

Número i nom del curs	[IFCT0109_CEN] Seguretat dels sistemes d'informació
Mòdul Formatiu que s'avalua	MF0487_3 Auditoria de seguretat informàtica

Nom i Cognoms de l'alumne/a	Julián Gordon
NIF de l'alumne/a	
Data de la prova	11/05/2024
Signatura	

INSTRUCCIONS DE LA PROVA

- En aquest full trobarà la informació necessària per a la realització de la prova.
- Abans de resoldre la pràctica llegeixi amb atenció l'enunciat de l'exercici i comprovi que té tots els materials i equipament necessari.
- Per qualsevol aclariment consulti a l'avaluador/ i/o formador/a.
- Cas de ser necessari, utilitzi els equips de protecció individual necessaris per la realització de la pràctica tenint en compte les normes de seguretat i higiene.

DESCRIPCIÓ DE LA PRÀCTICA	
DENOMINACIÓ	Comprendre i aplicar conceptes de seguretat en aplicacions web per identificar i mitigar vulnerabilitats.
ESPECIFICACIONS TÈCNIQUES	<p>Llegeix l'enunciat de l'exercici i implementa allò que se t'indica.</p> <p>Pots consultar qualsevol font per a la seva realització.</p> <p>Disposes de 4 hores per presentar la prova pràctica, que lliuraràs en un document amb les teves solucions justificades i documentades.</p>
MATERIAL	El teu propi ordinador
TEMPS	Quatre hores
<p>SUPUESTO</p> <p>La empresa IRON S.L. ha desarrollado una nueva aplicación web que utiliza un servidor Apache y una base de datos MariaDB en un entorno Linux.</p> <p>La aplicación cuenta con un formulario de login que es vulnerable a ataques de inyección SQL. Se facilita el fichero php con el código del formulario (loginvuln.php)</p> <p>Además, la empresa quiere asegurarse de que su aplicación web esté protegida contra ataques comunes, por lo que ha decidido implementar ModSecurity con el conjunto de reglas de OWASP.</p>	

Activitat 1 (2 punts)	<p>Instale, inicie y compruebe en un entorno local los servicios Apache y MariaDB.</p> <p>Inicie el servidor MariaDB y cree una base de datos llamada dfir.</p> <p>Cree una tabla usuarios con las columnas userid y password.</p> <p>Inserte al menos tres registros en la tabla usuarios con diferentes combinaciones de usuario y contraseña.</p> <p>Verifique que puede consultar los datos insertados correctamente.</p>
Activitat 2 (2 punts)	<p>Suba el archivo `Vuln.php` a la carpeta indicada del sistema para que sea accesible desde un navegador.</p> <p>Acceda a la página de login vulnerable a través de un navegador.</p> <p>Realice un ataque de inyección SQL para lograr un login exitoso sin conocer las contraseñas.</p>
Activitat 3 (2 punts)	<p>Instale ModSecurity en el servidor Apache.</p> <p>Configure el archivo modsecurity.conf para activar la prevención de ataques y reinicie el servidor Apache y asegúrese de que ModSecurity esté funcionando correctamente.</p>
Activitat 4 (2 punts)	<p>Descargue e instale el conjunto de reglas de OWASP para ModSecurity.</p> <p>Configure Apache para incluir las reglas descargadas y asegúrese de que estén activas revisando los logs de Apache después de intentar un ataque bloqueado.</p> <p>Realice pruebas para verificar que las nuevas reglas bloqueen un intento de inyección SQL similar al de la Actividad 2.</p>

Activitat 5 (2 punts)	<p>Cambia el banner de identificación del servidor Apache para eliminar cualquier tipo de vulnerabilidad.</p> <p>Bloquea una página http y genera una regla que pueda visualizarse en los logs de Apache. Muestre el bloqueo y los resultados.</p>
---------------------------------	--

PUNTUACIÓ FINAL PROVA PRÀCTICA					
ACT 1	ACT 2	ACT 3	ACT 4	ACT 5	<i>PUNTUACIÓ FINAL</i>

OBSERVACIONS

Signatura formador/a	Signatura responsable acció formativa

Actividad 1

Empezaremos esta práctica activando el servidor apache2.

```

root@kali: /home/kali
File Actions Edit View Help
[sudo] password for kali:
(root@kali)-[/home/kali]
# systemctl start apache2

(root@kali)-[/home/kali]
# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)
   Active: active (running) since Tue 2024-05-07 03:28:03 EDT; 33s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 10476 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 10494 (apache2)
       Tasks: 6 (limit: 9441)
    Memory: 19.8M (peak: 20.3M)
         CPU: 119ms
    CGroup: /system.slice/apache2.service
            └─10494 /usr/sbin/apache2 -k start
            └─10499 /usr/sbin/apache2 -k start
            └─10500 /usr/sbin/apache2 -k start
            └─10501 /usr/sbin/apache2 -k start
            └─10502 /usr/sbin/apache2 -k start
            └─10503 /usr/sbin/apache2 -k start

May 07 03:28:03 kali systemd[1]: Starting apache2.service - The Apache HTTP Server...
May 07 03:28:03 kali apachectl[10493]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, please see the /usr/sbin/apachectl (2)
May 07 03:28:03 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-20/20 (END)

```

Una vez tenemos activo apache2, ahora activaremos la base de datos mariadb.

```

File Actions Edit View Help
└─(root@kali)-[/home/kali]
# systemctl start mariadb

└─(root@kali)-[/home/kali]
# systemctl status mariadb
● mariadb.service - MariaDB 10.11.6 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; preset: disabled)
   Active: active (running) since Tue 2024-05-07 03:33:41 EDT; 2s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 13299 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysql (code=exited, status=0/SUC>
   Process: 13301 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SU>
   Process: 13304 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || VAR=cd /usr/bin/..; /usr/bi>
   Process: 13385 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/S>
   Process: 13387 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
 Main PID: 13373 (mariadb)
   Status: "Taking your SQL requests now..."
    Tasks: 13 (limit: 9441)
  Memory: 232.1M (peak: 233.6M)
     CPU: 929ms
   CGroup: /system.slice/mariadb.service
           └─13373 /usr/sbin/mariadb

May 07 03:33:41 kali mariadb[13373]: 2024-05-07 3:33:41 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/>
May 07 03:33:41 kali mariadb[13373]: 2024-05-07 3:33:41 0 [Note] Plugin 'FEEDBACK' is disabled.
May 07 03:33:41 kali mariadb[13373]: 2024-05-07 3:33:41 0 [Warning] You need to use --log-bin to make --expire-logs->
May 07 03:33:41 kali mariadb[13373]: 2024-05-07 3:33:41 0 [Note] InnoDB: Buffer pool(s) load completed at 240507 3:3>
May 07 03:33:41 kali mariadb[13373]: 2024-05-07 3:33:41 0 [Note] Server socket created on IP: '127.0.0.1'.
May 07 03:33:41 kali mariadb[13373]: 2024-05-07 3:33:41 0 [Note] /usr/sbin/mariadb: ready for connections.
May 07 03:33:41 kali mariadb[13373]: Version: '10.11.6-MariaDB-2' socket: '/run/mysql/mysql.sock' port: 3306 Deb>
May 07 03:33:41 kali systemd[1]: Started mariadb.service - MariaDB 10.11.6 database server.

lines 1-26 ... skipping ...
● mariadb.service - MariaDB 10.11.6 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; preset: disabled)
   Active: active (running) since Tue 2024-05-07 03:33:41 EDT; 2s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 13299 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysql (code=exited, status=0/SUCCESS)
   Process: 13301 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 13304 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= || VAR=cd /usr/bin/..; /usr/bin/galera_recovery'; [ $? -eq 0 ] 66 >
   Process: 13385 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)

```

Ahora crearemos un fichero php, con el nombre vuln.php.

```
(root@kali)-[/var/www/html]
# nano vuln.php

(root@kali)-[/var/www/html]
# cat vuln.php
<html>
<head>
<title>
Página de prueba SQLi dfir
</title>
</head>
<body>
<?php
    if(isset($_POST['login']))
    {
        $username = $_POST['username'];
        $password = $_POST['password'];
        $con = mysqli_connect('localhost','root','123456','dfir');
        $result = mysqli_query($con, "SELECT * FROM `usuarios` WHERE userid='$username' AND password='$password'");
        if(mysqli_num_rows($result) == 0)
            echo 'Usuario o Password Incorrecto, prueba otra vez';
        else
            echo '<h1>Dentro!!!</h1><p>Este texto lo ven sólo aquellos que han hecho un login correcto.</p>';
    }
    else
    {
?>
        <form action="" method="post">
            Username: <input type="text" name="username" /><br />
            Password: <input type="password" name="password" /><br />
            <input type="submit" name="login" value="Login" />
        </form>
    <?php
    }
?>
</body>
</html>
```

Mostraremos la base de datos.

```
MariaDB [(none)]> show databases
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.005 sec)
```

Ahora vamos a crear una base de datos que llamaremos dfir.

```
MariaDB [(none)]> create database dfir;  
Query OK, 1 row affected (0.001 sec)  
  
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| dfir      |  
| information_schema |  
| mysql     |  
| performance_schema |  
| sys       |  
+-----+  
5 rows in set (0.001 sec)  
  
MariaDB [(none)]> █
```

Luego que tengamos creada la base de datos, nos conectaremos a ella.

```
(root@kali)-[/var/www/html]  
# mariadb -u root -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 34  
Server version: 10.11.6-MariaDB-2 Debian n/a  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> connect dfir;  
Connection id: 35  
Current database: dfir
```

Y dentro de ella crearemos la tabla usuarios.

```
MariaDB [dfir]> CREATE TABLE usuarios (userid VARCHAR(100), password VARCHAR(100));  
Query OK, 0 rows affected (0.022 sec)
```

Ahora dentro de esta base de datos crearemos 3 usuarios, dfir1, dfir2, dfir3.

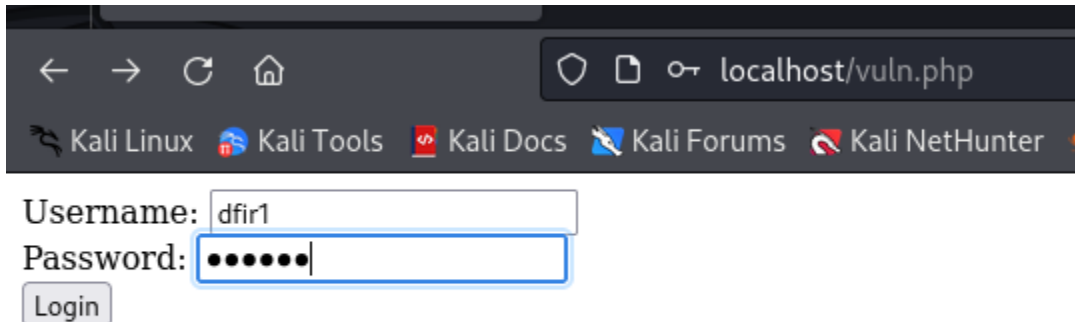
```
MariaDB [dfir]> insert into usuarios values('dfir1','123456');
Query OK, 1 row affected (0.008 sec)

MariaDB [dfir]> insert into usuarios values('dfir2','123456');
Query OK, 1 row affected (0.009 sec)

MariaDB [dfir]> insert into usuarios values('dfir3','123456');
Query OK, 1 row affected (0.008 sec)

MariaDB [dfir]> select * from usuarios;
+-----+-----+
| userid | password |
+-----+-----+
| dfir1  | 123456   |
| dfir2  | 123456   |
| dfir3  | 123456   |
+-----+-----+
3 rows in set (0.001 sec)
```

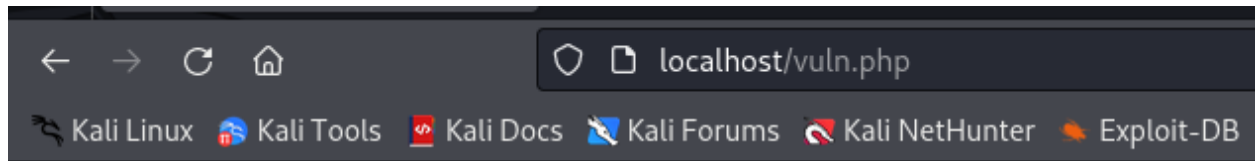
Ahora intentaremos acceder a esta base de datos desde el navegador, con usuario y contraseña 'dfir1' '123456'.



The screenshot shows a web browser window with the address bar displaying 'localhost/vuln.php'. Below the address bar, there are several tabs: 'Kali Linux', 'Kali Tools', 'Kali Docs', 'Kali Forums', and 'Kali NetHunter'. The main content area of the browser shows a login form with the following elements:

- A 'Username:' label followed by a text input field containing 'dfir1'.
- A 'Password:' label followed by a password input field containing seven dots (•••••••).
- A 'Login' button located below the password field.

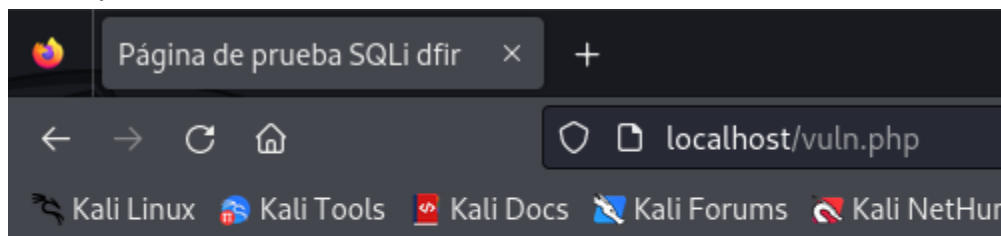
Podemos observar que tuvimos éxito al conectarnos.



Dentro!!!

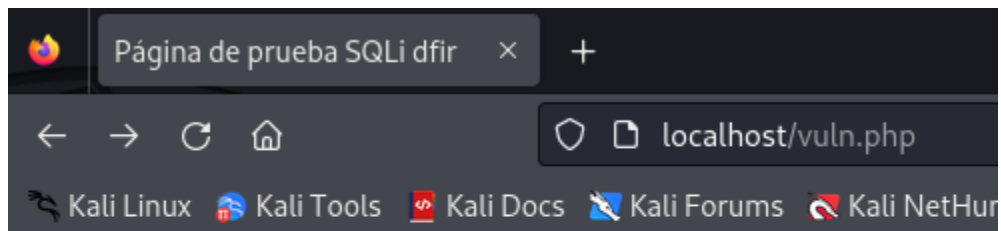
Este texto lo ven sólo aquellos que han hecho un login correcto.

Ahora probaremos con usuario/contraseña incorrecta.



Usuario o Password Incorrecto, prueba otra vez

El siguiente paso es comprobar si es vulnerable a la inyección SQL.

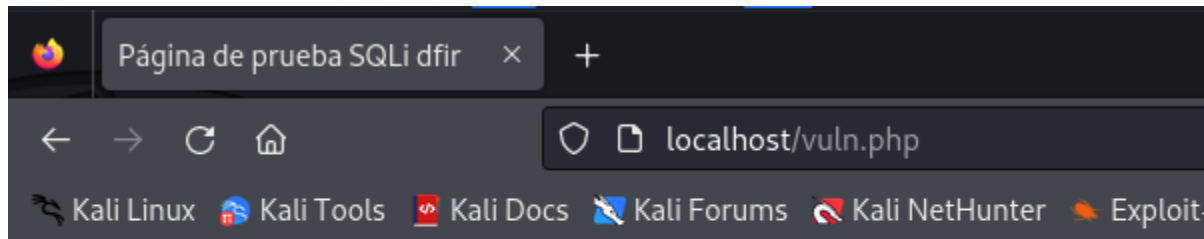


Username:

Password:

Login

Podemos observar que escribiendo la sentencia 'or 1=1#'. Obtuvimos acceso igualmente como si hubiésemos dado las credenciales correctas.



Dentro!!!

Este texto lo ven sólo aquellos que han hecho un login correcto.

Ahora instalaremos modsecurity

```
(root@kali)-[/home/kali/Downloads]
# apt install libapache2-mod-security2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libabsl20220623 libadwaita-1-0 libaio1 libappstream5 libatk-adaptor libboost1.83-dev libopenblas-dev libopenblas-pthread-dev libopenblas0 libpython3-all-dev libpython3.12
  libpython3.12-dev libstemmer0d libxmlb2 libxsimd-dev python3-all-dev python3-anyjson python3-beniget python3-gast python3-pyatspi python3-pydpf2 python3-pyppeteer python3-pyrsistent
  python3-pythrane python3.12-dev xtl-dev zenity zenity-common
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  liblua5.1-0 modsecurity-crs
Suggested packages:
  lua geoip-database-contrib python
The following NEW packages will be installed:
  libapache2-mod-security2 liblua5.1-0 modsecurity-crs
0 upgraded, 3 newly installed, 0 to remove and 4 not upgraded.
Need to get 531 kB of archives.
After this operation, 2,458 kB of additional disk space will be used.
Get:1 http://http.kali.org/kali kali-rolling/main amd64 liblua5.1-0 amd64 5.1.5-9+b1 [108 kB]
Get:2 http://http.kali.org/kali kali-rolling/main amd64 libapache2-mod-security2 amd64 2.9.7-1+b1 [259 kB]
Get:3 http://http.kali.org/kali kali-rolling/main amd64 modsecurity-crs all 3.3.5-2 [163 kB]
Fetched 531 kB in 0s (1,068 kB/s)
Selecting previously unselected package liblua5.1-0:amd64.
```

Modificamos la cabecera http para limitar el acceso.

```
(root@kali)-[/home/kali/Downloads]
# a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
systemctl restart apache2

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

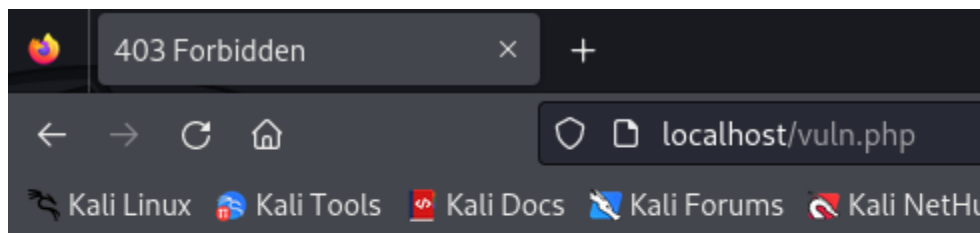
Ahora copiaremos el fichero modsecurity.

```
(root@kali)-[/etc/modsecurity]
# cp modsecurity.conf-recommended modsecurity.conf

(root@kali)-[/etc/modsecurity]
# ls
crs  modsecurity.conf  modsecurity.conf-recommended  unicode.mapping

(root@kali)-[/etc/modsecurity]
#
```

Ahora para probar si funciona, abriremos un navegador y probaremos la configuración para ver si bloquea la Inyección SQL. Vemos que funciona y nos da un error 403.



Forbidden

You don't have permission to access this resource.

Apache/2.4.58 (Debian) Server at localhost Port 80

Ahora borraremos las reglas que vienen por defecto para luego instalar las reglas OWASP recomendadas.

```
(root@kali)-[/usr/share]
# cd modsecurity-crs

(root@kali)-[/usr/share/modsecurity-crs]
# ls
owasp-crs.load  rules  util

(root@kali)-[/usr/share/modsecurity-crs]
# cd ..

(root@kali)-[/usr/share]
# rm -rf modsecurity-crs
```

Para instalar las reglas OWASP, haremos un git clone para clonar el repositorio.

```
(root@kali)-[/usr/share]
# git clone https://github.com/coreruleset/coreruleset /usr/share/modsecurity-crs
Cloning into '/usr/share/modsecurity-crs' ...
remote: Enumerating objects: 30393, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 30393 (delta 0), reused 1 (delta 0), pack-reused 30392
Receiving objects: 100% (30393/30393), 8.03 MiB | 10.11 MiB/s, done.
Resolving deltas: 100% (23755/23755), done.
```

Verificamos que se descargaron correctamente.

```
(root@kali)-[/usr/share]
# cd modsecurity-crs

(root@kali)-[/usr/share/modsecurity-crs]
# ls
CHANGES.md      CONTRIBUTORS.md  docs           KNOWN_BUGS.md  plugins        regex-assembly SECURITY.md      tests
CONTRIBUTING.md crs-setup.conf.example INSTALL.md      LICENSE         README.md      rules          SPONSORS.md    util
```

Ahora vamos a copiar el fichero crs-setup.conf.example creando un fichero crs-setup.conf .

```
(root@kali)-[/usr/share/modsecurity-crs]
# ls
CHANGES.md      CONTRIBUTORS.md  docs           KNOWN_BUGS.md  plugins        regex-assembly SECURITY.md      tests
CONTRIBUTING.md crs-setup.conf.example INSTALL.md      LICENSE         README.md      rules          SPONSORS.md    util

(root@kali)-[/usr/share/modsecurity-crs]
# cp crs-setup.conf.example crs-setup.conf
```

Hacemos lo mismo con el fichero REQUEST-900.

```
(root@kali)-[/usr/share/modsecurity-crs/rules]
# cp REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf

(root@kali)-[/usr/share/modsecurity-crs/rules]
# ls
iis-errors.data          REQUEST-913-SCANNER-DETECTION.conf      RESPONSE-952-DATA-LEAKAGES-JAVA.conf
java-classes.data        REQUEST-920-PROTOCOL-ENFORCEMENT.conf    RESPONSE-953-DATA-LEAKAGES-PHP.conf
java-code-leakages.data  REQUEST-921-PROTOCOL-ATTACK.conf         RESPONSE-954-DATA-LEAKAGES-IIS.conf
java-errors.data         REQUEST-922-MULTIPART-ATTACK.conf        RESPONSE-955-WEB-SHELLS.conf
lfi-os-files.data        REQUEST-930-APPLICATION-ATTACK-LFI.conf  RESPONSE-959-BLOCKING-EVALUATION.conf
php-config-directives.data REQUEST-931-APPLICATION-ATTACK-RFI.conf   RESPONSE-980-CORRELATION.conf
php-errors.data          REQUEST-932-APPLICATION-ATTACK-RCE.conf  RESPONSE-999-EXCLUSION-RULES-AFTER-CRS.conf.example
php-errors-pl2.data      REQUEST-933-APPLICATION-ATTACK-PHP.conf  restricted-files.data
php-function-names-933150.data REQUEST-934-APPLICATION-ATTACK-GENERIC.conf restricted-upload.data
php-function-names-933151.data REQUEST-941-APPLICATION-ATTACK-XSS.conf   scanners-user-agents.data
php-variables.data       REQUEST-942-APPLICATION-ATTACK-SQLI.conf sql-errors.data
REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf REQUEST-943-APPLICATION-ATTACK-SESSION-FIXATION.conf ssrf.data
REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example REQUEST-944-APPLICATION-ATTACK-JAVA.conf  unix-shell.data
REQUEST-901-INITIALIZATION.conf REQUEST-949-BLOCKING-EVALUATION.conf      web-shells-php.data
REQUEST-905-COMMON-EXCEPTIONS.conf RESPONSE-950-DATA-LEAKAGES.conf          windows-powershell-commands.data
REQUEST-911-METHOD-ENFORCEMENT.conf RESPONSE-951-DATA-LEAKAGES-SQL.conf
```

Ahora iremos a la carpeta `apache2` y editamos el fichero `security2.conf`

```
(root@kali)~# cd /etc/apache2
ls
apache2.conf  conf-available  conf-enabled  envvars  magic  mods-available  mods-enabled  ports.conf  sites-available  sites-enabled

(root@kali)~# cd /etc/apache2/mods-available
ls
access_compat.load  authz_dbd.load  data.load  heartbeat.load  md.load  proxy_connect.load  reqtimeout.load  ssl.conf
actions.conf  authz_dbm.load  dav_fs.conf  heartmonitor.load  mime.conf  proxy_express.load  request.load  ssl.load
actions.load  authz_groupfile.load  dav_fs.load  http2.conf  mime_magic.conf  proxy_fcgi.load  rewrite.load  status.conf
alias.conf  authz_host.load  dav_fs.load  http2.load  mime_magic.load  proxy_fdpass.load  security2.conf  status.load
alias.load  authz_owner.load  dav_lock.load  ident.load  mime_magic.load  proxy_ftp.conf  security2.load  substitute.load
allowmethods.load  authz_user.load  dbd.load  imagemap.load  mpm_event.conf  proxy_ftp.load  sed.load  suexec.load
asis.load  autotindex.conf  deflate.conf  include.load  mpm_event.load  proxy_hcheck.load  session_cookie.load  unique_id.load
auth_basic.load  autotindex.load  deflate.conf  info.conf  mpm_prefork.conf  proxy_html.conf  session_crypto.load  userdir.conf
auth_digest.load  brotli.load  dialup.load  info.load  mpm_prefork.load  proxy_html2.load  session_dbd.load  usertrack.load
auth_form.load  buffer.load  dir.conf  lbmethod_bybysnyness.load  mpm_worker.conf  proxy_http2.load  session.load  vhost_alias.load
auth_anon.load  cache_disk.conf  dir.load  lbmethod_byrequests.load  mpm_worker.load  proxy_http.load  setenvif.conf  usertrack.load
authn_core.load  cache_disk.load  dump_io.load  lbmethod_bytraffic.load  negotiation.conf  proxy_load  setenvif.conf  vhost_alias.load
authn_dbd.load  cache_socache.load  echo.load  lbmethod_heartbeat.load  negotiation.load  proxy_load  setenvif.conf  xml2enc.load
authn_dbm.load  cache_socache.load  env.load  ldap.conf  php8.2.conf  proxy_scgi.load  slotmem_plain.load
authn_file.load  cern_meta.load  expires.load  ldap.load  php8.2.load  proxy_uwsgi.load  slotmem_shm.load
authn_socache.load  cgid.conf  ext_filter.load  log_debug.load  proxy_balancer.conf  proxy_uwsgi.load  socache_dbm.load
authnz_fcgi.load  cgid.load  file_cache.load  log_forensic.load  proxy_balancer.load  proxy_wstunnel.load  socache_memcache.load
authnz_ldap.load  cgi.load  filter.load  lua.load  remoteip.load  reflector.load  socache_redis.load
authz_core.load  charset_lite.load  headers.load  macro.load  ratelimit.load  remotepip.load  socache_shmcb.load
  ssl.conf  status.conf  substitute.load  suexec.load  unique_id.load  userdir.conf  usertrack.load  vhost_alias.load  xml2enc.load
```

Debemos modificar la configuración de `security2.conf` para que quede así.

```
File Actions Edit View Help
GNU nano 7.2
<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 /etc/modsecurity/rules/*.conf

# Include OWASP ModSecurity CRS rules if installed
# IncludeOptional /usr/share/modsecurity-crs/*.load

</IfModule>
```

Ahora clonamos otro repositorio de github, coreruleset.

```
(root@kali)-[/etc/apache2/mods-available]
# cd /home/kali/Downloads

(root@kali)-[/home/kali/Downloads]
# git clone https://github.com/coreruleset/coreruleset.git
Cloning into 'coreruleset' ...
remote: Enumerating objects: 30393, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 30393 (delta 0), reused 1 (delta 0), pack-reused 30392
Receiving objects: 100% (30393/30393), 8.00 MiB | 10.86 MiB/s, done.
Resolving deltas: 100% (23751/23751), done.
```

Ahora, al igual que hicimos antes, copiaremos el fichero corerules para editarlo y no tocar el fichero ejemplo.

```
(root@kali)-[/home/kali/Downloads]
# cd coreruleset

(root@kali)-[/home/kali/Downloads/coreruleset]
# ls
CHANGES.md      CONTRIBUTORS.md  docs           KNOWN_BUGS.md  plugins        regex-assembly SECURITY.md      tests
CONTRIBUTING.md crs-setup.conf.example INSTALL.md      LICENSE         README.md      rules          SPONSORS.md    util

(root@kali)-[/home/kali/Downloads/coreruleset]
# cp crs-setup.conf.example /etc/modsecurity/crs-setup.conf
```

Luego debemos copiar el fichero la carpeta rules a la carpeta /etc/modsecurity.

```
(root@kali)-[/home/kali/Downloads/coreruleset]
# cp -av rules /etc/modsecurity
rules' → '/etc/modsecurity/rules'
rules/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example' → '/etc/modsecurity/rules/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example'
rules/REQUEST-901-INITIALIZATION.conf' → '/etc/modsecurity/rules/REQUEST-901-INITIALIZATION.conf'
rules/REQUEST-905-COMMON-EXCEPTIONS.conf' → '/etc/modsecurity/rules/REQUEST-905-COMMON-EXCEPTIONS.conf'
rules/REQUEST-911-METHOD-ENFORCEMENT.conf' → '/etc/modsecurity/rules/REQUEST-911-METHOD-ENFORCEMENT.conf'
rules/REQUEST-913-SCANNER-DETECTION.conf' → '/etc/modsecurity/rules/REQUEST-913-SCANNER-DETECTION.conf'
rules/REQUEST-920-PROTOCOL-ENFORCEMENT.conf' → '/etc/modsecurity/rules/REQUEST-920-PROTOCOL-ENFORCEMENT.conf'
rules/REQUEST-921-PROTOCOL-ATTACK.conf' → '/etc/modsecurity/rules/REQUEST-921-PROTOCOL-ATTACK.conf'
rules/REQUEST-922-MULTIPART-ATTACK.conf' → '/etc/modsecurity/rules/REQUEST-922-MULTIPART-ATTACK.conf'
rules/REQUEST-930-APPLICATION-ATTACK-LFI.conf' → '/etc/modsecurity/rules/REQUEST-930-APPLICATION-ATTACK-LFI.conf'
rules/REQUEST-931-APPLICATION-ATTACK-RFI.conf' → '/etc/modsecurity/rules/REQUEST-931-APPLICATION-ATTACK-RFI.conf'
rules/REQUEST-932-APPLICATION-ATTACK-RCE.conf' → '/etc/modsecurity/rules/REQUEST-932-APPLICATION-ATTACK-RCE.conf'
rules/REQUEST-933-APPLICATION-ATTACK-PHP.conf' → '/etc/modsecurity/rules/REQUEST-933-APPLICATION-ATTACK-PHP.conf'
rules/REQUEST-934-APPLICATION-ATTACK-GENERIC.conf' → '/etc/modsecurity/rules/REQUEST-934-APPLICATION-ATTACK-GENERIC.conf'
rules/REQUEST-941-APPLICATION-ATTACK-XSS.conf' → '/etc/modsecurity/rules/REQUEST-941-APPLICATION-ATTACK-XSS.conf'
rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf' → '/etc/modsecurity/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf'
rules/REQUEST-943-APPLICATION-ATTACK-SESSION-FIXATION.conf' → '/etc/modsecurity/rules/REQUEST-943-APPLICATION-ATTACK-SESSION-FIXATION.conf'
rules/REQUEST-944-APPLICATION-ATTACK-JAVA.conf' → '/etc/modsecurity/rules/REQUEST-944-APPLICATION-ATTACK-JAVA.conf'
rules/REQUEST-949-BLOCKING-EVALUATION.conf' → '/etc/modsecurity/rules/REQUEST-949-BLOCKING-EVALUATION.conf'
rules/RESPONSE-950-DATA-LEAKAGES.conf' → '/etc/modsecurity/rules/RESPONSE-950-DATA-LEAKAGES.conf'
rules/RESPONSE-951-DATA-LEAKAGES-SQL.conf' → '/etc/modsecurity/rules/RESPONSE-951-DATA-LEAKAGES-SQL.conf'
rules/RESPONSE-952-DATA-LEAKAGES-JAVA.conf' → '/etc/modsecurity/rules/RESPONSE-952-DATA-LEAKAGES-JAVA.conf'
rules/RESPONSE-953-DATA-LEAKAGES-PHP.conf' → '/etc/modsecurity/rules/RESPONSE-953-DATA-LEAKAGES-PHP.conf'
rules/RESPONSE-954-DATA-LEAKAGES-IIS.conf' → '/etc/modsecurity/rules/RESPONSE-954-DATA-LEAKAGES-IIS.conf'
rules/RESPONSE-955-WEB-SHELLS.conf' → '/etc/modsecurity/rules/RESPONSE-955-WEB-SHELLS.conf'
rules/RESPONSE-959-BLOCKING-EVALUATION.conf' → '/etc/modsecurity/rules/RESPONSE-959-BLOCKING-EVALUATION.conf'
```

Ahora verificamos si se copió correctamente.

```
(root@kali)-[/home/kali/Downloads/coreruleset]
# cd /etc/modsecurity

(root@kali)-[/etc/modsecurity]
# ls
crs  crs-setup.conf  modsecurity.conf  modsecurity.conf-recommended  rules  unicode.mapping

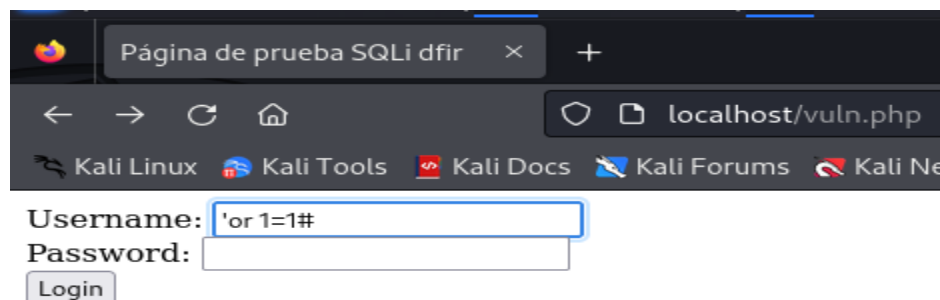
(root@kali)-[/etc/modsecurity]
# cd rules

(root@kali)-[/etc/modsecurity/rules]
# ls
iis-errors.data          REQUEST-920-PROTOCOL-ENFORCEMENT.conf  RESPONSE-953-DATA-LEAKAGES-PHP.conf
java-classes.data        REQUEST-921-PROTOCOL-ATTACK.conf        RESPONSE-954-DATA-LEAKAGES-IIS.conf
java-code-leakages.data  REQUEST-922-MULTIPART-ATTACK.conf        RESPONSE-955-WEB-SHELLS.conf
java-errors.data         REQUEST-930-APPLICATION-ATTACK-LFI.conf  RESPONSE-959-BLOCKING-EVALUATION.conf
java-files.data          REQUEST-931-APPLICATION-ATTACK-RFI.conf  RESPONSE-980-CORRELATION.conf
lfi-os-files.data        REQUEST-932-APPLICATION-ATTACK-RCE.conf  RESPONSE-999-EXCLUSION-RULES-AFTER-CRS.conf.example
php-config-directives.data  REQUEST-933-APPLICATION-ATTACK-PHP.conf  restricted-files.data
php-errors.data          REQUEST-934-APPLICATION-ATTACK-GENERIC.conf  restricted-upload.data
php-errors-pl2.data       REQUEST-941-APPLICATION-ATTACK-XSS.conf  scanners-user-agents.data
php-function-names-933150.data  REQUEST-942-APPLICATION-ATTACK-SQLI.conf  sql-errors.data
php-function-names-933151.data  REQUEST-943-APPLICATION-ATTACK-SESSION-FIXATION.conf  ssrf.data
php-variables.data        REQUEST-944-APPLICATION-ATTACK-JAVA.conf  unix-shell.data
REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf.example  REQUEST-949-BLOCKING-EVALUATION.conf  web-shells-php.data
REQUEST-901-INITIALIZATION.conf  RESPONSE-950-DATA-LEAKAGES.conf        windows-powershell-commands.data
REQUEST-905-COMMON-EXCEPTIONS.conf  RESPONSE-951-DATA-LEAKAGES-SQL.conf
REQUEST-911-METHOD-ENFORCEMENT.conf  RESPONSE-952-DATA-LEAKAGES-JAVA.conf
REQUEST-913-SCANNER-DETECTION.conf
```

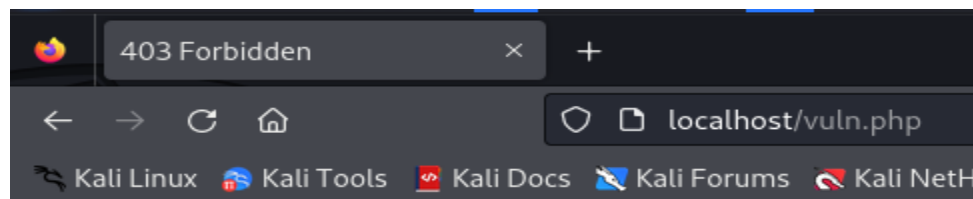
Haremos un restart ahora del servidor apache para ver si todo funciona correctamente.

```
(root@kali)-[/etc/modsecurity]
# systemctl restart apache2
```

Para probar volvemos a abrir el navegador y probamos la inyección 'or 1=1#



Podemos confirmar que funciona correctamente y no nos deja acceder.



Forbidden

You don't have permission to access this resource.

Apache/2.4.58 (Debian) Server at localhost Port 80

Otra forma que tenemos para probar si funciona es probar un RCE (Remote Code Execution).

Para ello usaremos el comando: `curl http://localhost/index.html?exec=/bin/bash` . Que intentaría abrir una shell remota para ejecutar comandos. Podemos verificar que modsecurity ha bloqueado esta vulnerabilidad.

```
(root@kali)-[/etc/modsecurity]
# curl http://localhost/index.html?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 localhost Port 80</address>
</body></html>
```

El siguiente paso que debemos hacer es cambiar el Banner que aparece, ya que al mostrarnos:

Apache/2.4.58 (Debian) Server at localhost Port 80 , está generando una vulnerabilidad, ya que un atacante al saber el tipo de servidor que está corriendo puede encontrar vulnerabilidades relacionadas a esta versión específica.

Para modificar, debemos abrir el fichero security.conf de la carpeta /etc/apache2/conf-available.

```
(root@kali)-[/home/kali]
# cd /etc/apache2

You don't have permission to access this resource.

(root@kali)-[/etc/apache2]
# cd conf-available

(root@kali)-[/etc/apache2/conf-available]
# nano security.conf
```


Comentamos la línea ServerSignature que está en ON y descomentamos la línea que está en OFF.

```
File Actions Edit View Help
GNU nano 7.2
# Changing the following options will not really affect the security of the
# server, but might make attacks slightly more difficult in some cases.

#
# ServerTokens
# This directive configures what you return as the Server HTTP response
# Header. The default is 'Full' which sends information about the OS-Type
# and compiled in modules.
# Set to one of: Full | OS | Minimal | Minor | Major | Prod
# where Full conveys the most information, and Prod the least.
#ServerTokens Minimal
ServerTokens OS
#ServerTokens Full

#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of: On | Off | EMail
ServerSignature Off
#ServerSignature On
#SecServerSignature Windows 403 Forbidden
#
# Allow TRACE method
```

Para comprobar que funciona, volvemos a ejecutar el comando:

curl <http://localhost/index.html?exec=/bin/bash>

Verificamos que ahora la respuesta es distinta y no aparece el nombre ni versión del servidor corriendo.

```
(root@kali)-[/etc/apache2/conf-available]
# curl http://localhost/index.html?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>
</body></html>
```

Ahora crearemos una regla para bloquear una página http y que pueda visualizarse en los logs de Apache. Para ello primero iremos a '/etc/apache2/sites-available'.

```
(root@kali)-[/etc/apache2/conf-available]
# cd /etc/apache2

(root@kali)-[/etc/apache2]
# ls
apache2.conf  conf-available  conf-enabled  envvars  magic  mods-available  mods-enabled  ports.conf  sites-available  sites-enabled

(root@kali)-[/etc/apache2]
# cd sites-available

(root@kali)-[/etc/apache2/sites-available]
# ls
000-default.conf  default-ssl.conf

(root@kali)-[/etc/apache2/sites-available]
# nano 000-default.conf
```

Abriremos el fichero 000-default.conf y lo vamos a editar.

```
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

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

SecRuleEngine On
SecRule ARGS:testparam "@contains test" "id:254,deny,status:403,msg:'Test con éxito'"

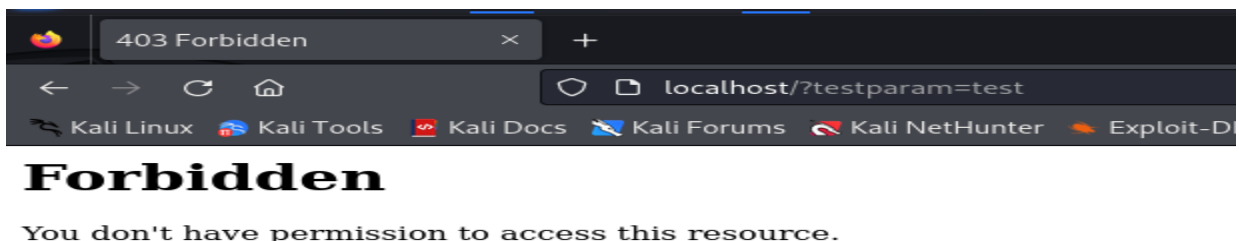
# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Agregaremos las 2 siguientes líneas:

SecRuleEngine On

SecRule ARGS:testparam "@contains test" "id:254,deny,status:403,msg:'Test con éxito'".

Ahora probamos esta regla desde el navegador y verificamos que funciona.



Agregaremos una regla más para estar seguros de que funciona. Editaremos el fichero /etc/apache2/sites-enabled/000-default.conf

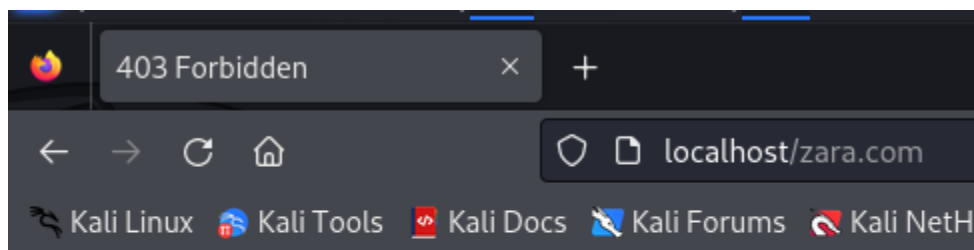
```
(root@kali)-[/var/log/apache2]  
# nano /etc/apache2/sites-enabled/000-default.conf
```

Y agregamos la línea:

SecRule REQUEST_URI "@contains /zara" "id:255,deny,status:403,msg:'Acceso prohibido aqui no se vende ropa'"

```
ErrorLog ${APACHE_LOG_DIR}/error.log  
CustomLog ${APACHE_LOG_DIR}/access.log combined  
#  
SecRuleEngine On  
SecRule ARGS:testparam "@contains test" "id:254,deny,status:403,msg:'Test con exito'"  
SecRule REQUEST_URI "@contains /zara" "id:255,deny,status:403,msg:'Acceso prohibido aqui no se vende ropa'"
```

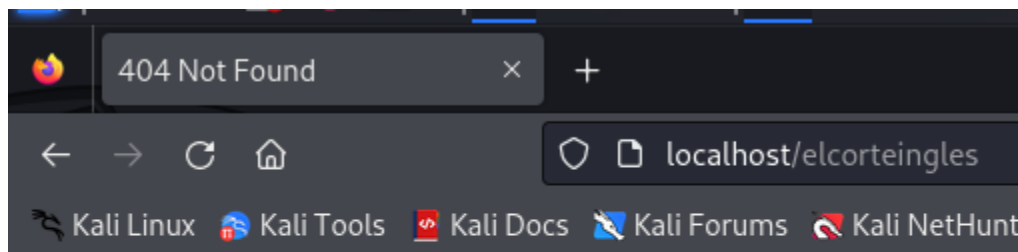
Para verificar si funciona accederemos al navegador a la url: localhost/zara.com



Forbidden

You don't have permission to access this resource.

Si accedemos a cualquier otro sitio inexistente podemos ver que el error es diferente.



Not Found

The requested URL was not found on this server.

Por último solo falta verificar donde están los logs generados. Para ello accedemos a la carpeta 'var/log/apache2' y abrir el fichero error.log . Allí podemos verificar los intentos de conexión que realizamos mientras hacíamos las pruebas.

```
File Actions Edit View Help
tform-multi"] [tag "attack-protocol"] [tag "paranoia-level/1"] [tag "OWASP_CRS"] [tag "capec/1000/210/272"] [tag "PCI/6.5.10"] [hostname "localhost"] [uri "/"
zara.com"] [unique_id "Zj3kX6eqHwmcU3L7ea1EbgAAAAE"]
[Fri May 10 05:09:51.193059 2024] [security2:error] [pid 16773] [client 127.0.0.1:46576] [client 127.0.0.1] ModSecurity: Access denied with code 403 (phase 2
). Operator GE matched 5 at TX:blocking_inbound_anomaly_score. [file "/etc/modsecurity/rules/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "233"] [id "949110"
] [msg "Inbound Anomaly Score Exceeded (Total Score: 5)"] [ver "OWASP_CRS/4.3.0-dev"] [tag "anomaly-evaluation"] [tag "OWASP_CRS"] [hostname "localhost"] [ur
i "/"zara.com"] [unique_id "Zj3kX6eqHwmcU3L7ea1EbgAAAAE"]
[Fri May 10 05:09:51.193352 2024] [security2:error] [pid 16773] [client 127.0.0.1:46576] [client 127.0.0.1] ModSecurity: Warning. Unconditional match in SecA
ction. [file "/etc/modsecurity/rules/RESPONSE-980-CORRELATION.conf"] [line "98"] [id "980170"] [msg "Anomaly Scores: (Inbound Scores: blocking=5, detection=5
, per_pl=5-0-0-0, threshold=5) - (Outbound Scores: blocking=0, detection=0, per_pl=0-0-0-0, threshold=4) - (SQLI=0, XSS=0, RFI=0, LFI=0, RCE=0, PHPI=0, HTTP=
0, SESS=0, COMBINED_SCORE=5)"] [ver "OWASP_CRS/4.3.0-dev"] [tag "reporting"] [tag "OWASP_CRS"] [hostname "localhost"] [uri "/"zara.com"] [unique_id "Zj3kX6eqH
wmcU3L7ea1EbgAAAAE"]
[Fri May 10 05:10:15.375131 2024] [security2:error] [pid 16774] [client 127.0.0.1:54380] [client 127.0.0.1] ModSecurity: Warning. String match within ".asa/
.asax/.ascx/.backup/.bak/.bat/.cdx/.cer/.cfg/.cmd/.com/.config/.conf/.cs/.csproj/.csr/.dat/.db/.dbf/.dll/.dos/.htr/.htw/.ida/.idc/.id
q/.inc/.ini/.key/.licx/.lnk/.log/.mdb/.old/.pass/.pdb/.pol/.printer/.pwd/.rdb/.resources/.resx/.sql/.swp/.sys/.vb/.vbs/.vbproj/.vsdisc
o/.webinfo/.xsd/.xsl/" at TX:extension. [file "/etc/modsecurity/rules/REQUEST-920-PROTOCOL-ENFORCEMENT.conf"] [line "1112"] [id "920440"] [msg "URL file e
xtension is restricted by policy"] [data ".com"] [severity "CRITICAL"] [ver "OWASP_CRS/4.3.0-dev"] [tag "application-multi"] [tag "language-multi"] [tag "pla
tform-multi"] [tag "attack-protocol"] [tag "paranoia-level/1"] [tag "OWASP_CRS"] [tag "capec/1000/210/272"] [tag "PCI/6.5.10"] [hostname "localhost"] [uri "/"
elcorteingles.com"] [unique_id "Zj3kdyDAX6WmJfHd975KRwAAAAI"]
[Fri May 10 05:10:15.376812 2024] [security2:error] [pid 16774] [client 127.0.0.1:54380] [client 127.0.0.1] ModSecurity: Access denied with code 403 (phase 2
). Operator GE matched 5 at TX:blocking_inbound_anomaly_score. [file "/etc/modsecurity/rules/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "233"] [id "949110"
] [msg "Inbound Anomaly Score Exceeded (Total Score: 5)"] [ver "OWASP_CRS/4.3.0-dev"] [tag "anomaly-evaluation"] [tag "OWASP_CRS"] [hostname "localhost"] [ur
i "/"elcorteingles.com"] [unique_id "Zj3kdyDAX6WmJfHd975KRwAAAAI"]
[Fri May 10 05:10:15.377095 2024] [security2:error] [pid 16774] [client 127.0.0.1:54380] [client 127.0.0.1] ModSecurity: Warning. Unconditional match in SecA
ction. [file "/etc/modsecurity/rules/RESPONSE-980-CORRELATION.conf"] [line "98"] [id "980170"] [msg "Anomaly Scores: (Inbound Scores: blocking=5, detection=5
, per_pl=5-0-0-0, threshold=5) - (Outbound Scores: blocking=0, detection=0, per_pl=0-0-0-0, threshold=4) - (SQLI=0, XSS=0, RFI=0, LFI=0, RCE=0, PHPI=0, HTTP=
0, SESS=0, COMBINED_SCORE=5)"] [ver "OWASP_CRS/4.3.0-dev"] [tag "reporting"] [tag "OWASP_CRS"] [hostname "localhost"] [uri "/"elcorteingles.com"] [unique_id "
Zj3kdyDAX6WmJfHd975KRwAAAAI"]
```