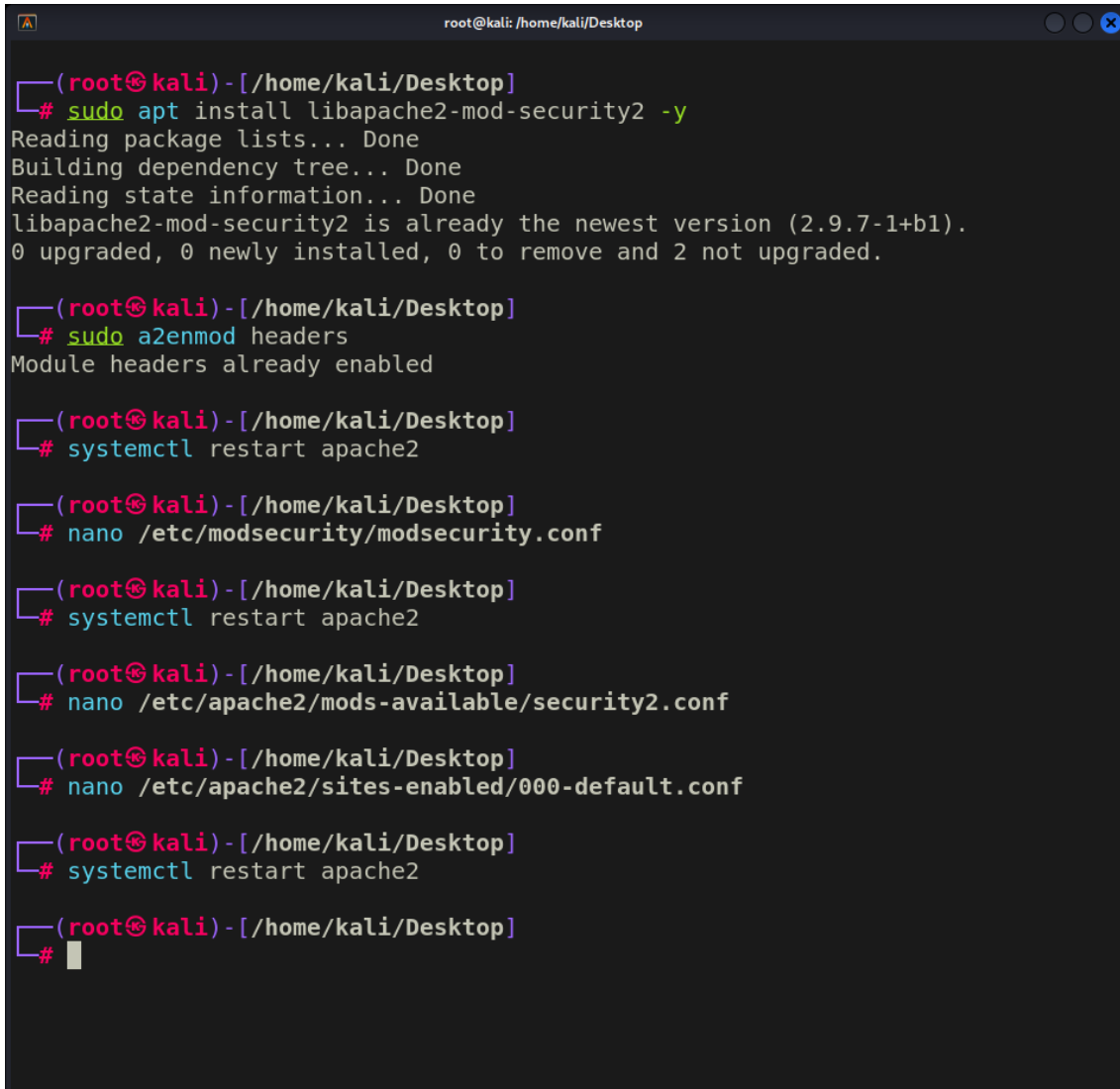
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    SecRuleEngine On
</VirtualHost>
```

Reiniciamos el servicio otra vez.

```
sudo systemctl restart apache2
```

Todos los pasos que hemos hecho resumidamente.

A terminal window titled 'root@kali: /home/kali/Desktop' showing a series of commands and their outputs. The commands are: 1. 'sudo apt install libapache2-mod-security2 -y' which outputs package list and dependency tree information, stating that libapache2-mod-security2 is already the newest version (2.9.7-1+b1) and that 0 packages were upgraded, 0 newly installed, 0 to be removed, and 2 not upgraded. 2. 'sudo a2enmod headers' which outputs 'Module headers already enabled'. 3. 'systemctl restart apache2'. 4. 'nano /etc/modsecurity/modsecurity.conf'. 5. 'systemctl restart apache2'. 6. 'nano /etc/apache2/mods-available/security2.conf'. 7. 'nano /etc/apache2/sites-enabled/000-default.conf'. 8. 'systemctl restart apache2'. The terminal ends with a prompt '#' and a cursor.

```
(root@kali)-[/home/kali/Desktop]
# sudo apt install libapache2-mod-security2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libapache2-mod-security2 is already the newest version (2.9.7-1+b1).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.

(root@kali)-[/home/kali/Desktop]
# sudo a2enmod headers
Module headers already enabled

(root@kali)-[/home/kali/Desktop]
# systemctl restart apache2

(root@kali)-[/home/kali/Desktop]
# nano /etc/modsecurity/modsecurity.conf

(root@kali)-[/home/kali/Desktop]
# systemctl restart apache2

(root@kali)-[/home/kali/Desktop]
# nano /etc/apache2/mods-available/security2.conf

(root@kali)-[/home/kali/Desktop]
# nano /etc/apache2/sites-enabled/000-default.conf

(root@kali)-[/home/kali/Desktop]
# systemctl restart apache2

(root@kali)-[/home/kali/Desktop]
#
```


Con esta configuración modsecurity debería de funcionar plenamente en nuestro DVWA.

Testing.

```
root@kali: ~  
# curl http://127.0.0.1/DVWA/?exec=/bin/bash  
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">  
<html><head>  
<title>403 Forbidden</title>  
</head><body>  
<h1>Forbidden</h1>  
<p>You don't have permission to access this resource.</p>  
<hr>  
<address>Apache/2.4.58 (Debian) Server at 127.0.0.1 Port 80</address>  
</body></html>  
#
```

192.168.1.29/DVWA/vulnerabilities/sql/

Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

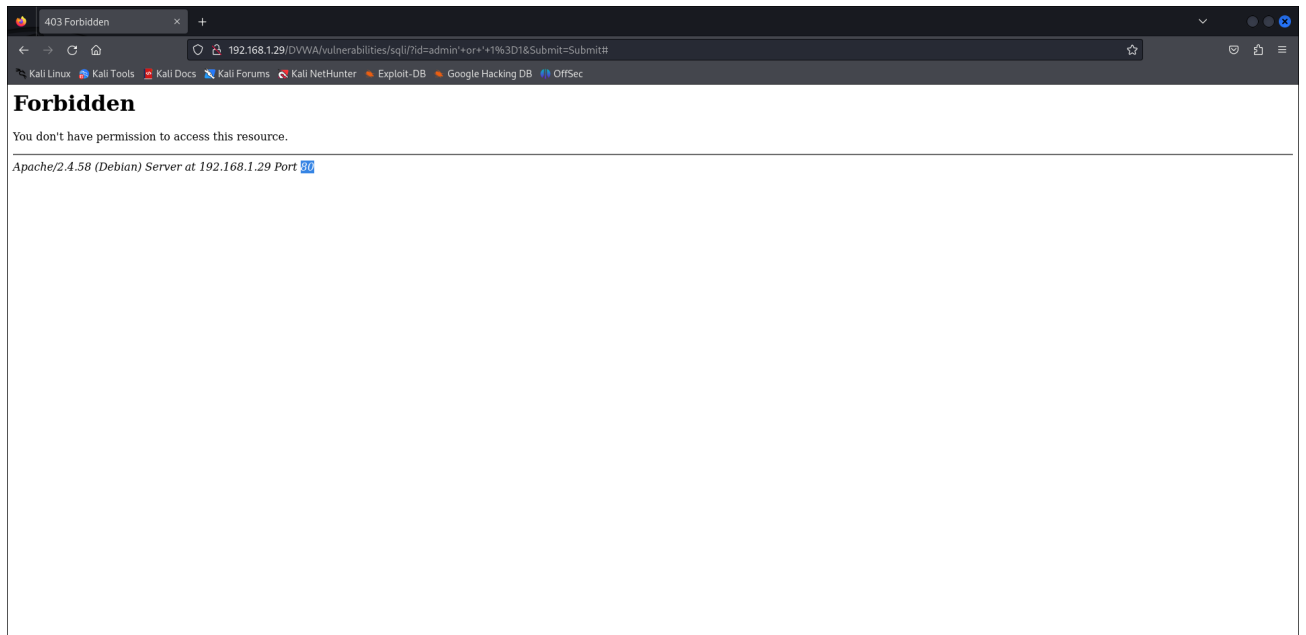
XSS (DOM)

Vulnerability: SQL Injection

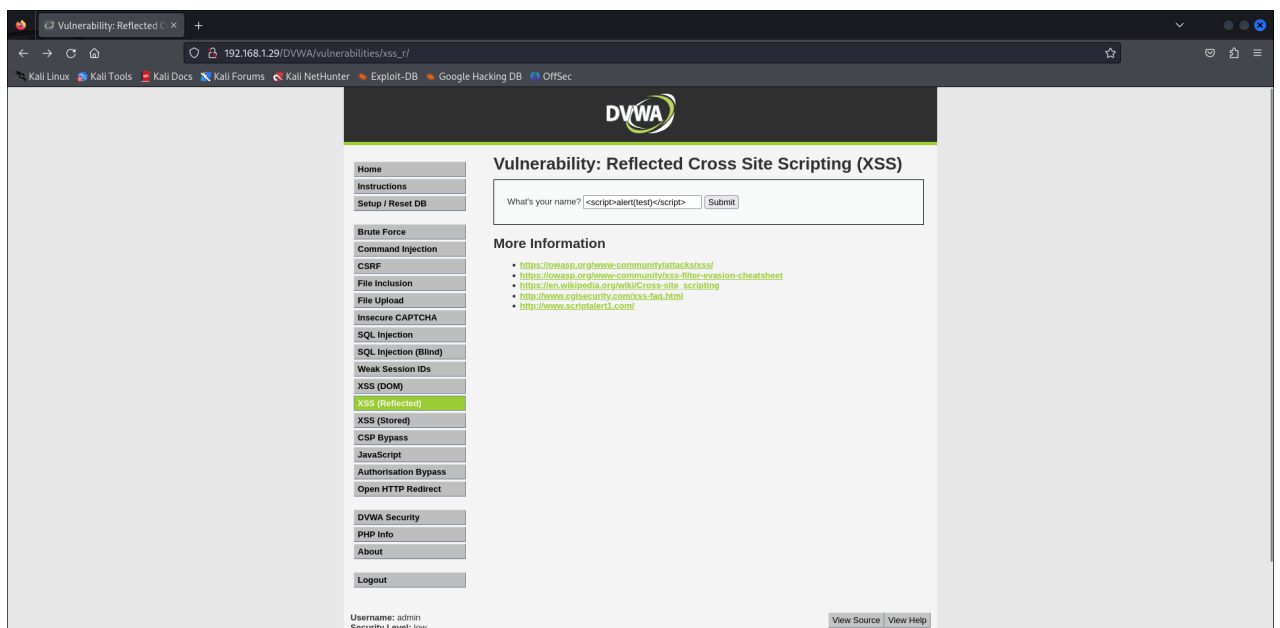
User ID:

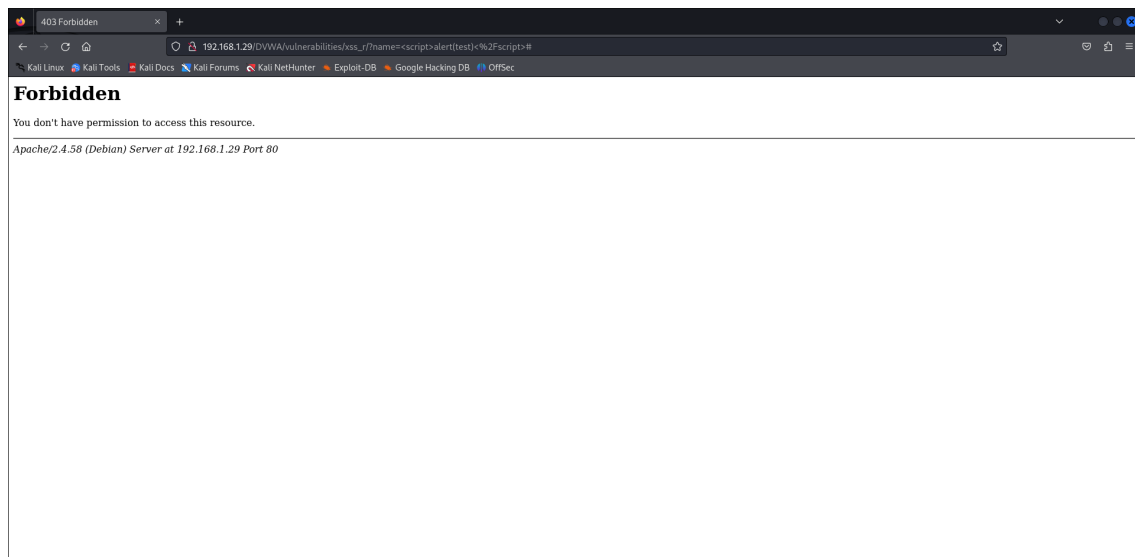
More Information

- https://en.wikipedia.org/wiki/SQL_injection
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- https://owasp.org/www-community/attacks/SQL_injection
- <https://bobby-tables.com/>



Para la fase de testeo he hecho un command injection y un SQLi básico y en las dos me deniega el acceso a los datos.





Aquí ejecuto un XSS y me deniega con un “no tienes permisos a estos recursos.”