MEDAC
Instituto Oficial de Formación Profesional

Investigación y presentación sobre protocolos de desarrollo web

Trabajo realizado por Jesús Fernández Rodríguez.

Investigación sobre el protocolo HTTPS

1. Breve historia y evolución

HTTPS (HyperText Transfer Protocol Secure) es la versión segura del protocolo HTTP, que usamos para acceder a páginas web. Fue creado por Tim Berners-Lee en 1991 para permitir la transferencia de información en la web.

- 1994: Netscape introdujo HTTPS combinando HTTP con SSL (Secure Sockets Layer), lo que permitió cifrar los datos entre el navegador y el servidor.

- Década de 2000: SSL fue reemplazado por TLS (Transport Layer Security), que ofrecía mayor seguridad y fiabilidad.

- 2014-2020: Grandes empresas como Google promovieron HTTPS como estándar, incentivando el uso de certificados gratuitos y obligando a muchos sitios web a ofrecer conexiones seguras.

Hoy en día, HTTPS es esencial para proteger la privacidad de los usuarios y garantizar que los sitios web sean auténticos.

2. Funcionamiento y usos

HTTPS protege la información transmitida entre el navegador y la web usando TLS para cifrar los datos, de modo que nadie pueda leerlos mientras viajan por internet.

Proceso básico:

1- El navegador solicita una conexión segura al sitio web.
2- El servidor envía un certificado digital que confirma su identidad.
3- El navegador verifica la validez del certificado con autoridades reconocidas.
4- Se genera una clave secreta compartida que cifra toda la comunicación.

Usos en desarrollo web:

1- Mantener seguros los datos ingresados en formularios, como registro, inicio de sesión y pagos en línea.
2- Proteger la información que intercambian APIs y servicios web.
3- Dar confianza a los usuarios y mejorar la posición del sitio en buscadores.

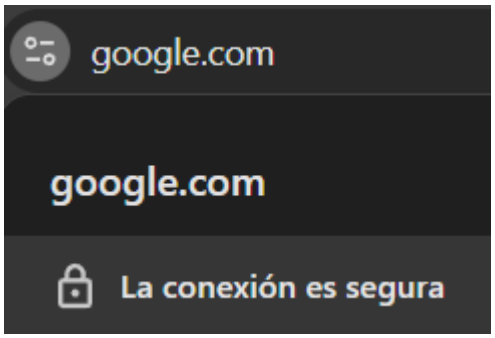
3. Ventajas y riesgos

Ventajas	Riesgos
Los datos están cifrados y protegidos	Certificados caducados o configurados incorrectamente
Confirma que el sitio es legítimo	Si no se verifica el certificado, puede haber ataques de intermediarios
Evita la manipulación de la información durante la transmisión	TLS antiguo puede ser vulnerable
Incrementa la confianza del usuario	Sitios maliciosos pueden usar certificados falsos

4. Ejemplos prácticos

a) Navegación segura:

Al acceder a un sitio como **https://www.google.com**, el navegador muestra un candado junto a la URL, indicando que la conexión es segura.



b) Comprobación rápida con terminal (curl):

```
C:\Users\jesus>curl -I https://www.google.com
HTTP/1.1 200 OK
Content-Type: text/html; charset=ISO-8859-1
Content-Security-Policy-Report-Only: object-sr
-dynamic' 'report-sample' 'unsafe-eval' 'unsaf
-hp
```

Esto indica que la página responde correctamente usando HTTPS.

c) Redirección de HTTP a HTTPS

Si escribimos **http://www.ejemplo.com**, la página se redirige automáticamente a **https://www.ejemplo.com**, asegurando que la comunicación sea segura.

d) Desarrollo local con HTTPS

En un entorno de prueba se puede crear un certificado local y acceder a **https://localhost**, permitiendo probar formularios y APIs de forma segura antes de publicar la web.

5.Bibliografía

https://www.cloudflare.com/es-es/learning/ssl/what-is-https/

https://www.one.com/es/seguridad-de-su-web/que-es-https

https://www.akamai.com/es/glossary/what-is-https

https://ed.team/blog/que-es-y-para-que-se-usa-https-explicacion-sencilla

https://www.azion.com/es/learning/websec/que-es-https/