Guía de Seguridad

Fecha de publicación: 02/06/2020

Tema: HSTS, un mecanismo de seguridad adicional a HTTPs.

Contenido:

---

1. ¿Qué es HSTS? (HTTP Strict Transport Security)

HSTS es un mecanismo de seguridad para la protección contra ataques hombre-en-el-medio (**man-in-the-middle)** que solo permite a los navegadores las conexiones con el protocolo **HTTPs** en lugar de **HTTP**. Si la cabecera de **HSTS** está activada en el servidor web, este se encarga de informar al navegador que no debe cargar un sitio que utilice el protocolo **HTTP** y automáticamente convierte todos los intentos de acceso al sitio a solicitudes **HTTPs**.

HSTS permite protegernos de un ataque del tipo **SSLstrip**, forzando la conexión segura desde el inicio de la sesión, evitando que se pueda interceptar la conexión entre el cliente y el navegador antes de que se establezca la conexión segura. La mayoría de los servidores web soportan la configuración HSTS.

2. ¿Cómo funciona un ataque como SSLStrip?

SSLStrip es un tipo de ataque MITM (del inglés., **man-in-the-middle** - hombre-en-el-medio), mediante el cual el atacante intercepta la conexión entre el cliente y el navegador en el momento previo al establecimiento de la conexión HTTPs y fuerza que la víctima vaya por HTTP en lugar del HTTPS, es decir, «transforma» el HTTPS en HTTP para que con un sniffer o analizador de protocolos (como Wireshark) obtener toda la información que se está transmitiendo.

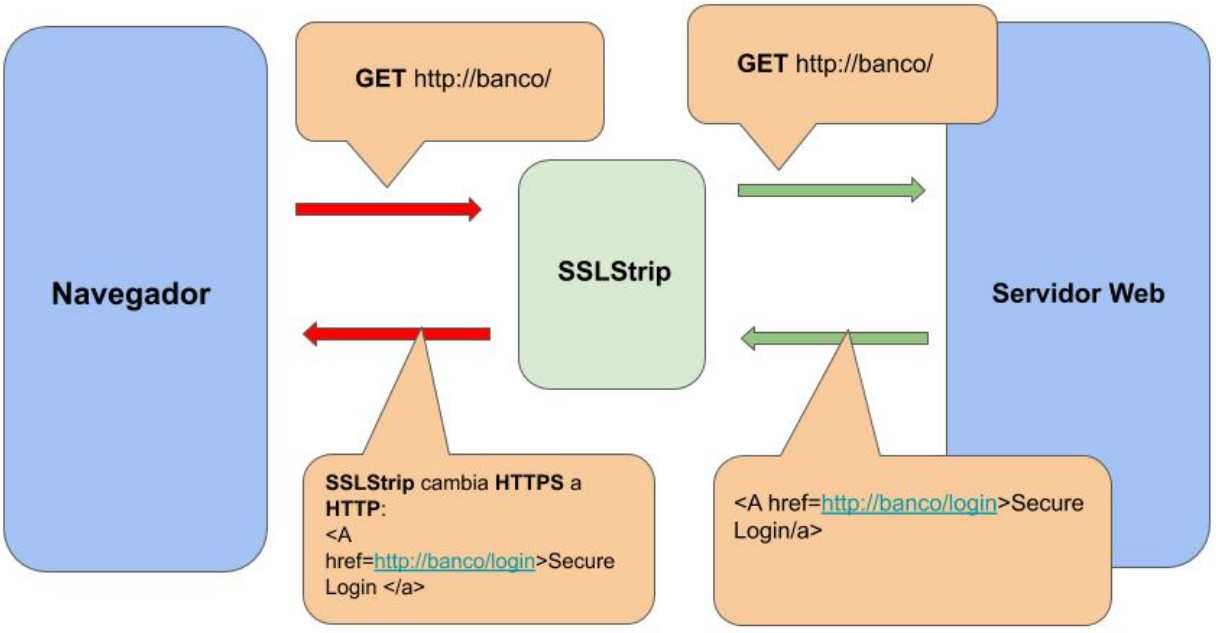


Figura 1: Esquema de peticiones al inicio del ataque SSLStrip

De esta manera la víctima y el atacante se comunican a través del protocolo **HTTP**, mientras que el atacante y el servidor se comunican por el protocolo **HTTPS** con el certificado del servidor, de tal manera a que el servidor no pueda percibir el ataque. Con esto el atacante podría ver todo los datos que viajan a través de la red en **texto plano** transformando toda solicitud por el protocolo **HTTPS** al protocolo **HTTP** con SSLStrip sin levantar sospechas.

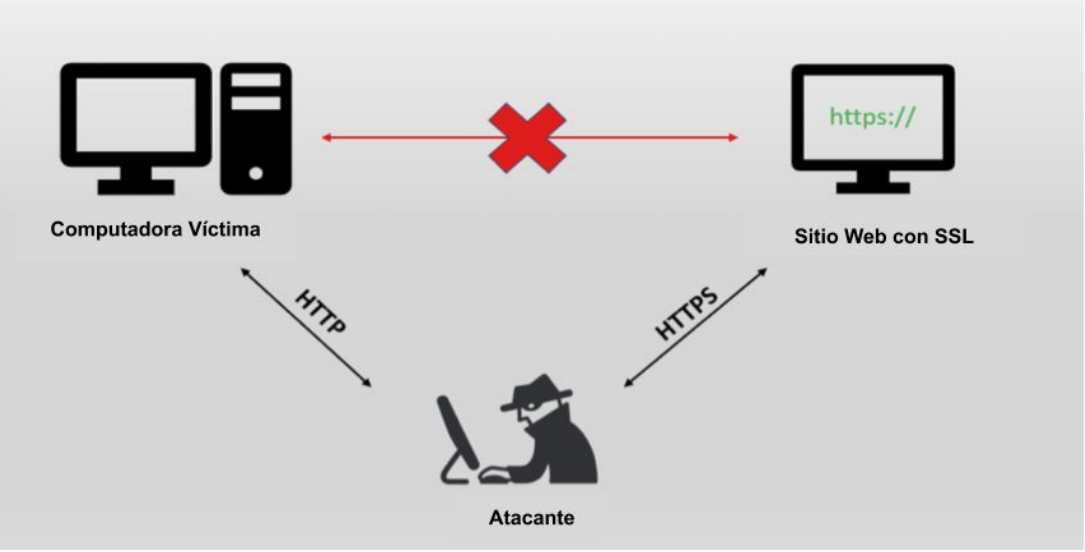


Figura 2: Esquema del ataque cuando se encuentra en curso.

3. ¿Cómo maneja el navegador la cabecera?

La primera vez que se accede a un sitio web utilizando el protocolo **HTTPS**, éste retorna una cabecera **Strict-Transport-Security**, que el navegador registra con el fin de que en futuros intentos de carga del sitio usando el protocolo **HTTP** redirija automáticamente a la utilización del protocolo **HTTPS**.

Sin embargo, pasado el tiempo de expiración especificado en la cabecera, el proceso

volverá a su normalidad. Por ello, en cualquier momento que se entrega información sobre la cabecera **Strict-Transport-Security** al navegador, éste modifica el **tiempo de expiración** para el sitio, refrescando la información.

4. Implementación de HSTS en algunos de los servidores web más utilizados.

4.1. Apache

- Activar el módulo de cabecera Apache **mod_headers**, introduciendo el siguiente comando en la terminal con el privilegio de **administrador**:

```
sudo a2enmod headers
```

- Luego, reiniciar el servidor web para que los cambios se efectúen, con el siguiente comando:

```
sudo service apache restart
```

- Finalmente, para configurar **HSTS** y garantizar siempre una conexión cifrada con el protocolo **HTTPS** como un operador web, añadir la siguiente línea de comando en el archivo de configuración de Apache "**https.conf**":

```
Header always set Strict-Transport-Security "max-age=4838400;
includeSubdomains;"
```

- Junto con otras directivas del contenedor **VirtualHost**:

```
<VirtualHost *:443>
    ServerAdmin support@example.com
    ServerName www.example.com
    SSLEngine on
    SSLCertificateFile /path/to/www.example.com.cert
    SSLCertificateKeyFile /path/to/www.example.com.key
    [...]
    Header always set Strict-Transport-Security "max-age=4838400;
includeSubdomains;"
    DocumentRoot /var/www/
</VirtualHost>
```

4.2. NGINX

NGINX permite generar conexiones cifradas **SSL/TLS** con tan solo añadir la siguiente línea de código dentro del archivo **/etc/nginx/nginx.conf** :

- add_header Strict-Transport-Security "max-age=4838400; includeSubDomains";
- Y, para definir las directivas se utilizan los llamados **server blocks** como se puede ver en la imagen:

```
server {
listen 443 ssl;
ssl_certificate www.example.com.crt;
ssl_certificate www.example.com.key;
```

```
[...]

add_header Strict-Transport-Security "max-age=4838400;
includeSubDomains";
}
```

4.3. IIS

- **Para las versiones anteriores a 1709 de IIS 10.0**, debe seguir los siguientes pasos:

    a. Redirigir todo tráfico **HTTP** a **HTTPS** utilizando el **HTTP Redirect Module** con dos sitios web separados, uno para **HTTP** y el otro para **HTTPS** para evitar un ciclo infinito. A continuación se detalla un código **XML** de ejemplo:

```
<sites>
    <site name="Contoso-http" id="1" serverAutoStart="true">
        <application path="/" applicationPool="Contoso-http">
            <virtualDirectory path="/"
physicalPath="C:\inetpub\Contoso-http" />
        </application>
        <bindings>
            <binding protocol="http"
bindingInformation="*:80:contoso.com" />
        </bindings>
    </site>
```

```
    <site name="Contoso-https" id="2" serverAutoStart="true">
        <application path="/" applicationPool="Contoso-https">
            <virtualDirectory path="/"
physicalPath="C:\inetpub\Contoso-https" />
        </application>
        <bindings>
            <binding protocol="https"
bindingInformation="*:443:contoso.com" sslFlags="0" />
        </bindings>
    </site>
    <siteDefaults>
        <logFile logFormat="W3C"
directory="%SystemDrive%\inetpub\logs\LogFiles" />
        <traceFailedRequestsLogging
directory="%SystemDrive%\inetpub\logs\FailedReqLogFiles" />
    </siteDefaults>
    <applicationDefaults applicationPool="DefaultAppPool" />
    <virtualDirectoryDefaults allowSubDirConfig="true" />
</sites>
```

b. Luego, en el archivo **web.config** del sitio **HTTP redirigir** el tráfico al sitio **HTTPS.**

```
 <?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <httpRedirect enabled="true"
```

```
destination="https://contoso.com" httpResponseStatus="Permanent" />
    </system.webServer>
</configuration>
```

c. Finalmente agregar la cabecera **Strict-Transport-Security** a través de **Custom Headers**, dentro del archivo **web.config** del sitio web **HTTPS.**

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <httpProtocol>
            <customHeaders>
                <add name="Strict-Transport-Security"
value="max-age=31536000" />
            </customHeaders>
        </httpProtocol>
    </system.webServer>
</configuration>
```

- **Para IIS 10.0 Versión 1709**, debe seguir los siguientes pasos:
    - Con esta versión, **HSTS** es soportado nativamente, y para activarlo se debe:
        - Configurar los atributos del elemento **<hsts>** debajo de cada elemento **<site>** a través de **IISAdministration PowerShell cmdlets**, como puede ser visualizado en la imagen de ejemplo.

```
Import-Module IISAdministration
Reset-IISServerManager -Confirm:$false
```

```
Start-IISCommitDelay

$sitesCollection = Get-IISConfigSection -SectionPath
"system.applicationHost/sites" | Get-IISConfigCollection
$siteElement = Get-IISConfigCollectionElement -ConfigCollection
$sitesCollection -ConfigAttribute @{"name"="Contoso"}
$hstsElement = Get-IISConfigElement -ConfigElement $siteElement
-ChildElementName "hsts"
Set-IISConfigAttributeValue -ConfigElement $hstsElement
-AttributeName "enabled" -AttributeValue $true
Set-IISConfigAttributeValue -ConfigElement $hstsElement
-AttributeName "max-age" -AttributeValue 31536000
Set-IISConfigAttributeValue -ConfigElement $hstsElement
-AttributeName "redirectHttpToHttps" -AttributeValue $true

Stop-IISCommitDelay
Remove-Module IISAdministration
```

- A continuación se muestra una configuración **HSTS** de ejemplo para un sitio web:

```
    <binding protocol="http" bindingInformation="*:80:contoso.com"
/>
        <binding protocol="https"
bindingInformation="*:443:contoso.com" sslFlags="0" />
    </bindings>
    <hsts enabled="true" max-age="31536000"
redirectHttpToHttps="true" />
```

```
</site>
```

4.4. Lighttpd

Para configurar **HSTS** en un servidor **Lighttpd**, debe seguir los siguientes pasos:

- Abrir el archivo **lighttpd.conf** ubicado en el directorio **/etc/**, con el siguiente comando:
    - sudo vi /etc/lighttpd/lighttpd.conf
    - sudo vi /usr/local/etc/lighttpd/lighttpd.conf (para FreeBSD unix)
- Agregar el módulo **mod_setenv** de la siguiente manera:

```
server.modules += ( "mod_setenv" )
```

- Añadir el header **Strict-Transport-Security** siguiendo esta sintaxis:

```
setenv.add-response-header=("Strict-Transport-Security"=>
"max-age=SECONDS"; includeSubdomains; ")
```

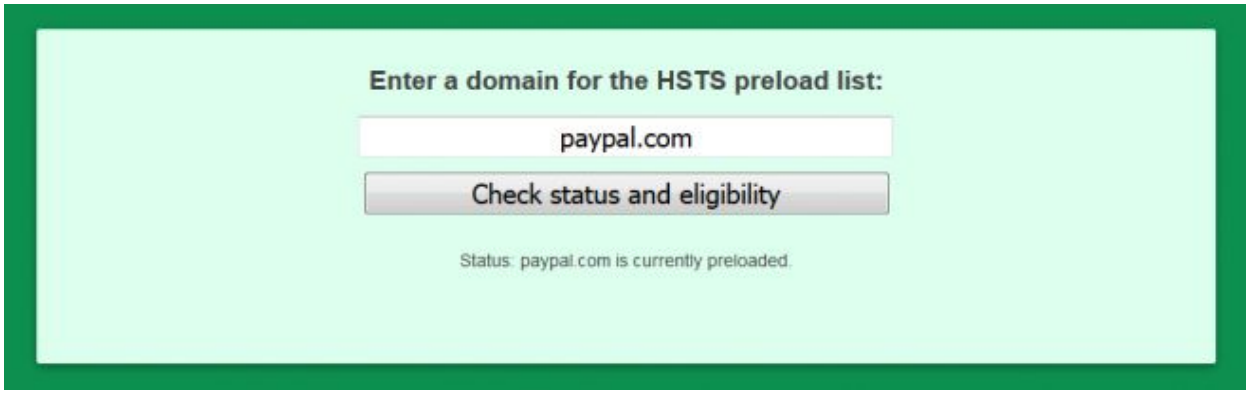
- Guardar y cerrar el archivo,
- Reiniciar el servidor web con el siguiente comando:
    - sudo service lighttpd restart

5. HSTS Preload, para comprobar la implementación de HSTS.

Es también posible reforzar la conexión segura por el protocolo **HTTPS** incluso antes de visitar un sitio web por primera vez a través de la **HSTS Preload list**, una lista de dominios que soportan por defecto **HSTS**. Esta lista es manejada por **Google** y ampliamente utilizada por navegadores como **Chrome**, **Firefox**, **Opera**, **Safari**, **Internet Explorer** y **Edge**.

A través de hstspreload.org, es posible visualizar el estado de las **HSTS Preload List**, así como también si esta es soportada por el navegador.

Este proyecto es también útil para verificar si un servidor web cuenta con **HSTS** y de ser así, si este se encuentra en la lista **Preload**.



Para realizar una solicitud para agregar un dominio a la **Preload List** es necesario agregar un **header HSTS válido** al servidor web, por ejemplo:

```
Strict-Transport-Security: max-age=63072000; includeSubDomains;
preload
```

A parte de esto, el **sitio web** debe de cumplir unos **requerimientos específicos** detallados en el apartado **Submission Requirements**. Una vez se cumplen estos

requisitos, es posible verificar el estado de la solicitud dentro de la misma página o visitando **chrome://net-internals/#hsts** en el navegador.

- Otra forma de comprobar si el servidor web cuenta con **HSTS**, es utilizando el comando **curl**, de la siguiente manera:

```
curl -I https://el-dominio-aqui/
```

```
$ curl -I https://www.cyberciti.biz/
HTTP/2 200
date: Mon, 01 Jun 2020 12:12:17 GMT
content-type: text/html; charset=UTF-8
set-cookie: __cfduid=d20036997cba7a61b3e3e71999f3419511591013537;
expires=Wed, 01-Jul-20 12:12:17 GMT; path=/; domain=.cyberciti.biz;
HttpOnly; SameSite=Lax; Secure
strict-transport-security: max-age=15552000
x-whome: l-ncbz02
cf-cache-status: HIT
age: 139685
cf-request-id: 03116510a3000074fb87397200000001
expect-ct: max-age=604800,
report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-
ct"
server: cloudflare
cf-ray: 59c8d7943bba74fb-EZE
alt-svc: h3-27=":443"; ma=86400
```

- Verificar que exista la cabecera **Strict-Transport-Security** entre las directivas correspondientes.

IMPORTANTE:

Recuerde que antes de solicitar la inscripción de su dominio a la **Preload List** debe tener un certificado válido y correcto en su dominio y cada uno de sus subdominios.

El protocolo HSTS (HTTP Strict Transport Security) es un mecanismo que fuerza una conexión web a través de un canal HTTPS seguro. En otras palabras: sin un certificado SSL válido, su sitio web no se cargará en su navegador, ni siquiera mostrará la opción de ignorar la advertencia SSL.

Referencias

- https://www.chromium.org/hsts
- https://developer.mozilla.org/es/docs/Web/HTTP/Headers/Strict-Transport-Security
- https://www.redeszone.net/2017/03/05/comprobar-una-pagina-web-soporta-hsts-http-strict-transport-security/
- https://www.redeszone.net/seguridad-informatica/sslstrip/
- https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/protege-tu-proyecto-web-del-sslstrip/
- https://docs.microsoft.com/en-us/iis/get-started/whats-new-in-iis-10-version-1709/iis-10-version-1709-hsts
- https://www.cyberciti.biz/faq/lighttpd-setup-hsts-http-strict-transport-security/
- https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/hsts-como-lograr-conexiones-http-seguras/
- https://scotthelme.co.uk/hsts-preloading/
- https://hstspreload.org
- https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf