Unidad 2

Servidor web seguro HTTPS

SERVIDOR WEB SEGURO

El servidor web decimos que es seguro cuando garantiza la comunicación con el cliente web con **autenticación** y **confidencialidad**.

SERVIDOR WEB SEGURO

HTTPS (Protocolo Seguro de Transferencia Hipertexto) funciona desde el puerto 443 y utiliza un cifrado basado en SSL/TLS con el fin de crear un canal cifrado entre el cliente y el servidor. SSL (Secure Sockets Layer) y TLS (Transmission Layer Security) son dos protocolos utilizados para enviar paquetes cifrados a través de Internet. Se pueden utilizar para más de un protocolo, no sólo con HTTP. HTTP + SSL/TLS = HTTPS.

SERVIDOR WEB SEGURO

¿Cómo funciona?

HTTP funciona en la capa de aplicación (séptima capa) del Modelo OSI, que es la capa más alta. Sin embargo, el cifrado que da lugar a HTTPS, se realiza en una capa más baja, mediante SSL/TLS.

HTTPS se basa en el sistema de clave pública y clave privada. El administrador de un servidor debe crear un certificado de clave pública, el cual debe estar firmado por una autoridad de certificación. Si no está firmado, el navegador Web no lo aceptará y nos dirá que el sitio no es seguro. Si el sitio es seguro, veremos un candado cerrado en la URL del navegador

SERVIDOR WEB SEGURO

Intercambio de claves

La seguridad de las transacciones se basa en el intercambio de claves entre un cliente y un servidor. Las claves pública y privada están relacionadas de tal forma que no podemos usar una sin la otra. Para enviar un mensaje cifrado a un servidor, lo cifro con su clave pública para que él pueda descifrarlo con su clave privada. Sólo quien tenga la privada podrá descifrarlo. Por eso es importante tenerla siempre a buen recaudo.

Tras haber concretado los detalles técnicos que se utilizarán en la transferencia, como por ejemplo la versión del protocolo o los algoritmos de cifrado, el navegador procede a cifrar una clave generada en ese mismo momento con la clave pública del servidor al que se está conectando y se la envía. Al final el cliente y el servidor tienen la misma clave, que se utilizará para cifrar y descifrar los datos de la comunicación.

Unidad 2

PROTOCOLO HTTPS

Se basa en dos tipos de criptografía:

- Criptografía simétrica o de clave compartida.
  - Confidencialidad: cifra la información transmitida.
  - VENTAJA: Es rápido para el cifrado.
- Criptografía asimétrica de clave pública/privada.
  - Confidencialidad
  - Autenticación del servidor con la que se establece la conexión. Certificado del servidor.
  - INCONVENIENTE: Es lento para el cifrado.

PROTOCOLO HTTPS

DEBILIDADES DE HTTPS

- Custodia de las claves privadas de un CA. Un atacante podría crear y firmar certificados válidos para cualquier dominio sin que nadie lo impidiese y por lo tanto engañar a los usuarios que se conectasen a servidores falsos.

GENERAR CERTIFICADOS

1.Generar la **clave privada**

2.Crea tu C**ertificate Signing Request (CSR)** o **solicitud de certificado** es la *clave pública* (garantizan la integridad de los datos.) y algunos *datos personales* (autenticación) y garantizan la integridad de los datos.

3.Esta solicitud se podría enviar a una autoridad de certificación para generar el certificado (**CRT**).

GENERAR UN CSR PARA APACHE CON OPENSSL

1.Generar la clave privada

```
openssl genrsa 2048 > fichero.key
```

1.Genera la solicitud de certificado (CSR-PKCS#10)

```
openssl req -new -key fichero.key > fichero.csr
```

Unidad 2

INFORMACIÓN SOLICITA POR EL SISTEMA (CSR)

.**Country Name (2 letter code) []**: (SP in Spain for example)

.**State or Province Name (full name) [Some-State]:** (your state or province name, name of your departament in Spain)

.**Locality Name (eg, city) []**: (the name of your city)

.**Organization Name (eg, company) []**: (your organization name) Preferente en mayúsculas.

.**Organizational Unit Name (eg, section) []**: (do not fill - advised - or enter a generic term such as "IT Department".)

.**Common Name (eg, YOUR name) []**: (domain name/ server name/ FQDN is the name of the website to be secured) No admite espacios en blanco.

.**Email Address []**: (let blank)

FIRMAR EL CERTIFICADO DIGITAL

Crea el certificado digital autofirmado usando la clave privada

```
openssl x509 -req -days 365 -in fichero.csr -signkey
fichero.key > fichero.crt
```

Unidad 2

ACTIVAR LA SEGURIDAD DE APACHE

MÓDULO SSL DE APACHE: ACTIVACIÓN

- El método de cifrado SSL/TLS utiliza un método de cifrado de clave pública (cifrado asimétrico) para la autenticación del servidor.
- El **módulo ssl** es quien permite cifrar la información entre navegador y servidor web.
- Este módulo proporciona SSL v2/v3 y TLS v1 para el Servidor Apache HTTP; y se basa OpenSSL para porporcionar el modo de criptografía.

INSTALACIÓN CERTIFICADO EN EL SITIO POR DEFECTO

1. Mover la clave privada(fichero.key) al directorio `/etc/ssl/private`
2. Propietario de dicho fichero es `root:ssl-cert` y permisos 640
3. Mover el certificado (fichero.crt) al directorio `/etc/ssl/certs`

Propietario de dicho fichero es root:root

CONFIGURACIÓN DEL SITIO VIRTUAL SEGURO

Copia el archivo default-ssl.conf en un archivo nuevo. Ejemplo: empresa1-ssl.conf

```
SSLEngine on

SSLCertificateFile /etc/ssl/certs/___.crt

SSLCertificateKeyFile
/etc/ssl/private/__.key
```

Unidad 2

CONFIGURACIÓN DEL HOST SITIO VIRTUAL SEGURO

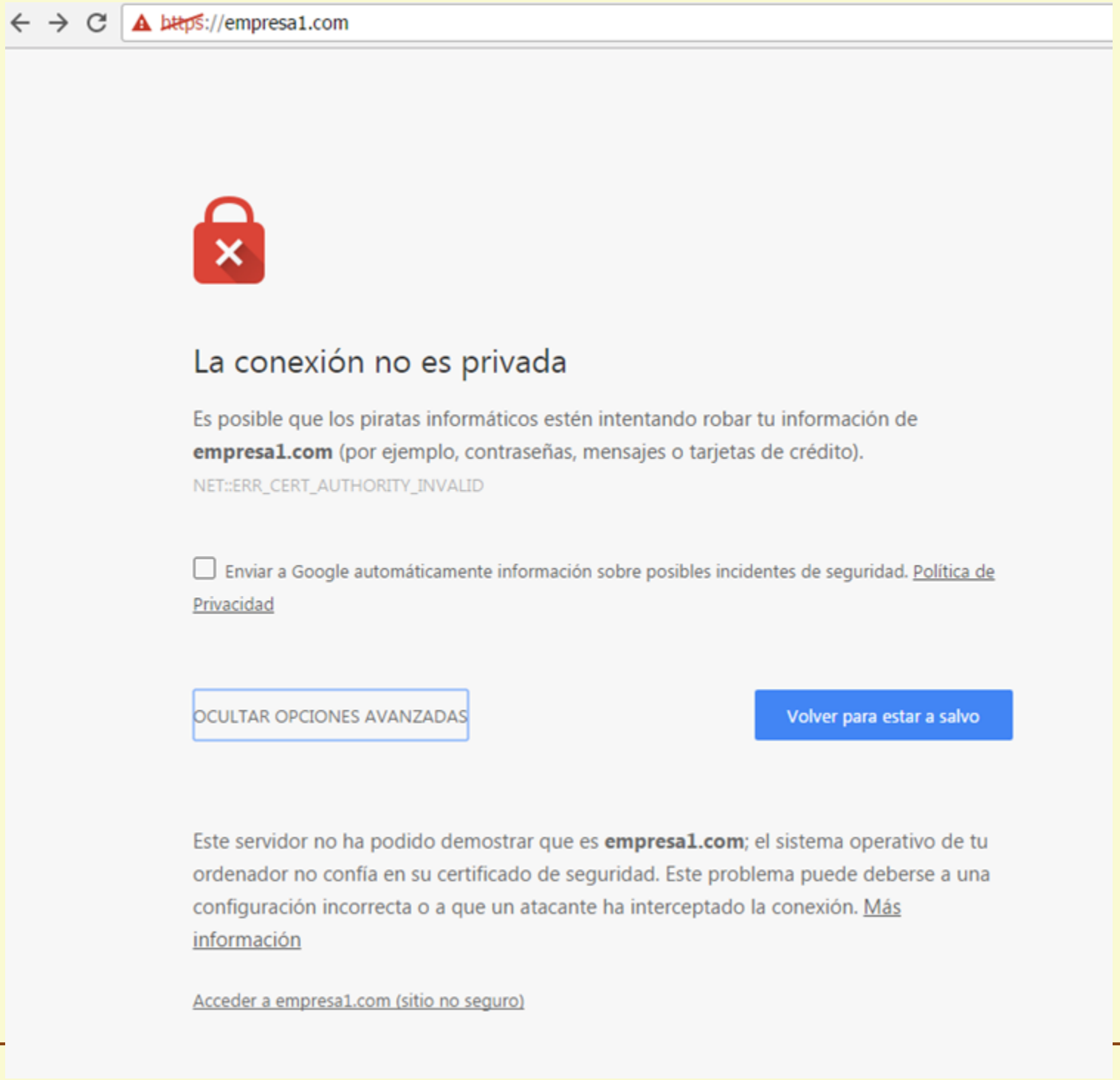
```
<IfModule mod_ssl.c>
    <VirtualHost *:443>
        ServerAdmin w_____
        ServerName _____
        ServerAlias _____
        DocumentRoot _____
....    ErrorLog ${APACHE_LOG_DIR}/.....-ssl-error.log
        CustomLog ${APACHE_LOG_DIR}/.....-ssl-access.log combined
.....
```

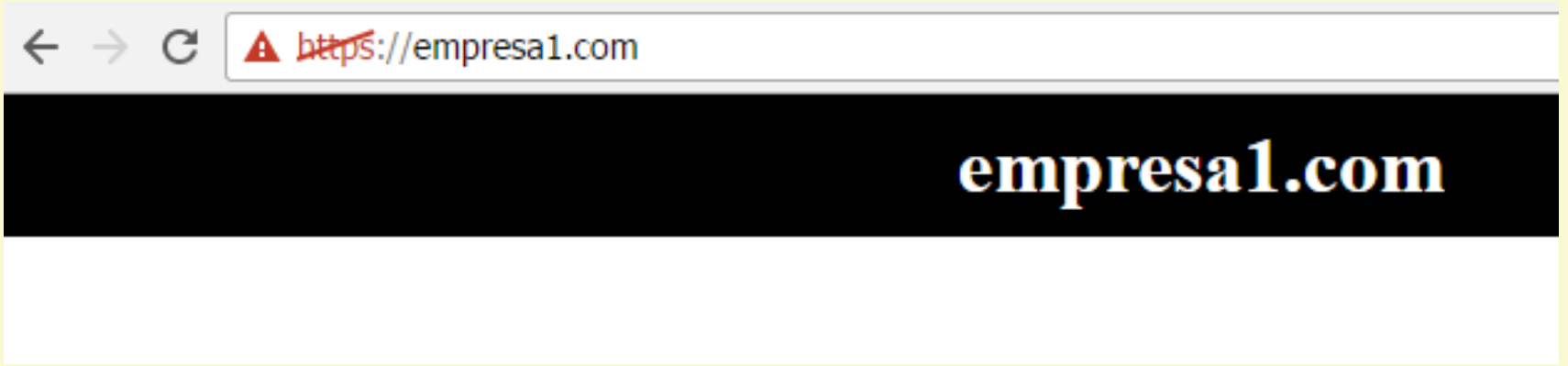
Sustituir _default_ :443 por *:443

ACTIVACIÓN DEL SITIO VIRTUAL

```
sudo a2ensite fichero.conf
```

- Reiniciar el servicio para que los cambios tengan efecto
- Comprobación desde el navegador estableciendo una conexión segura. https://·…


https://empresa1.com
La conexión no es privada
Es posible que los piratas informáticos estén intentando robar tu información de empresa1.com (por ejemplo, contraseñas, mensajes o tarjetas de crédito).
NET::ERR_CERT_AUTHORITY_INVALID
Enviar a Google automáticamente información sobre posibles incidentes de seguridad. Política de Privacidad
OCULTAR OPCIONES AVANZADAS
Volver para estar a salvo
Este servidor no ha podido demostrar que es empresa1.com; el sistema operativo de tu ordenador no confía en su certificado de seguridad. Este problema puede deberse a una configuración incorrecta o a que un atacante ha interceptado la conexión. Más información
Acceder a empresa1.com (sitio no seguro)


https://empresa1.com
empresa1.com

Unidad 2

REDIRECCIONAR HTTP A HTTPS

En .htaccess de ejemplo,

RewriteEngine On

RewriteCond %{SERVER_PORT} 80

RewriteRule ^(.*)$ https://www.dominio.com/$1 [R,L]

Simplemente, visite http://dominio.com en su navegador, y debería ver que se muestra la misma página, pero la dirección ha cambiado a https://www.dominio.com.

Explicación del código

Options +FollowSymLinks - es una directiva de Apache, requisito previo para mod_rewrite.

RewriteEngine On - habilita mod_rewrite.

RewriteCond %{SERVER_PORT} 80 - sirve para indicar que todas las peticiones que se realicen al puerto 80 (puerto por defecto de Apache para servicio web), deseamos que vayan a través de la regla especificada.

RewriteRule - define una regla particular.

Dentro de la regla de reescritura, la primera cadena de caracteres después de RewriteRule, define lo que la URL original parece. La segunda cadena después de RewriteRule define la nueva URL.

$1 - Este carácter especial, sustituye (o indica) la parte entre paréntesis, especificada en la primera cadena. Básicamente, lo que haces es asegurar que las sub-páginas redireccionan a la misma sub-página y no a la página principal. Puede omitirlo para redirigir a la página principal. (Si usted no tiene el mismo contenido en el nuevo directorio que había en el antiguo directorio, deje esta expresión regular

[R,L] - Esta opción, realiza una redirección, y también deshabilita que las reglas de reescritura que estén escritas después afecten a la dirección URL (una buena idea para añadir después de la última rewrite rule).