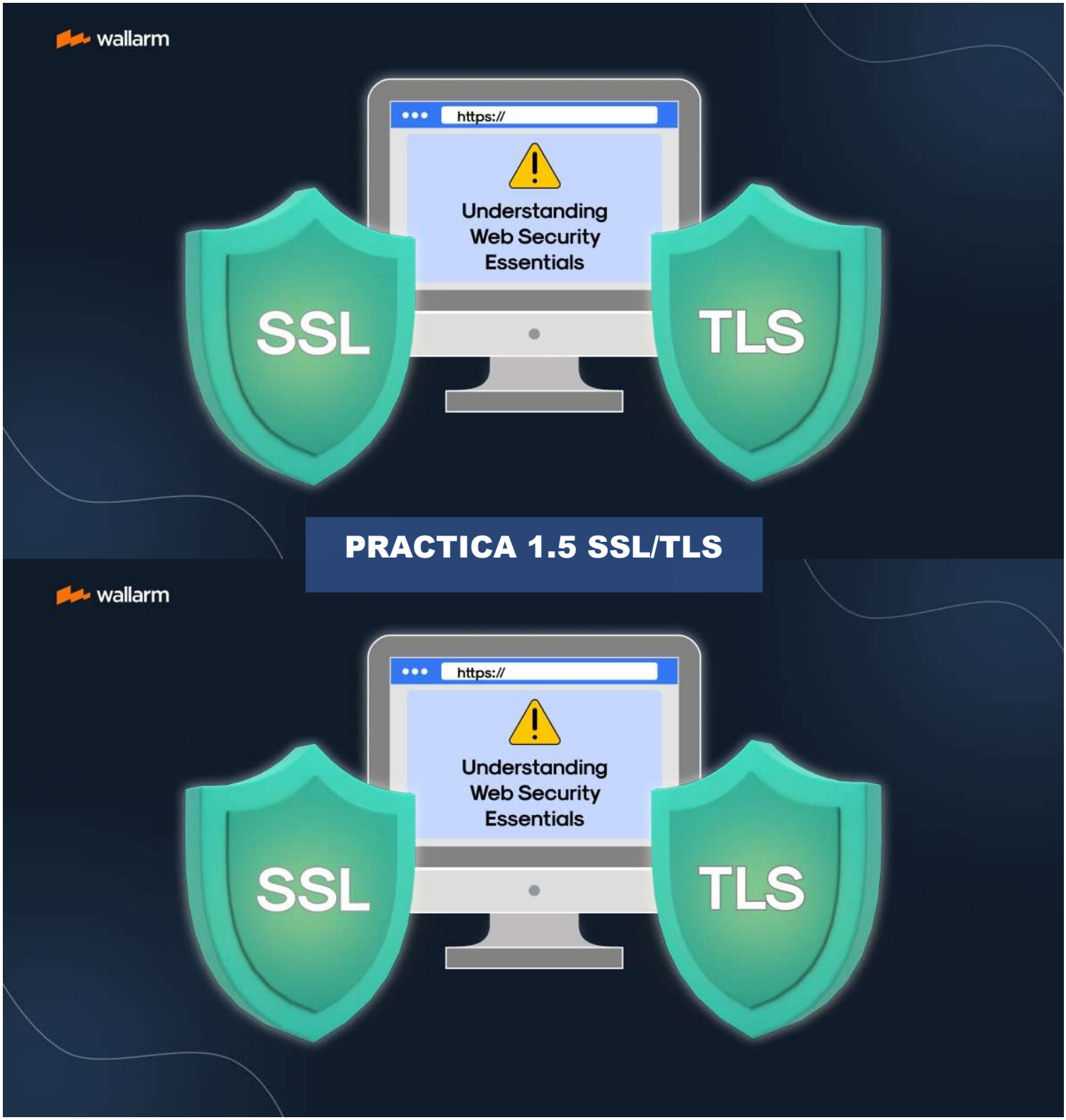


PRACTICA 1.5 SSL/TLS

Hecho por Izan Navarro

ÍNDICE

1.Introducción

En el contexto de la seguridad informática, garantizar la confidencialidad, integridad y autenticidad de los datos en tránsito es fundamental para proteger la comunicación entre usuarios y sistemas. Para lograr este objetivo en redes como Internet, se han desarrollado protocolos de cifrado, siendo SSL (Secure Sockets Layer) y TLS (Transport Layer Security) los más reconocidos y utilizados.

SSL, creado en la década de 1990, fue el primer protocolo ampliamente aceptado para establecer conexiones seguras entre clientes y servidores. Sin embargo, a lo largo del tiempo, se descubrieron varias vulnerabilidades en sus versiones, lo que llevó a la evolución de SSL en el protocolo TLS. TLS mantiene el mismo objetivo de SSL, pero ofrece mayor seguridad y rendimiento, incorporando algoritmos de cifrado más robustos y mecanismos de autenticación mejorados.

En esta práctica se explorarán los principios fundamentales de SSL y TLS, sus diferencias técnicas, y cómo implementar una conexión segura utilizando estos protocolos. Además, se abordará la estructura y funcionamiento del proceso de negociación (handshake), el papel de los certificados digitales, y los tipos de algoritmos de cifrado utilizados. Al concluir la práctica, los estudiantes comprenderán cómo configurar e inspeccionar conexiones seguras y estarán mejor preparados para identificar y mitigar posibles riesgos de seguridad en redes de comunicación.

Este conocimiento es esencial no solo para los profesionales de seguridad, sino también para cualquier persona que participe en el desarrollo y administración de sistemas y redes en un mundo cada vez más interconectado.

Requisitos

Para esta práctica se necesitará:

- Un dispositivo con tu sistema operativo habitual.
- Una máquina virtual, preferiblemente Linux.

2. Diferencias entre SSL/TLS

· Comenta con tus palabras que es SSL y TLS, sus características, su funcionamiento, así como sus diferencias.

➔ SSL: Significa Secure Sockets Layer, es uno de los primeros protocolos creados para proteger la información que se transfiere entre un cliente y un servidor. Para todo esto su principal función era **cifrar los datos**, para que en caso de interceptarlos, nadie pueda entenderlos. Todo esto vino previamente a TLS.

➔ TLS: significa Transport Layer Security, es la versión mejorada y más segura de SSL. Esto hace lo mismo, cifrar y proteger la conexión, pero con algoritmos más modernos y fuertes. Un ejemplo sería el " https:// " de las páginas web, esto es TLS.

Características principales

- **Cifrado**: los datos se vuelven ilegibles para cualquiera que los intercepte.
- **Autenticación**: el servidor (y a veces el cliente) se verifica mediante certificados digitales.
- **Integridad**: asegura que los datos no se alteren durante el envío.
- **Confidencialidad**: solo el emisor y el receptor pueden leer la información.

Para concluir:

SSL fue el "abuelo" de las conexiones seguras, pero TLS lo reemplazó porque es más moderno y confiable. Gracias a TLS podemos navegar por internet, comprar cosas o entrar a nuestras cuentas sin que alguien pueda ver lo que estamos haciendo.

3. OpenSSL

• Comenta con tus palabras, que es y que nos puede ofrecer la herramienta OpenSSL.

OpenSSL es una herramienta de software de código abierto que sirve para gestionar la seguridad en las comunicaciones digitales. Se basa en protocolos SSL y TLS para las transmisiones por internet.

Esta nos permite:

- Cifrar datos

- Verificar la identidad de servidores y usuarios

- Firmas digitales

- Pruebas de seguridad de conexiones

• Comenta con tus palabras que es el cifrado simétrico y asimétrico.

El cifrado simétrico usa la misma clave para cifrar y descifrar la información. Es rápido y eficiente, pero compartir la clave de forma segura es un trabajo complejo, ya que si se obtiene, también se obtiene acceso completo a la información.

El cifrado asimétrico usa 2 claves diferentes pero relacionadas matemáticamente:

· Una clave pública

· Una clave privada

Esto permite comunicaciones seguras sin necesidad de compartir una clave secreta.

• Comenta con tus palabras todo lo que sepas de los ataques POODLE y DROWN.

· POODLE o Padding Oracle On Downgraded Legacy Encryption es un ataque que aprovecha una debilidad del protocolo SSL 3.0 que se fija en el padding en modos CBC, pudiendo observar una bajada de protocolo y haciendo de “man-in-the-middle”. Esto permite descifrar datos sensibles.

· DROWN o Decrypting RSA with Obsolete and Weakened Encryption es un ataque “cross-protocol” que permite a un atacante aprovechar soporte de SSLv2 en algún servidor para poder romper la seguridad de conexiones TLS modernas. Debe existir soporte SSLv2 activo en el mismo servidor.

• Codifica y cifra con cifrado simétrico un fichero de texto que contenga tu nombre completo, detalla todos los pasos que has seguido, explicando dichos pasos con tus palabras y adjuntando las evidencias oportunas que consideres para demostrarlo.

```
izan@Izan-Ubuntu20:~/Escritorio$ echo "Izan Navarro Lujan" > izan.txt
izan@Izan-Ubuntu20:~/Escritorio$ openssl enc -aes-256-cbc -salt -in izan.txt -o
ut izan_cifrado.bin
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
izan@Izan-Ubuntu20:~/Escritorio$
```

1) Creamos el fichero con mi nombre.

2) Cifrarlo con AES-256-CBC (simétrico).

3) La contraseña elegida es “Izan1234” para poder desencriptar.

```
izan@Izan-Ubuntu20:~/Escritorio$ ls
izan_cifrado.bin  izan.txt
```

Aquí tenemos los ficheros creados; “izan.txt” como archivo normal y “izan_cifrado.bin” como archivo cifrado.

```
izan@Izan-Ubuntu20:~/Escritorio$ hexdump -C izan_cifrado.bin | head
00000000  53 61 6c 74 65 64 5f 5f  c1 d3 6f f0 36 85 a8 04  |Salted__..o.6...|
00000010  c4 0b a7 fc 70 54 cf ab  54 55 ae 72 81 3f 6c bb  |....pT..TU.r.?l.|
00000020  de 55 cc 87 15 25 80 4f  01 12 af a0 3e de 42 c5  |.U...%.O....>.B.|
00000030
izan@Izan-Ubuntu20:~/Escritorio$
```

Así comprobamos que el archivo está cifrado, sino, saldría algo así:

```
izan@Izan-Ubuntu20:~/Escritorio$ hexdump -C izan.txt | head
00000000  49 7a 61 6e 20 4e 61 76  61 72 72 6f 20 4c 75 6a  |Izan Navarro Luj|
00000010  61 6e 0a                                          |an.|
00000013
```

• Codifica y cifra con cifrado asimétrico, mediante el sistema de claves públicas y privadas, un fichero de texto con tu nombre completo. Detalla todos los pasos que has seguido, explicando los pasos con tus palabras y adjuntando evidencias.

```
izan@Izan-Ubuntu20:~/Escritorio$ openssl genpkey -algorithm RSA -out clave_priv
ada.pem -pkeyopt rsa_keygen_bits:2048
.....+++++
.....+++++
izan@Izan-Ubuntu20:~/Escritorio$ openssl rsa -pub
-pubin    -pubout
izan@Izan-Ubuntu20:~/Escritorio$ openssl rsa -pubout -in clave_privada.pem -out
 clave_publica.pem
writing RSA key
izan@Izan-Ubuntu20:~/Escritorio$
```

Generamos las claves tanto públicas como privadas con RSA (cifrado asimétrico).

```
izan@Izan-Ubuntu20:~/Escritorio$ openssl rsautl -encrypt  -inkey clave_publica.
pem -pubin -in izan.txt -out izan_rsa_cifrado.bin
izan@Izan-Ubuntu20:~/Escritorio$ ls
clave_privada.pem  izan_cifrado.bin      izan.txt
clave_publica.pem  izan_rsa_cifrado.bin
```

Encriptamos el archivo “izan.txt” gracias a la clave pública y lo renombramos como “izan_rsa_cifrado.bin”. Acto seguido muestro todos los archivos existentes.

```
izan@Izan-Ubuntu20:~/Escritorio$ hexdump -C izan_rsa_cifrado.bin | head
00000000  53 72 3c cc f5 04 12 82  7a f0 5a ef 4b d2 83 22  |Sr<.....z.Z.K..”|
00000010  96 3c 39 a0 da 58 5c 6c  c2 b6 4c 10 97 26 4d c4  |.<9..X\l..L..&M.|
00000020  3c 97 e1 a1 b9 8f 45 57  fe db eb e5 13 39 83 33  |<.....EW.....9.3|
00000030  e6 0b aa d9 a0 35 2e 44  50 30 06 63 4d 99 a7 ec  |.....5.DP0.cM...|
00000040  a7 d9 16 ef 13 6f 36 bc  f0 8a 03 93 74 3d 39 7e  |.....o6.....t=9~|
00000050  87 6c 94 b8 db db ac e0  56 41 70 71 d4 82 10 c6  |.l......VApq....|
00000060  f7 4d 40 16 b3 ee a0 0b  7f 27 18 c2 8d 7a 91 1e  |.M@......'...z..|
00000070  29 56 ce 1d c6 44 36 2f  12 35 71 85 4a b2 bf d2  |)V...D6/.5q.J...|
00000080  3a 41 b3 72 19 28 55 da  68 eb b4 9a 5f 89 d4 b2  |:A.r.(U.h..._...|
00000090  d7 b9 53 15 d6 59 51 a8  4b 4f 1c 27 b5 0e 95 57  |..S..YQ.KO.'...W|
izan@Izan-Ubuntu20:~/Escritorio$
```

Comprobamos que el archivo se ha encriptado correctamente.

4. Let’s Encrypt

· Comenta con tus palabras todo lo que puedas averiguar sobre Let’s Encrypt y los servicios que ofrece.

Let’s Encrypt es una **autoridad certificadora (CA)** gratuita, automatizada y abierta, creada en 2015 por la organización sin ánimo de lucro **Internet Security Research Group (ISRG)**.
Su objetivo es **facilitar el uso del cifrado HTTPS** en todos los sitios web, promoviendo una web más segura y privada.

· Servicios que ofrece:

1. **Certificados digitales gratuitos**
    - Permite obtener certificados SSL/TLS sin coste.
    - Valida la identidad del dominio mediante **validación de dominio (DV)**.

2. **Renovación automática**
    - Los certificados duran 90 días, pero se pueden **renovar automáticamente** con herramientas como *Certbot*.
3. **Automatización del proceso**
    - Todo (solicitud, validación, emisión y renovación) se hace de forma automática, sin intervención manual.
4. **Compatibilidad total**
    - Los certificados son reconocidos por todos los principales navegadores y sistemas operativos.

5. Creación de un dominio gratuito, con certificado

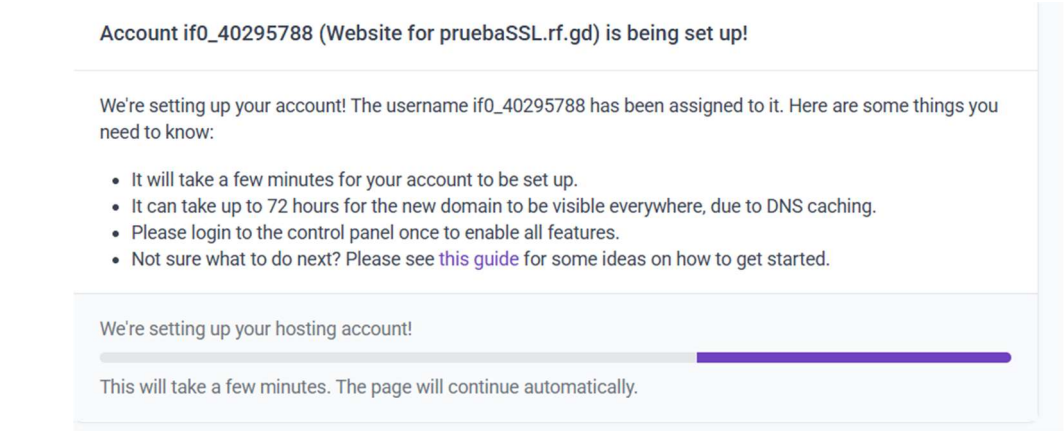
Para realizar este apartado, podéis serviros de Infinity Free, Weeebly o cualquier otra herramienta que consideréis.

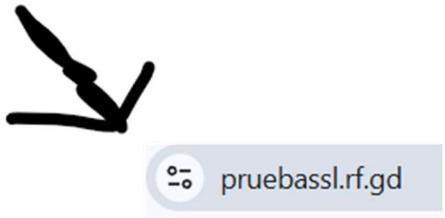
• Registra un dominio de prueba y crea un certificado, para dicho dominio publicado. Detalla el proceso y adjunta capturas.

• Una vez el dominio este publicado, muestra la información del certificado por el navegador. Donde se pueda ver para quien va destinado el certificado, quien es la CA emisora, periodo de validez, hash tanto del certificado como de la clave pública.

En este caso he creado una nueva cuenta en “Infinity Free” y he creado el dominio dándole al botón “Create Account” y creando el “pruebaSSL.rf.gd”.

Esta es la pantalla que saldrá durante la creación del dominio.





Una vez creado el dominio ponemos la url en el buscador predeterminado elegido por el sistema y entramos a **“La conexión es segura > El certificado es válido”** y nos mostrará algo como esto:

Aquí se nos muestra para quien va destinado el certificado, quien es la CA emisora, periodo de validez y hash tanto del certificado como de la clave pública

6. Documentación para entregar

• Memoria técnica detallada, con todos los puntos anteriormente mencionados y las evidencias que el alumno considere oportunas para demostrar la configuración realizada, en un formato adecuado.

• Claves generadas con su contraseña.

• Ficheros cifrados.