

Índice:

Resumen informe de INCIBE (INTECCO): "Guía básica para la securización del servidor Web Apache"	2
a) Instalación	2
b) Configuración usuario y grupo	2
c) Ocultar versiones	3
d) Exposición mínima de módulos	4
e) Directiva "Options"	5
f) Directiva "Auth*" y "Require"	6
g) Fichero .htaccess	8
h) Módulo "mod_security"	9
Fuente	13

Resumen informe de INCIBE (INTECCO): “Guía básica para la securización del servidor Web Apache”.

a) Instalación

Solución:

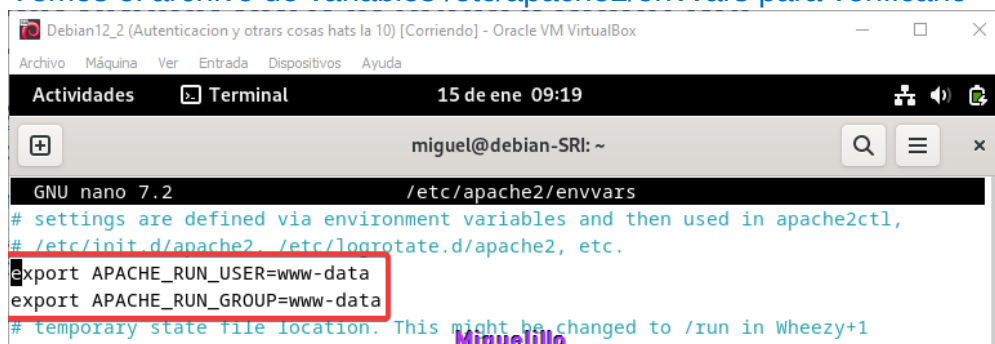
A la hora de instalara apache es recomendable borrar el fichero predeterminado index.htmls ya que puede exponer la configuración de nuestro servidor apache. Que por defecto se encuentra en `/var/www/html/index.html`

b) Configuración usuario y grupo

Solución:

Si arrancamos el servidor web con root corremos el riesgo que frente a un ataque exitoso poder otorgar permisos de administrador al atacante, con lo cual es recomendable configurar los servicios acorde a un usuario con las funcionalidades justas para minimizar el impacto de un ataque. Por defecto viene configurado pero es importante verificar que el usuario y el grupo sea **www-data**

Vemos el archivo de variables `/etc/apache2/envvars` para verificarlo

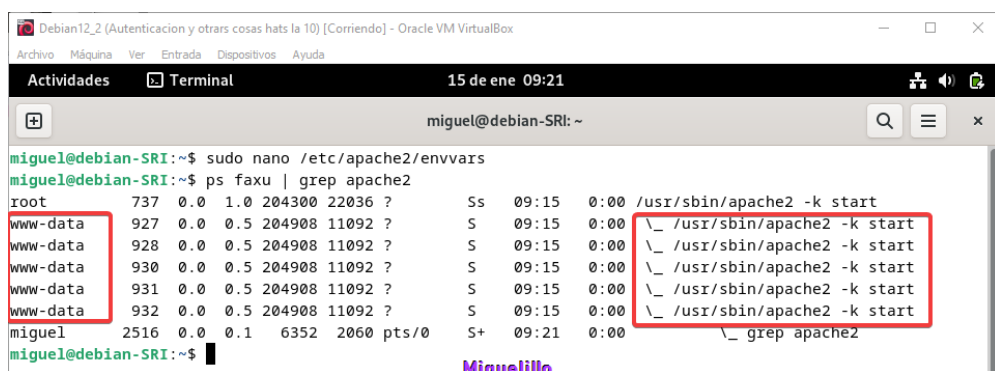


```

GNU nano 7.2 /etc/apache2/envvars
# settings are defined via environment variables and then used in apache2ctl,
# /etc/init.d/apache2, /etc/logrotate.d/apache2, etc.
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data
# temporary state file location. This might be changed to /run in Wheezy+1

```

Y miramos los procesos para verificar la ejecución con ese usuario



```

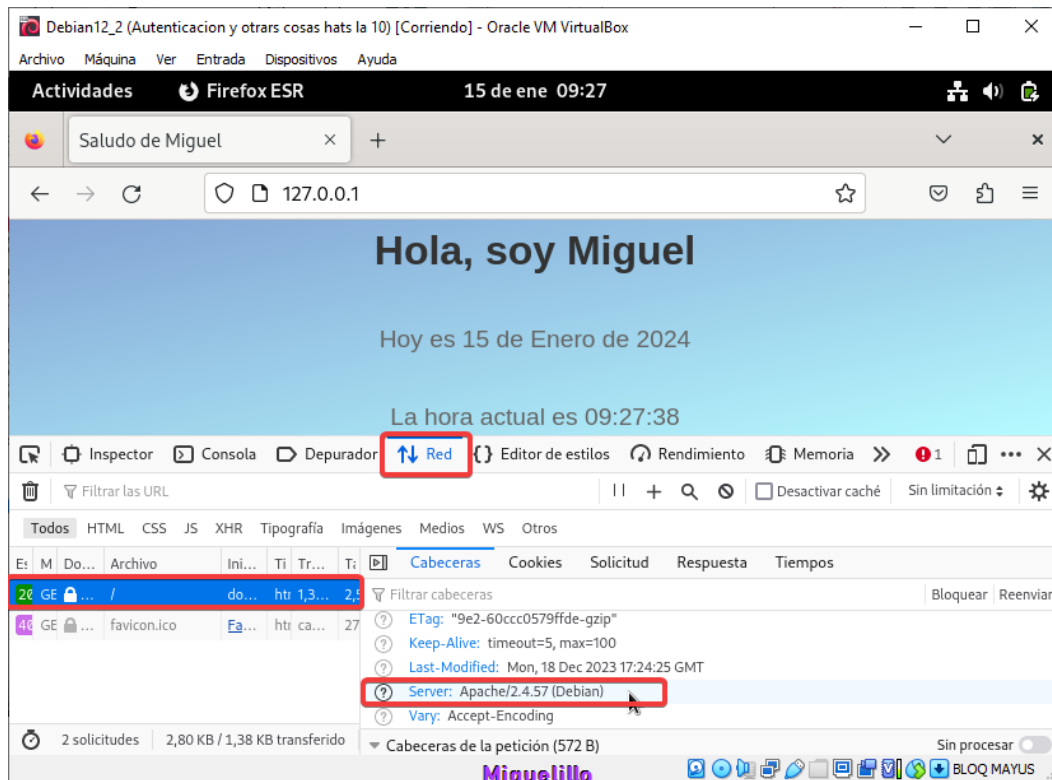
miguel@debian-SRI:~$ sudo nano /etc/apache2/envvars
miguel@debian-SRI:~$ ps faxu | grep apache2
root      737   0.0   1.0 204300 22036 ?        Ss   09:15   0:00 /usr/sbin/apache2 -k start
www-data  927   0.0   0.5 204908 11092 ?        S    09:15   0:00 \_ /usr/sbin/apache2 -k start
www-data  928   0.0   0.5 204908 11092 ?        S    09:15   0:00 \_ /usr/sbin/apache2 -k start
www-data  930   0.0   0.5 204908 11092 ?        S    09:15   0:00 \_ /usr/sbin/apache2 -k start
www-data  931   0.0   0.5 204908 11092 ?        S    09:15   0:00 \_ /usr/sbin/apache2 -k start
www-data  932   0.0   0.5 204908 11092 ?        S    09:15   0:00 \_ /usr/sbin/apache2 -k start
miguel    2516  0.0   0.1   6352  2060 pts/0    S+   09:21   0:00 \_ grep apache2
miguel@debian-SRI:~$

```

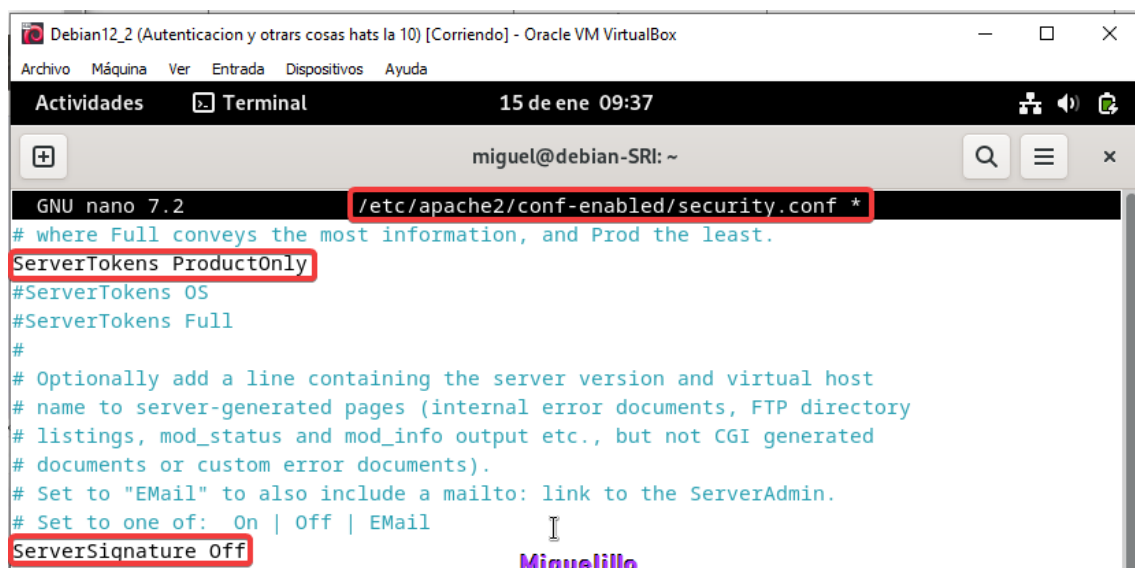
c) Ocultar versiones

Solución:

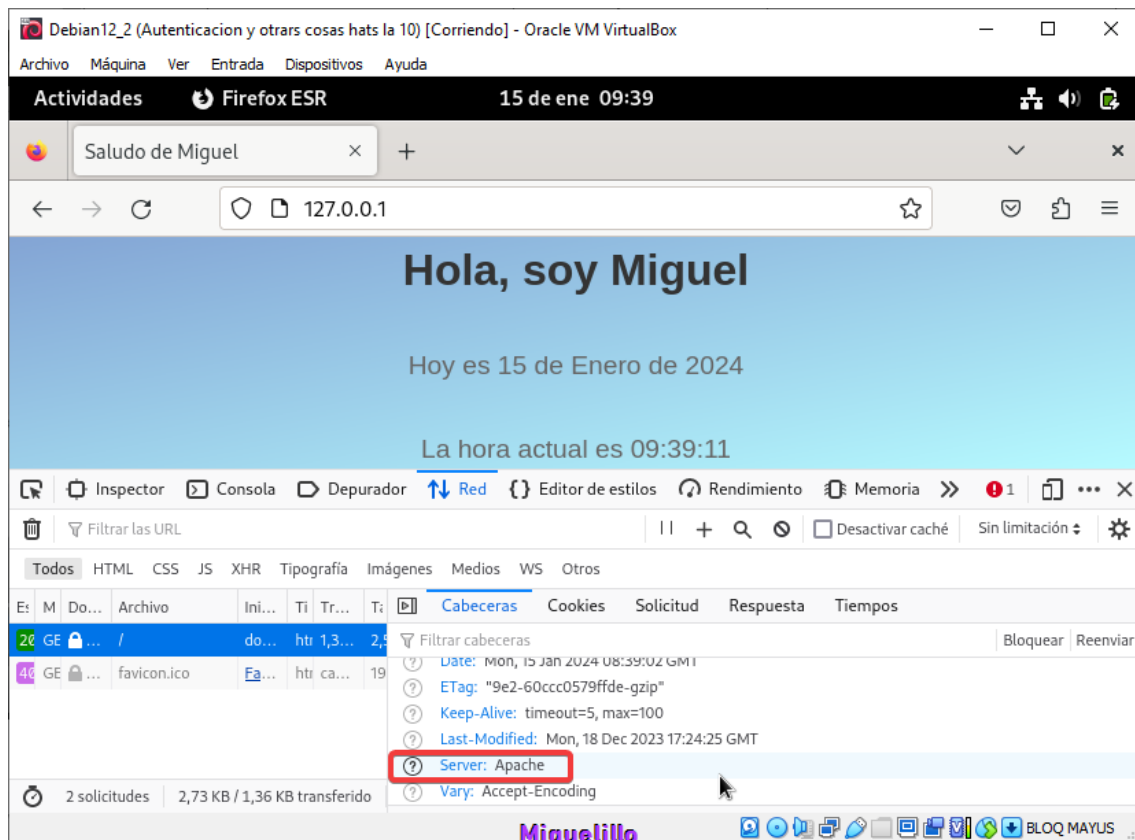
El nombre de producto y versión permite a los atacantes obtener información sobre la versión permitiéndoles buscar exploits. Por defecto son visibles si inspeccionamos la pagina vamos a networks y vamos a / y a Server podemos ver la versión y sistema operativo



Para deshabilitarlo editaremos el archivo `/etc/apache2/conf-enabled/security.conf` y comentaremos con un `#` las directivas `"ServerSignature On"` y `"ServerTokens OS"`. Remplazandolas (descomentando) por `"ServerSignature Off"` y `"ServerTokens ProductOnly"`



Una vez reiniciado el servicio y aplicado podemos ver que no se ve la versión solamente el servidor web que se está utilizando



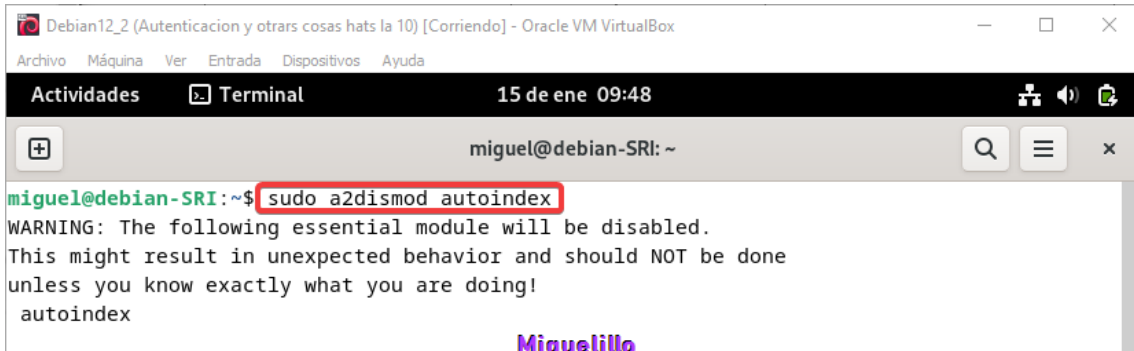
d)Exposición mínima de módulos

Solución:

Desactivar módulos por defecto que no estemos utilizando reducirá la exposición y la carga del servidor. Para ver los módulos por defecto podemos hacer un `sudo apache2ctl -M`



Para desactivarlo usamos *a2dismod*. Es importante que investiguemos que hace cada modulo porque podemos dejar el servidor no operativo



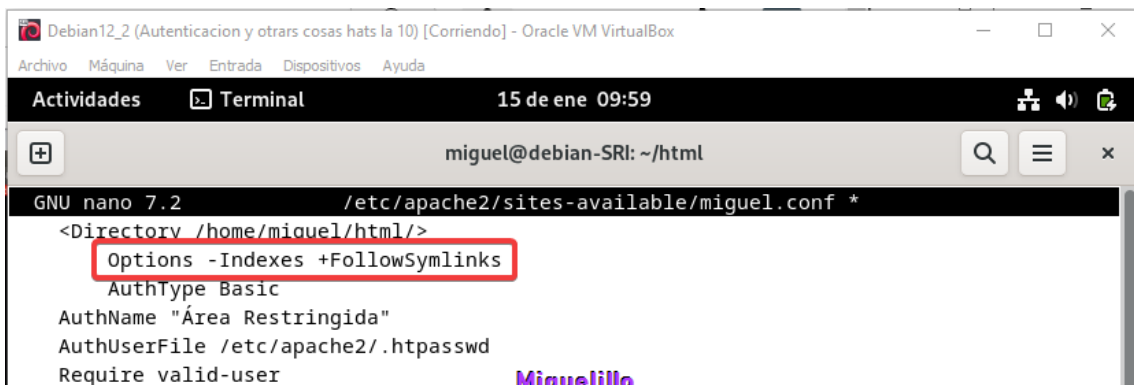
```
Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 15 de ene 09:48
miguel@debian-SRI: ~
miguel@debian-SRI:~$ sudo a2dismod autoindex
WARNING: The following essential module will be disabled.
This might result in unexpected behavior and should NOT be done
unless you know exactly what you are doing!
autoindex
```

Miguelillo

e) Directiva "Options"

Solución:

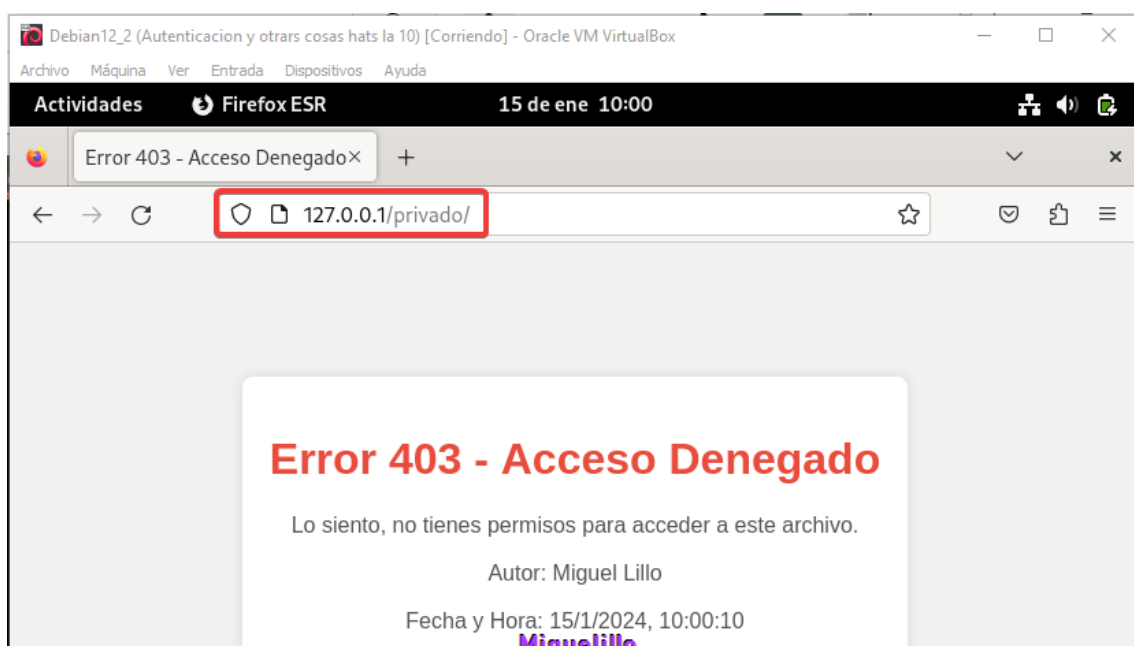
Si no queremos que se indexen los ficheros de un directorio usamos la opción "Options -Indexes +FollowSymlinks"



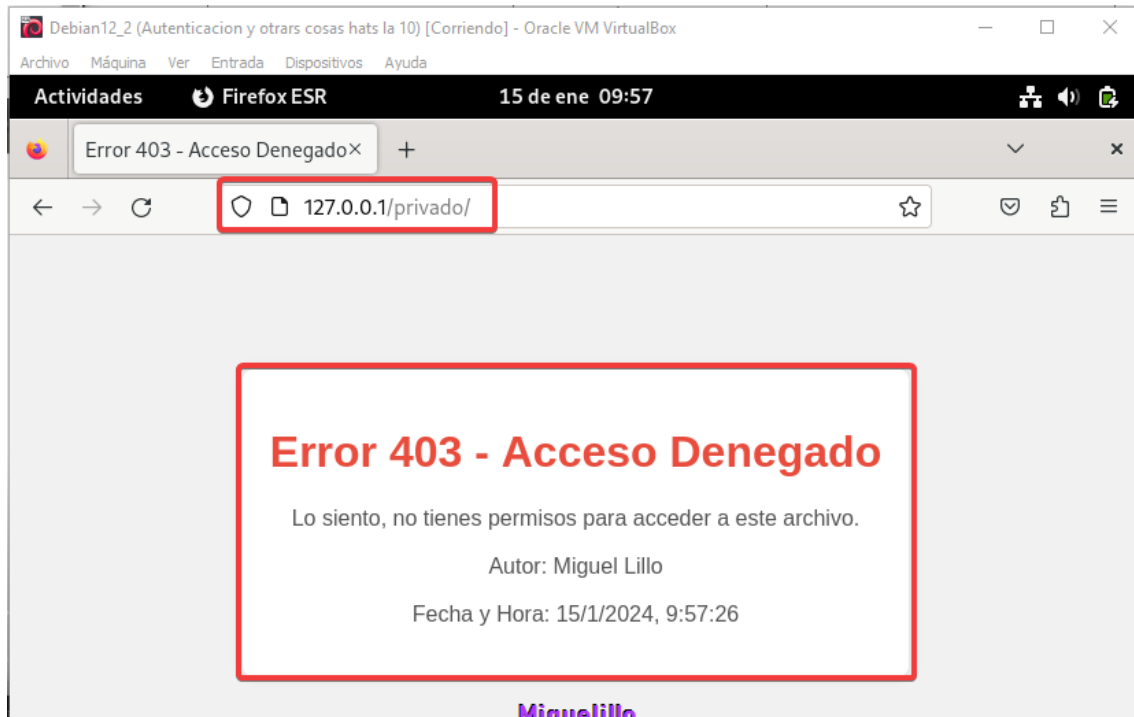
```
Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 15 de ene 09:59
miguel@debian-SRI: ~/html
GNU nano 7.2 /etc/apache2/sites-available/miguel.conf *
<Directory /home/miguel/html/>
  Options -Indexes +FollowSymlinks
  AuthType Basic
  AuthName "Área Restringida"
  AuthUserFile /etc/apache2/.htpasswd
  Require valid-user
```

Miguelillo

Si probamos a acceder nos da un fallo por permisos



Miguelillo



f) Directiva “Auth*” y “Require”

Solución:

Si queremos proteger la web podemos poner directivas de autenticación para ello generamos una contraseña con `sudo htpasswd -c passwords inves` D
Directivas Auth

Directivas Auth*

1. **AuthType Basic:** Es el tipo de autenticación más básico, codificado en Base64. Es recomendable usarlo en conexiones cifradas mediante SSL.
2. **AuthName “Acceso restringido”:** Cadena de texto que aparecerá en el cuadro de diálogo.
3. **AuthBasicProvider file:** Proveedor de autenticación, en este caso un fichero generado con la herramienta htpasswd.
4. **AuthUserFile “password”:** El fichero de claves generado anteriormente.
5. **Require user incibe:** El usuario “incibe” será el único que podrá acceder.

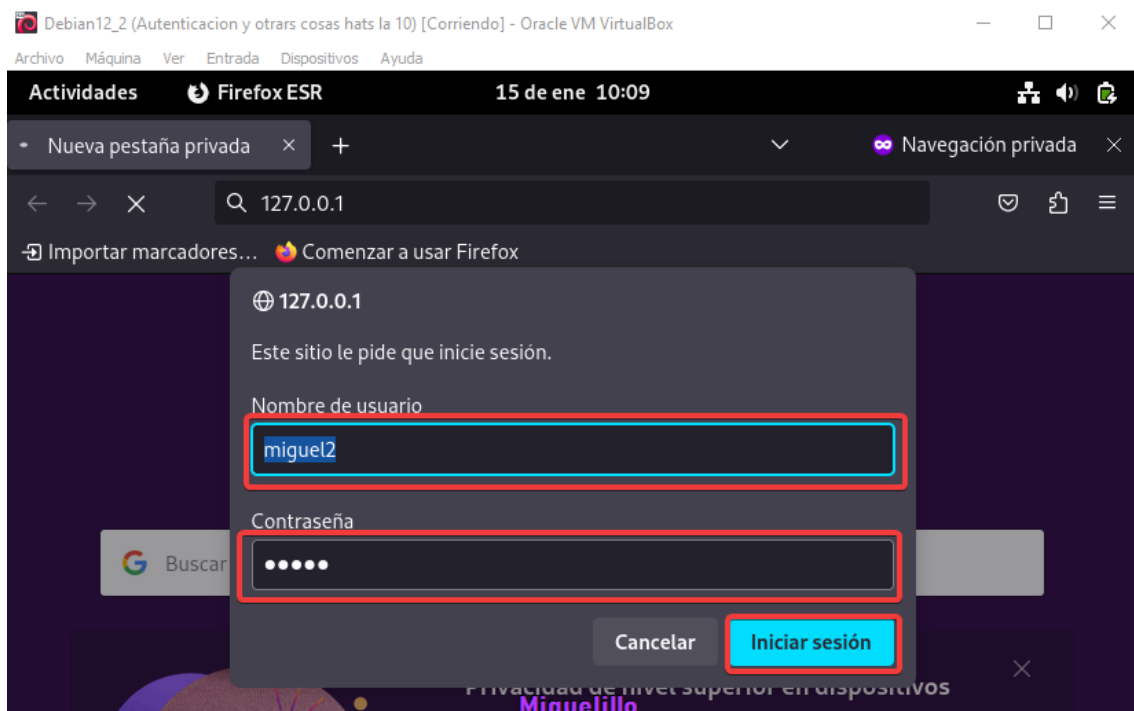
Miguelillo

Configuramos las directivas

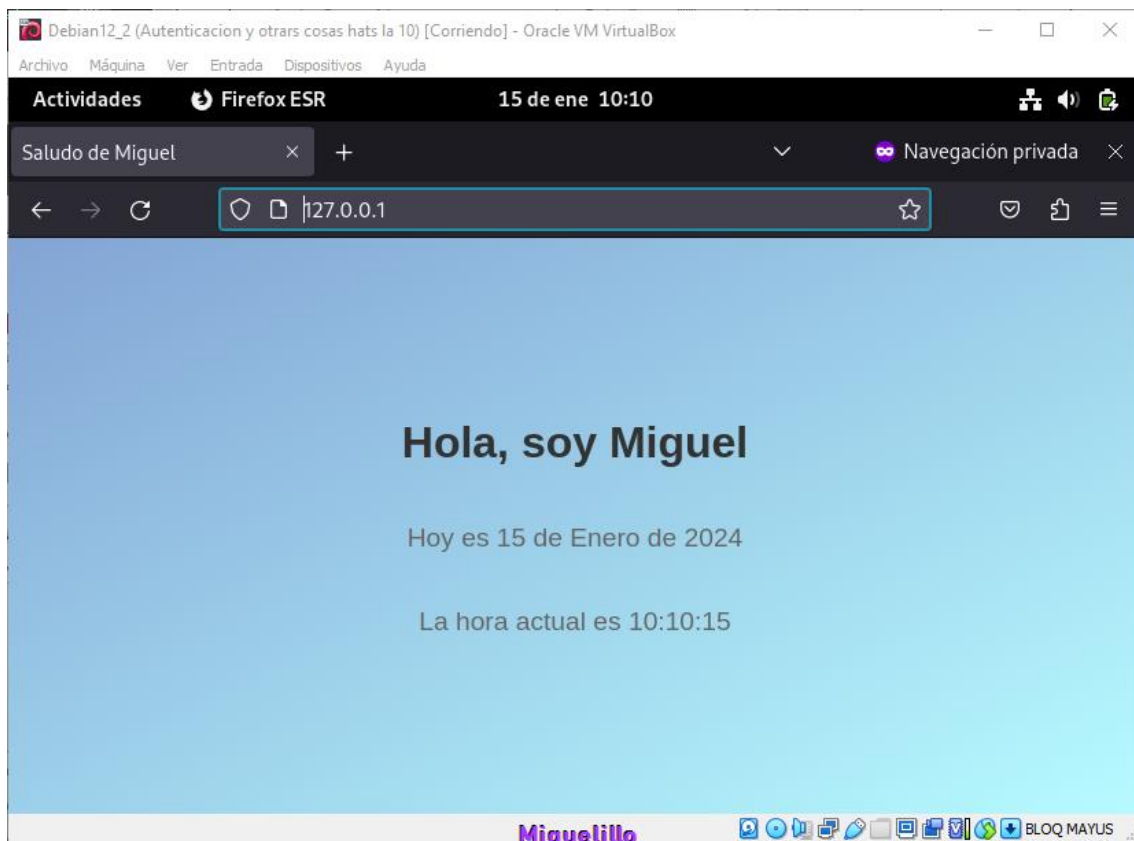


Miguelillo

Nos pide las credenciales



Accedemos



Con `required` podemos restringir el acceso al contenido por host por ip por usuario por grupo. Por ejemplo en este ejemplo se deniega el acceso al local host pero se permite el acceso a todos los demás usuarios. Con lo cual podemos usarlo de manera jerárquica

```
<Directory /var/www/html/privado>
  <RequireAll>
    Require all granted
    Require not host localhost
  </RequireAll>
</Directory>
```

Miguelillo

Otros tipos de required

Require all granted: Proporciona acceso de forma incondicional a los recursos donde se aplique.

Require all denied: Deniega acceso de forma incondicional a los recursos donde se aplique.

Require env nombre: Proporciona acceso sólo si existe una variable de entorno de nombre "nombre" en el sistema.

Require method http-method: Proporciona acceso a través de métodos HTTP definidos en "httpmethod".

Require [not] ip <IP/IPs>: Proporciona acceso a una lista de IPs. Si incluye [not] deniega el acceso a esa lista.

Require [not] host <host/hosts>: Proporciona acceso a una lista de hosts. Si incluye [not] deniega el acceso a esa lista.

Miguelillo

g)Fichero .htaccess

Solución:

Permiten personalizar directivas y parámetros que algunos pueden ser muy útiles como las siguientes directivas

- **Control de acceso al sitio:** Deniega el acceso a ciertas IPs.
 - *Allow from all*
 - *Deny from 1.2.3.4*
 - *Deny from 12.24*
- **Forzar protocolo HTTPS:** Aplicar un redireccionamiento al protocolo seguro HTTPS.
 - *RewriteEngine on*
 - *RewriteCond %{HTTPS} !=on [NC]*
 - *RewriteRule ^.*\$ https://%{SERVER_NAME}%{REQUEST_URI} [R,L]*
- **Evitar "HotLinking":** El "hotlinking" es el abuso de recursos ajenos a través de enlaces directos desde otras webs con el consiguiente consumo de datos.
 - *RewriteEngine on*
 - *RewriteCond %{HTTP_REFERER} !=^\$*
 - *RewriteCond %{HTTP_REFERER} !=http://(www\.)?midominio.org/.*\$ [NC]*
 - *RewriteRule .*\. (jpe?g|gif|bmp|png)\$ [L]*

Miguelillo

h) Módulo “mod_security”

Solución:

Este módulo protege de ataques web comúnmente conocidos. Actúa como firewall de aplicación. Dispone de modos de actuación activos y pasivos el método pasivo almacena los ataques, pero no los para. Inspecciona el tráfico HTTP y deniega acceso a aquellas peticiones donde encuentra algún patrón de ataque conocido.

Lo instalamos



```
Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Actividades Terminal 15 de ene 10:21
miguel@debian-SRI: ~/html
miguel@debian-SRI:~/html$ sudo apt install libapache2-mod-security2
[sudo] contraseña para miguel:
Leyendo lista de paquetes... Hecho
Miauelillo
```

Activamos el módulo



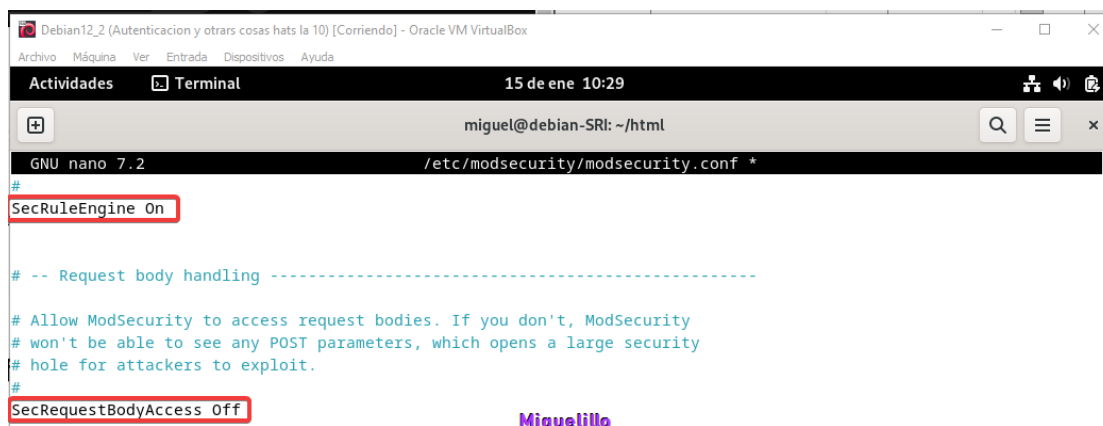
```
Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Actividades Terminal 15 de ene 10:22
miguel@debian-SRI: ~/html
miguel@debian-SRI:~/html$ sudo a2enmod security2
Miauelillo
```

Renombramos el fichero de configuración



```
Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Actividades Terminal 15 de ene 10:27
miguel@debian-SRI: ~/html
miguel@debian-SRI:~/html$ sudo mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf
miguel@debian-SRI:~/html$
Miauelillo
```

Editamos el fichero de configuración y cambiamos “SecRuleEngine” a “On” y “SecRequestBodyAccess” a “Off”

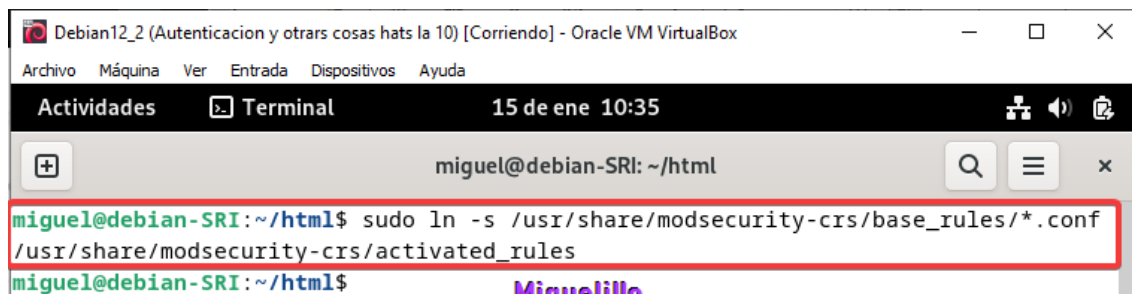


```
Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Actividades Terminal 15 de ene 10:29
miguel@debian-SRI: ~/html
GNU nano 7.2 /etc/modsecurity/modsecurity.conf *
#
SecRuleEngine On

# -- 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 Off
Miauelillo
```

Las reglas están disponible en `/usr/share/modsecurity-crs/base_rules`, `/usr/share/modsecurity-crs/optional_rules` y `/usr/share/modsecurity-crs/experimental_rules`. Para habilitarlas creamos un enlace simbólico

Para activarlas hacemos un enlace simbólico



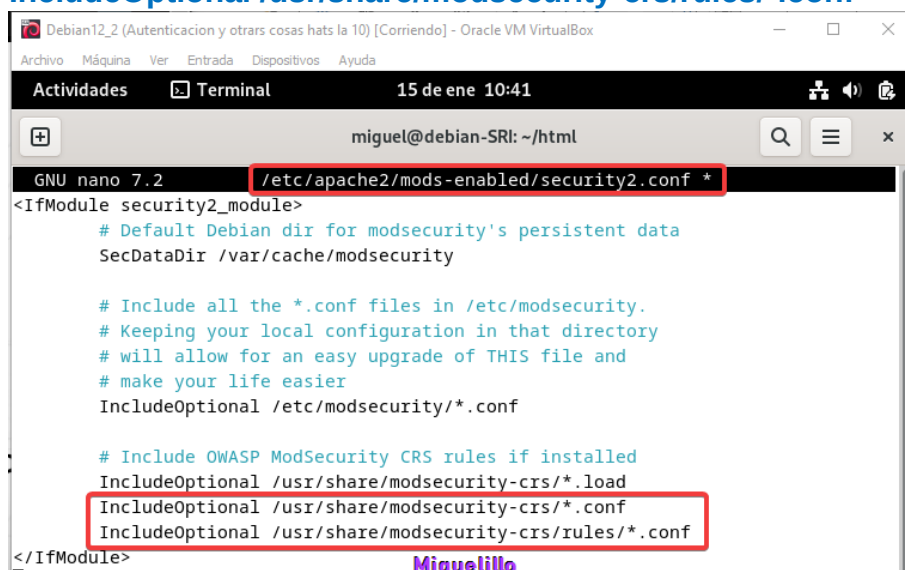
```

Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal  15 de ene 10:35
miguel@debian-SRI: ~/html
miguel@debian-SRI:~/html$ sudo ln -s /usr/share/modsecurity-crs/base_rules/*.conf
/usr/share/modsecurity-crs/activated_rules
miguel@debian-SRI:~/html$
Miguelillo

```

Para añadir las reglas editamos el fichero `/etc/apache2/mods-enabled/security2.conf` y añadimos estas 2 líneas de comando
"IncludeOptional /usr/share/modsecurity-crs/*.conf"

IncludeOptional /usr/share/modsecurity-crs/rules/*.conf"



```

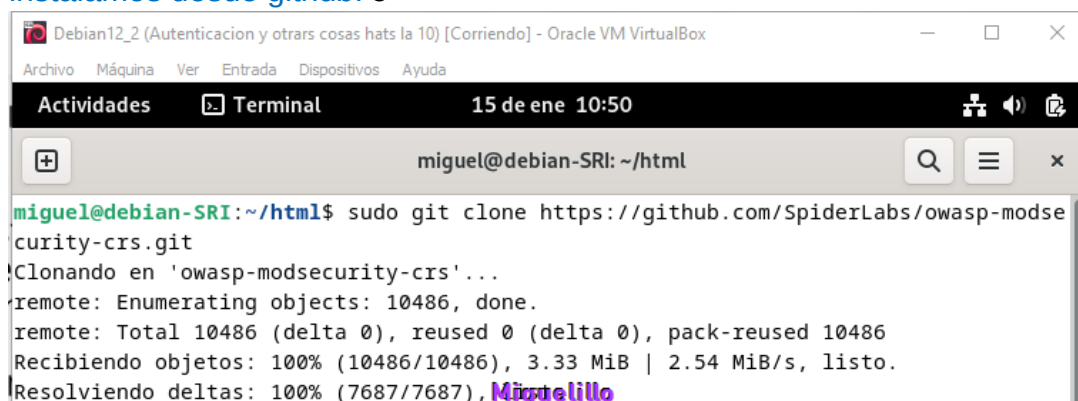
Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal  15 de ene 10:41
miguel@debian-SRI: ~/html
GNU nano 7.2 /etc/apache2/mods-enabled/security2.conf *
<IfModule security2_module>
# Default Debian dir for modsecurity's persistent data
SecDataDir /var/cache/modsecurity

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

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

```

Para descargar las reglas OWASP creadas por la comunidad recomendadas para la mayoría de los ataques web, estas reglas detecta inyecciones SQL y las deniega. También detecta ataques de inclusión de ficheros entre otros las instalamos desde github. c



```

Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal  15 de ene 10:50
miguel@debian-SRI: ~/html
miguel@debian-SRI:~/html$ sudo git clone https://github.com/SpiderLabs/owasp-modse
curity-crs.git
Clonando en 'owasp-modsecurity-crs'...
remote: Enumerating objects: 10486, done.
remote: Total 10486 (delta 0), reused 0 (delta 0), pack-reused 10486
Recibiendo objetos: 100% (10486/10486), 3.33 MiB | 2.54 MiB/s, listo.
Resolviendo deltas: 100% (7687/7687), Miguelillo

```

Cambiamos le nombre al directorio y movemos el archivo que hemos descargado

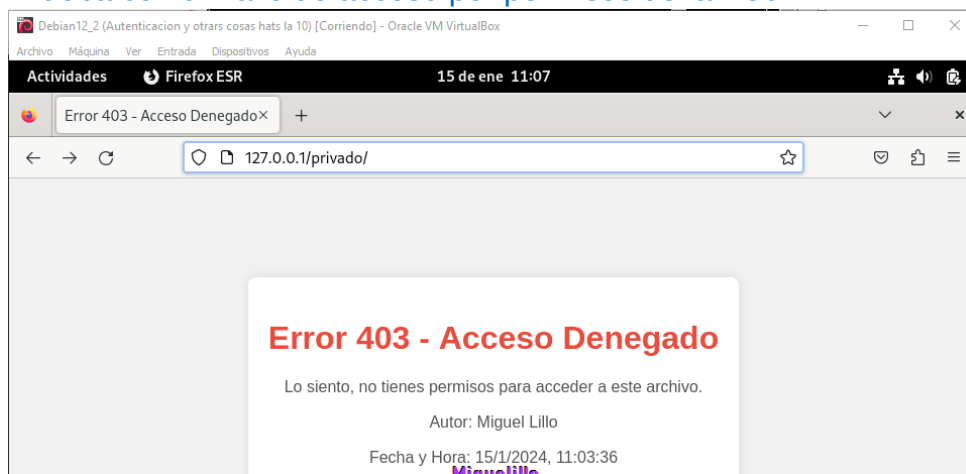
```
Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 15 de ene 10:56
miguel@debian-SRI: ~/html
miguel@debian-SRI:~/html$ sudo mv /usr/share/modsecurity-crs /usr/share/modsecurity-crs.git
miguel@debian-SRI:~/html$ sudo mv owasp-modsecurity-crs /usr/share/modsecurity-crs
```

Renombramos el fichero del interior

```
Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 15 de ene 10:58
miguel@debian-SRI: ~/html
miguel@debian-SRI:~/html$ sudo mv /usr/share/modsecurity-crs/crs-setup.conf.example /usr/share/modsecurity-crs/crs-setup.conf
```

Reiniciamos el servicio apache y para monitorear ponemos `sudo tail -f /var/log/apache2/modsec_audit.log` dependiendo lo que queramos monitorizar pondremos filtros al final del comando como `| grep ATTACK-SQL`

Prueba con un fallo de acceso por permisos de la web



Este registro se va actualizando dinámicamente hasta que lo cierras con **Ctrl + C**

```
Debian12_2 (Autenticacion y otras cosas hats la 10) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal 15 de ene 11:03
miguel@debian-SRI: ~/html
root@debian-SRI: /home/miguel/html# tail -f /var/log/apache2/modsec_audit.log
--ee3ce472-A--
[15/Jan/2024:11:03:34.890619 +0100] ZaUC9n0N4CcTvprzby2nWgAAAAA 127.0.0.1 38834 127.0.0.1 80
--ee3ce472-B--
GET /privado/ HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Authorization: Basic bWlndWVsMjppbnZlcw==
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
```

Un ejemplo de un ataque SQL

[illegible]

Fuente

<https://youtu.be/rjsgaxV8irM?si=8YJ9egRFiJV3EHjX>