Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Seguridad web

IIC2513 - Tecnologías y Aplicaciones Web

Sebastián Vicencio R.
2do Semestre 2020

Vulnerabilidades en la Web

¿Por qué existen?

Vulnerabilidades en la Web

¿Por qué existen?

- La Web es una plataforma abierta, cualquiera puede acceder
- Especificaciones de la Web son públicas
- Sitios web implementados de forma “básica”
- Una puerta de entrada “atractiva” para hackers

Seguridad web

¿Qué podemos hacer frente a vulnerabilidades?

Seguridad web

Definición

“Act/practice of protecting websites from unauthorized access, use, modification, destruction, or disruption."

Fuente: MDN

Ataques más conocidos

Un primer acercamiento

Cross-site scripting (XSS)

Inyección de código client-side

Código client-side que se inyecta “desde el servidor”, por lo que debería ser confiable (pero no lo es)

Hay dos tipos:

- **Reflejado**
- **Persistente**

Cross-site scripting (XSS)

XSS reflejado

Un form con un input cuyo valor se muestra en una página de resultados

El valor puede ser un script que se ejecuta automáticamente

http://mysite.com?q=<script>alert("Hello")</script>

Cross-site scripting (XSS)

XSS persistente

Un script es persistido en base de datos y luego mostrado a todo usuario que acceda esa página

Ejemplo: comentario que tenga como contenido un script

```
<p>
  Content
  <script>alert('Hello')</script>
</p>
```

SQL Injection

Ejecución de código SQL en una base de datos a través de algún input que viene desde el browser

```
<input type="text" name="name" />
```

```
"SELECT * FROM users WHERE name = '" + name + "';"
```

```
name = “Pepito”  ⟶  SELECT * FROM users WHERE name = 'Pepito';
```

SQL Injection

Ejecución de código SQL en una base de datos a través de algún input que viene desde el browser

```
<input type="text" name="name" />
```

```
"SELECT * FROM users WHERE name = '" + name + "';"
```

```
name = "a';DROP TABLE
users; SELECT * FROM
userinfo WHERE 't' = 't"
```

```
SELECT * FROM users WHERE name =
'a';DROP TABLE users; SELECT * FROM
userinfo WHERE 't' = 't';
```

Cross-site Request Forgery (CSRF)

Permite a un atacante ejecutar acciones utilizando las credenciales de un usuario

Un form desde otra página al sitio que se quiere atacar

Si el usuario esta logueado, cookies se agregan automáticamente por el browser

Cross-site Request Forgery (CSRF)

Link engañoso que en realidad es un form

```
<form action="bank_site/transaction" method="POST">
  <input type="hidden" name="account_id" value="123456" />
  <input type="hidden" name="amount" value="10000000" />
  <input type="submit" class="link-style" value="¡Gana dinero fácil!" />
</form>
```

Denial of Service (DoS)

Ejecutar operaciones costosas en un servidor hasta que deje de responder

Hay dos tipos:

- **Sobrecarga de recursos**
- **Sobrecarga de conexiones**

Man in the middle

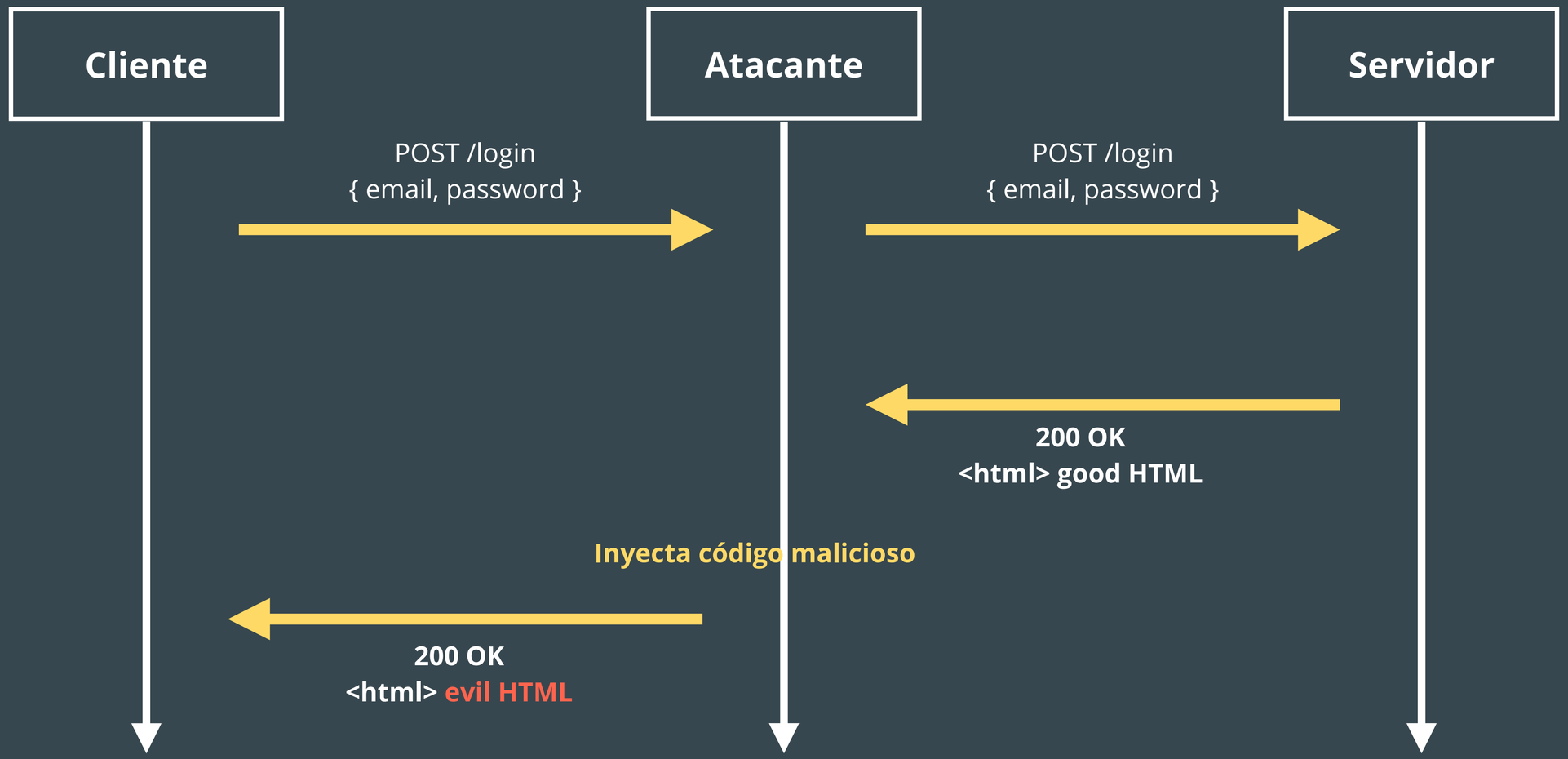
Conexiones interceptadas por un tercero

- Sólo “mirar” o también robar información
- Suele darse en redes wifi públicas

HTTPS

Hypertext Transfer Protocol Secure

HTTP no es seguro



HTTPS

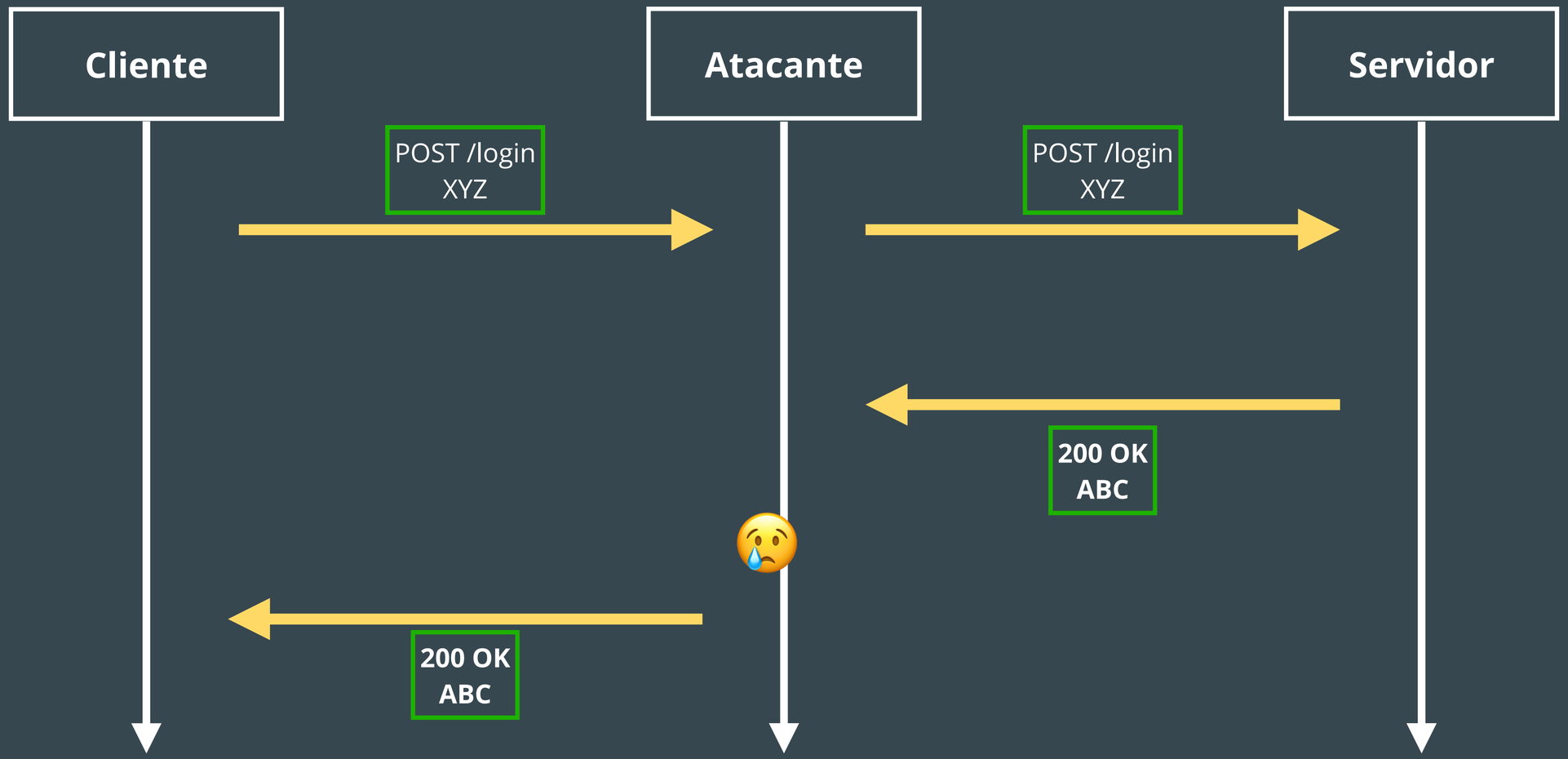
¿Qué es?

Versión segura de HTTP

HTTP sobre TLS

Comunicación encriptada entre cliente y servidor

HTTPS es seguro



TLS

Transport Layer Security

Protocolo de intercambio de información seguro entre dos partes

Provee 3 cosas:

- **Autenticación**
- **Encriptación**
- **Integridad**

Public-key cryptography

Encriptación asimétrica

Utiliza un par de llaves: private (secreta) y public (compartida con el mundo)

Mensajes encriptados con private key sólo pueden ser desencriptados con public key (y viceversa)

¿Cómo asegurar que public key corresponde a servidor?

Certificado TLS (SSL)

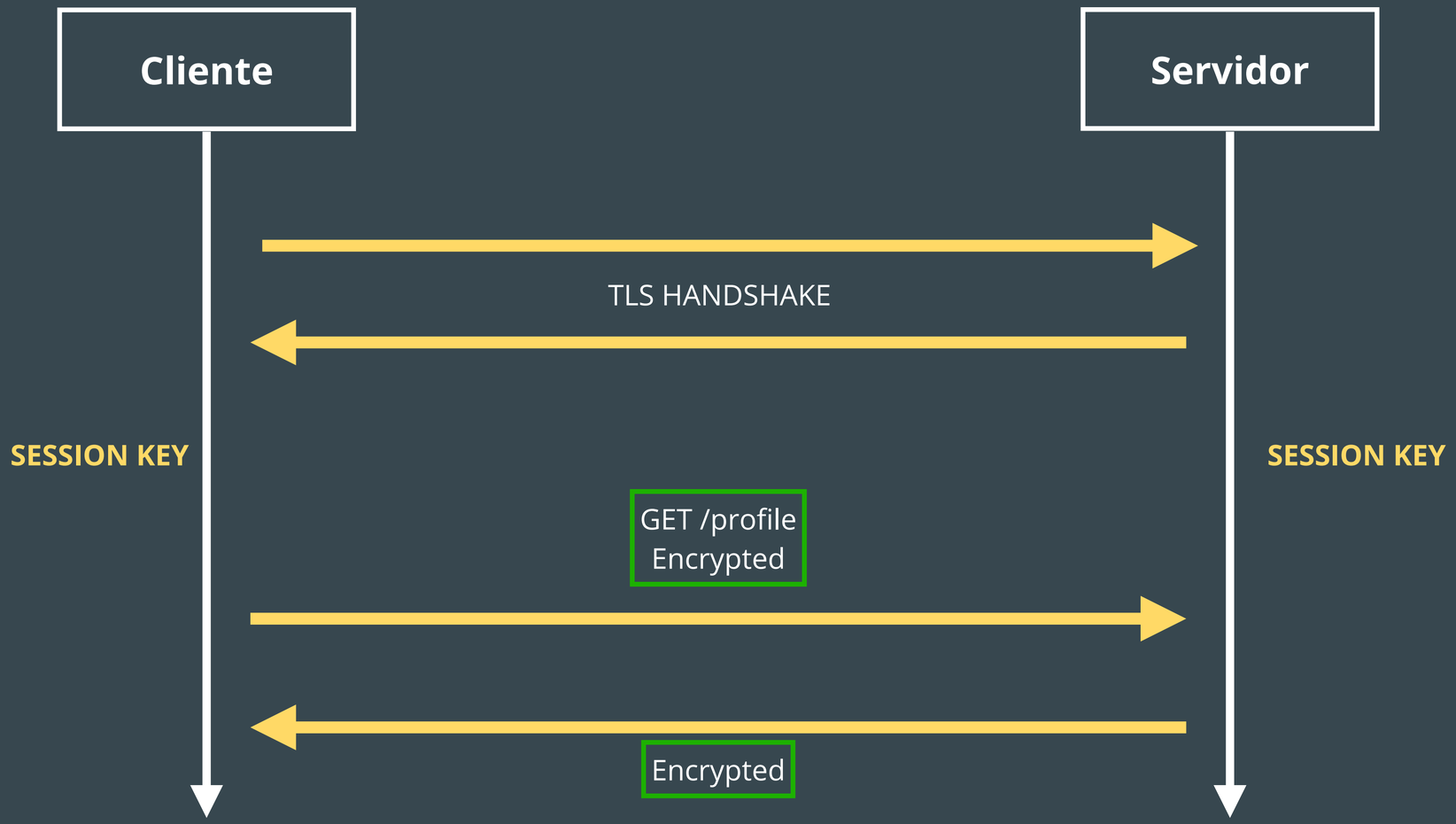
Servidor debe obtener un certificado (archivo) y guardarlo

Entidad que entrega certificados: Certificate Authority

Entidad certifica que X dominio es dueño de Y public key

TLS

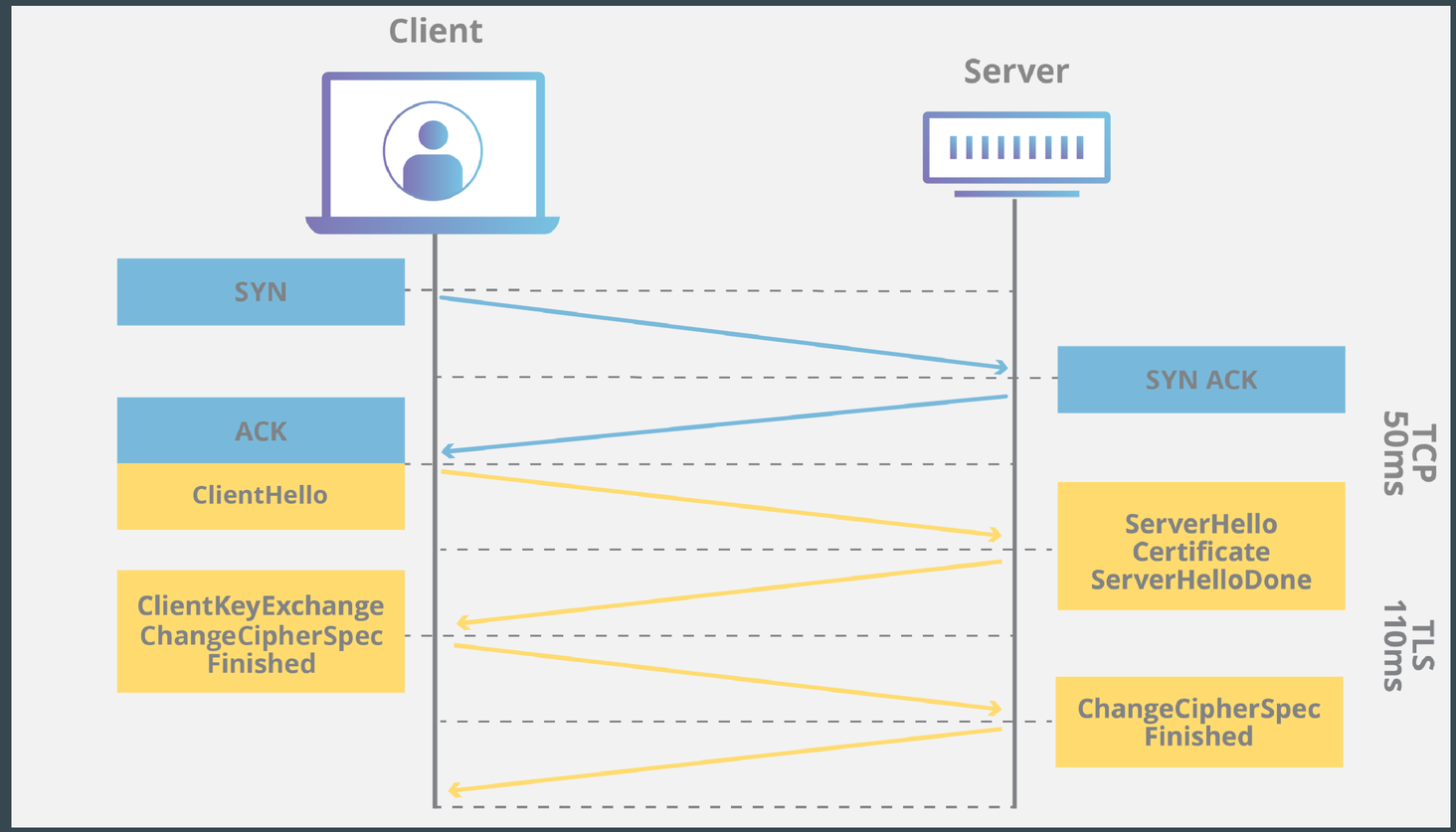
Flujo de comunicación



TLS

Flujo de comunicación

TLS handshake



Fuente: Cloudflare

Ejemplo

ing.puc.cl

Beneficios HTTPS

¿Por qué deberíamos usarlo?

- Imagen de confianza hacia usuarios
- Browsers modernos destacan sitios sin HTTPS como no seguros
- HTTPS es más seguro, tanto para clientes como servidores
- Permite asegurar que un servidor es quien dice ser
- Capacidad de utilizar features modernos de browsers (geolocalización, push notifications)

HTTPS en la práctica

¿Cómo agregarlo a mi sitio web?

Cloudflare

Certificados TLS gratuitos

Let's Encrypt

Certificados TLS gratuitos

HTTPS is Easy!

Guía para incluir HTTPS

Recomendaciones finales seguridad

- **Nunca confiar en los datos que vienen desde un browser**
- **Siempre revisar y sanitizar datos de input**
- **Ponerse en el peor caso**

CORS

Cross-Origin Resource Sharing

¡Próxima clase!

Referencias

- MDN - "Website security"
- MDN - "Web security"
- Cloudflare - "What Is Web Application Security?"
- MDN - "Transport Layer Security"
- Cloudflare - "What is SSL? | SSL definition"
- Stanford - "CS 253 Web Security"
- High Performance Browser Networking - "Transport Layer Security (TLS)"