Actividad 2: Implementación de HTTP Strict Transport Security (HSTS)

Tema: *Protección contra ataques MITM*

Objetivo: *Configurar HSTS para forzar HTTPS y prevenir ataques de degradación a HTTP*

¿Qué es HSTS?

HSTS (HTTP Strict Transport Security) obliga al navegador a usar **únicamente HTTPS**, reduciendo riesgos de **ataques MITM y downgrade a HTTP**.

Configurar HSTS en el servidor web

En Apache:

```
Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"
```

- **max-age=31536000**: Obliga el uso de HTTPS durante 31,536,000 segundos (1 año).
- **includeSubDomains**: Aplica HSTS a todos los subdominios.
- **preload**: Solicita la inclusión en la lista de precarga de HSTS https://hstspreload.org/ de Google Chrome.

El parámetro *preload* hace que el dominio sea agregado a la **lista de precarga de HSTS** en navegadores como Chrome, Firefox y Edge. Una vez en esta lista, **no se puede eliminar fácilmente** y todos los visitantes siempre intentarán conectarse por HTTPS, incluso si se elimina el encabezado HSTS en el servidor.

Cuando **NO** usar *preload*:

- Si aún tienes subdominios que solo funcionan en HTTP.
- Si no quieres un compromiso permanente con HTTPS en todo tu dominio.
- Si recién estás implementando HSTS y quieres probarlo primero.

Cuando **SÍ** usar *preload*:

- Si todos los subdominios ya usan HTTPS correctamente.
- Si planeas mantener HTTPS de forma **permanente** y sin excepciones.
- Si quieres que los navegadores siempre accedan a tu dominio de manera segura, incluso en la primera visita.

Pasos para Configurar HSTS en Localhost con Apache

Habilitar el módulo *headers*

Si no está habilitado, ejecutar:

```
sudo a2enmod headers
sudo systemctl restart apache2
```

Generar un *certificado SSL* (Opcional)

Si no se dispone de un certificado SSL, se puede crear uno *autofirmado*:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
-keyout /etc/ssl/private/localhost.key \
-out /etc/ssl/certs/localhost.crt \
-subj "/CN=localhost"
```

Mover y configurar permisos de los certificados

Asegurarse de que los archivos del certificado y la clave privada tienen los permisos correctos:

```
sudo chmod 600 /etc/ssl/private/localhost.key
sudo chmod 644 /etc/ssl/certs/localhost.crt
```

Esto protege la clave privada y permite que Apache acceda a los certificados.

Habilitar *SSL* en Apache

Si el módulo SSL no está activado en Apache, ejecutar:

```
sudo a2enmod ssl
sudo systemctl restart apache2
```

Configurar Apache en *localhost*

Editar el archivo de configuración para localhost:

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

Agregar o editar las siguientes líneas dentro de *<VirtualHost _default_:443>*:

```
<VirtualHost _default_:443>
    ServerName localhost

    # Habilitar HSTS en localhost
    Header always set Strict-Transport-Security "max-age=31536000"

    DocumentRoot /var/www/html
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/localhost.crt
    SSLCertificateKeyFile /etc/ssl/private/localhost.key
```

```
</VirtualHost>
```

NO usar *includeSubDomains* **ni** *preload*, ya que no aplican en *localhost*.

Habilitar el sitio SSL y reiniciar Apache

Si aún no se tiene habilitado el sitio SSL en Apache, ejecutar:

```
sudo a2ensite default-ssl
sudo systemctl restart apache2
```

Probar que HSTS funciona correctamente

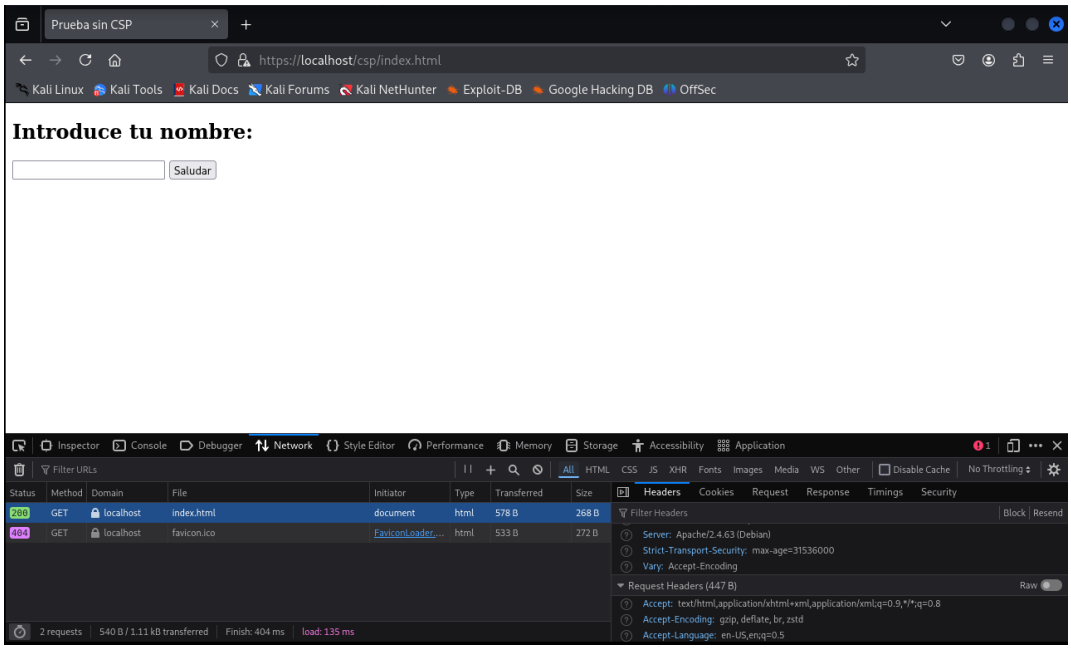
Ejecutar en la terminal:

```
curl -I https://localhost --insecure
```

Se debería obtener una respuesta con:

```
Strict-Transport-Security: max-age=31536000
```

- HSTS **NO** se aplicará en **localhost** en **Chrome** o **Firefox** por defecto.
- Solo servirá si accedes con *https://localhost* y confías en el certificado.
- Si se necesita una implementación real, es mejor probar en un **dominio de desarrollo con HTTPS**.



Mitigación y Mejores Prácticas

- Habilitar HSTS solo en sitios completamente migrados a HTTPS.
- Usar *preload* para asegurar que el navegador recuerde la configuración incluso después de cerrar la sesión.