HTSH - HTTP Strict Transport Security Policy

Introducción

> De aquí en adelante, los artículos contendrán una introducción que albergará información y referencias de las tecnologías implicadas en el contexto y técnicas padre que sostendrán el peso del propio informe.

Cuando se habla de servicios en red que funcionan a través de peticiones de recursos o datos, por ejemplo las páginas web que permiten el intercambio de documentos formateados en HTML compartidos en red, estamos hablando del protocolo de transferencia de hipertexto "HTTP" en sus siglas inglesas. 1

Este protocolo tiene una utilidad bastante variada, por este motivo necesita configuraciones específicas para poder alcanzar todas esas funcionalidades que con el paso del tiempo, van incrementando poco a poco los valores de configuraciones según las necesidades del objetivo. Esta es la introducción de la tecnología de la cual ramifica la próxima documentación, en caso de necesitar más información al respecto, puedes leer este documento sobre HTTP2

La información sobre este protocolo que debemos comprender para este documento sobre HTSH, se centra en las diferencias entre conexiones inseguras "http://" y las conexiones seguras "https://". Para este objetivo, no hay nada mejor que una tabla comparativa como la que se muestra a continuación.

HTTPS		HTTP
- Los URL empiezan por “https://”		- Los URL empiezan por “http://”
- El protocolo seguro utiliza el puerto 443 por omisión		- Utiliza el puerto 80 para las comunicaciones
- Canal cifrado entre servidor y cliente (bidireccional)		- Interpretación directa del flujo de datos
- Texto cifrado		- Uso de texto plano
- Indica que la página a la que accedes es la original		- No tienes forma de saber que la página es la original

Y con esta referenciación de la tecnología padre, comenzamos con el informe de HTSH.

Una vez comprendido el concepto de uso de HTTP, nos centramos en los "headers" o cabeceras que implementa este protocolo, estas nos permiten generar configuraciones en las respuestas a peticiones y que podemos visualizar a través de diferentes métodos como por ejemplo, haciendo uso de la orden de sistema "curl" que permite trabajar con las conexiones cliente-servidor.

> CURL funciona con la librería libcurl (módulo de transferencia de datos para conexiones web) y nos permite entablar conexión utilizando diferentes protocolos populares, aunque no todos son compatibles con este módulo, trabaja con los principales que se muestran en la siguiente lista.

PROTOCOLOS COMPATIBLES CON CURL
HTTP y HTTPS
FTP y FTPS
IMAP e IMAPS
POP3 y POP3S
SMB y SMBS
SFTP
SCP
TELNET
GOPHER
LDAP y LDAPS
SMTP y SMTPS

Veamos la utilidad de todo esto de forma pragmática utilizando "curl" con su parámetro especificado para mostrar "headers" en la conexión. La URL es de mi propio blog pero podemos realizar la técnica con cualquier activo o URL pública ya que, las cabeceras e información de conexiones, son de lectura pública.

```
Papi-Shelly > curl -I https://shelldredd.github.io/
HTTP/2 200
server: GitHub.com
content-type: text/html; charset=utf-8
permissions-policy: interest-cohort=()
last-modified: Fri, 23 Jun 2023 23:05:51 GMT
access-control-allow-origin: *
strict-transport-security: max-age=31556952
etag: "6496254f-6266"
expires: Sat, 25 Nov 2023 13:24:18 GMT
cache-control: max-age=600
x-proxy-cache: MISS
x-github-request-id: 1C5A:11CAD:2BD4367:2C84F4A:6561F329
accept-ranges: bytes
date: Sat, 25 Nov 2023 13:14:18 GMT
via: 1.1 varnish
age: 0
x-served-by: cache-mad22043-MAD
x-cache: MISS
x-cache-hits: 0
x-timer: S1700918058.208322,VS0,VE164
vary: Accept-Encoding
x-fastly-request-id: afc60d11dc7a6e129ad8d2fd0c2b0146e5d7e36f
content-length: 25190
```

Como se observa en el output de la orden de sistema, obtenemos las cabeceras en la respuesta a la petición realizada con curl. Entre todas ellas, la que nos interesa comprobar es la cabecera identificada con el nombre de "strict-transport-security", que pertenece a la configuración del protocolo de conexiones seguras "HTSH" y su usabilidad permite forzar las peticiones entrantes con protocolo inseguro "http"