1. Administración de servidores web

El foco es asegurar la capa de comunicación y la capa de acceso a recursos del servidor.

1.1. El Protocolo HTTPS (HTTP + SSL/TLS)

HTTPS (Hypertext Transfer Protocol Secure) es la implementación estándar para garantizar la **seguridad** en la web. No es un protocolo distinto a HTTP, sino HTTP encapsulado dentro de una capa de seguridad: **SSL/TLS**.

Concepto	Importancia
Confidencialidad	Garantiza que la comunicación (datos, credenciales, *cookies*) entre el cliente y el servidor se mantenga **cifrada**. Si un atacante intercepta el tráfico, solo verá datos ilegibles.
Integridad	Mediante *checksums* o funciones *hash*, se asegura que los datos **no han sido modificados** durante la transmisión.
Autenticación	El cliente (navegador) puede **verificar la identidad** del servidor web, evitando ataques *Man-in-the-Middle* y *phishing*.

A. Funcionamiento Básico: El Handshake SSL/TLS

El **Handshake** es el apretón de manos inicial que establece la conexión segura, negociando los parámetros de cifrado:

1. **Negociación:** El cliente y el servidor acuerdan el protocolo y los algoritmos de cifrado que ambos soportan.
2. **Autenticación:** El servidor envía su **Certificado Digital** (que incluye su clave pública).
3. **Intercambio de Claves:** El cliente genera una clave de sesión (clave simétrica), la cifra con la clave pública del servidor, y la envía. Solo el servidor puede descifrarla con su clave privada.
4. **Cifrado Simétrico:** A partir de aquí, ambos usan la clave simétrica de sesión para cifrar y descifrar la transferencia de datos de forma rápida.

1.2. Certificados Digitales y Autoridades de Certificación (CA)

El **Certificado Digital** es la credencial que permite la autenticación en el *handshake*.

A. Componentes del Certificado

Un certificado **X.509** contiene información clave, incluyendo:

- La **clave pública** del servidor.
- El **nombre del dominio** (CN - Common Name, ej: localhost).
- La **firma digital** de la Autoridad de Certificación (CA).

B. Tecnologías de Certificación

1. **Certificado Autofirmado (*Self-Signed*):**
    - **Propósito:** Entornos de desarrollo o pruebas internas de la arquitectura (como la nuestra en el módulo).
    - **Generación:** Se usa la propia clave privada del servidor para firmar el certificado.
    - **Despliegue:** Se configura directamente en el servidor Web (ej., módulo mod_ssl de **Apache** o directivas ssl_certificate y ssl_certificate_key de **Nginx**). El navegador mostrará una advertencia de seguridad, ya que no confía en la firma.
2. **Certificado de una CA Pública:**
    - **Propósito:** Despliegues en producción.
    - **Ejemplo: Let's Encrypt**. Es una CA gratuita y automatizada.
    - **Tecnología de gestión: Certbot**. Esta herramienta automatiza la generación de claves, la solicitud del certificado (mediante el protocolo ACME) y su instalación/renovación automática en Apache o Nginx.

1.3. Autenticación y Control de Acceso (Servidores Web)

Incluso con HTTPS, necesitamos restringir el acceso a zonas sensibles, como directorios de administración o archivos de configuración.

A. Autenticación Básica HTTP (HTTP Basic Auth)

Es el método más simple y se gestiona directamente a nivel de servidor Web, sin necesidad de código de aplicación.

- **Fundamento:** El servidor solicita al cliente que envíe el usuario y la contraseña codificados en Base64 en la cabecera HTTP (de ahí la necesidad imperiosa de usar **HTTPS** para cifrar esa cabecera).

B. Implementación en la Arquitectura de la UD1 (Asunción: Apache)

Asumiremos que estamos trabajando con **Apache HTTP Server** (el más común para este tipo de prácticas).

Fichero de Configuración	Propósito	Tecnología o Comando
.htpasswd	Almacena los usuarios y sus contraseñas **cifradas** (mediante algoritmos como MD5 o SHA).	Comando htpasswd para crear el archivo y añadir usuarios.
.htaccess	Fichero de configuración que se coloca en el directorio a proteger. Contiene las directivas que *activan* la autenticación y **apuntan** al archivo .htpasswd.	Directivas AuthType, AuthName, AuthUserFile, Require valid-user.

2. Reto Práctico para la Sesión (2 horas)

El reto combina los tres contenidos vistos y se implementa sobre el Virtual Host de nuestro proyecto (C:\xampp\htdocs\wordpress).

Desafío del Laboratorio: Despliegue Seguro

El equipo debe realizar las siguientes tareas sobre su máquina virtual/contenedor de desarrollo:

Fase 1: Configuración SSL/TLS con Certificado Autofirmado (RA2.c)

1. **Generación:** Crear la clave privada (wordpress.key) y la Solicitud de Firma de Certificado (CSR).
2. **Firma:** Generar el **Certificado Autofirmado** (wordpress.crt) a partir de la CSR.
3. **Activación de HTTPS:** Modificar el *Virtual Host* para:
    - Escuchar en el **puerto 443**.
    - Cargar la clave privada y el certificado (.key y .crt).
4. **Redirección Forzada (RA2.b):** Configurar un bloque de *Virtual Host* en el puerto 80 para que, al acceder por HTTP, se **redirija automáticamente** al mismo dominio por HTTPS (puerto 443).
    - *Criterio de éxito: El navegador solo debe mostrar el tráfico cifrado.*

Fase 2: Protección de la Zona de Administración (RA2.a)

1. **Directorio:** Crear el directorio /gestion en la raíz de la aplicación (/var/www/html/wordpress/gestion).
2. **Credenciales:** Usar el comando **htpasswd** para crear un archivo de usuarios (.htpasswd) con al menos un usuario llamado admin con una contraseña segura.
3. **Protección:** Colocar el archivo **.htaccess** en el directorio /gestion para activar la **Autenticación Básica HTTP** y apuntar al archivo .htpasswd.
    - *Criterio de éxito: Al acceder a https://localhost/wordpress/gestion, el navegador debe mostrar un pop-up solicitando usuario y contraseña antes de cargar el contenido.*

Despliegue Seguro (Paso a Paso)

El problema principal en entornos XAMPP/WAMP es que OpenSSL no encuentra su archivo de configuración (openssl.cnf) en las rutas por defecto. La solución es **indicarle su ubicación correcta** mediante la variable de entorno OPENSSL_CONF.

FASE 0: Ejecución Exitosa de OpenSSL

Esta fase garantiza que sus archivos de certificado y clave privada se generen correctamente.

0.1. Abrir la Consola y Establecer la Ubicación

1. **Abra la Consola de Windows** (Símbolo del sistema o PowerShell).
2. **Navegue a la carpeta bin de Apache** (donde está el ejecutable openssl.exe):

   Bash

   ```
   cd C:/xampp/apache/bin
   ```
3. **Establezca la variable de entorno OPENSSL_CONF** con la ruta correcta del archivo openssl.cnf (asumiendo que está en conf):
   - **Si usa CMD (Símbolo del sistema):**

     Bash

     ```
     set OPENSSL_CONF=C:\xampp\apache\conf\openssl.cnf
     ```
   - **Si usa PowerShell:**

     PowerShell

     ```
     $env:OPENSSL_CONF="C:\xampp\apache\conf\openssl.cnf"
     ```

0.2. Generación del Certificado Autofirmado

Ejecute el comando para crear la clave privada y el certificado, guardándolos en la carpeta conf/ssl.

Bash

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \ -keyout C:/xampp/apache/conf/ssl.key/mi-aplicacion.key \
-out C:/xampp/apache/conf/ssl.crt/mi-aplicacion.crt
```

0.3. Responder a las Preguntas de OpenSSL

Rellene la información para el certificado (la respuesta para España es **ES**):

Pregunta de OpenSSL	Respuesta Ejemplo
Country Name (2 letter code) [AU]:	ES
State or Province Name (full name) [Some-State]:	Andalucia
Locality Name (eg, city) []:	Ubrique
Organization Name (eg, company) [Internet Widgits Pty Ltd]:	DAWEB_2DAW

Pregunta de OpenSSL	Respuesta Ejemplo
Organizational Unit Name (eg, section) []:	Despliegue
Common Name (e.g. server FQDN or YOUR name) []:	localhost
Email Address []:	(Vacío)
A challenge password []:	(Vacío)
An optional company name []:	(Vacío)

FASE I: Configuración SSL/TLS y Redirección (HTTPS)

Esta fase requiere la modificación del archivo de configuración de Virtual Hosts de Apache.

1. Modificación del Archivo hosts de Windows

Abra el archivo C:\Windows\System32\drivers\etc\hosts y añada esta línea para mapear el dominio local a su máquina:

```
127.0.0.1    localhost
```

2. Configuración de Virtual Hosts

Edite el archivo C:\xampp\apache\conf\extra\httpd-vhosts.conf (o el archivo de configuración de hosts en su WAMP).

A. Bloque para HTTP (Puerto 80) - Redirección

Este bloque captura el tráfico no seguro y lo envía a HTTPS.

Apache

```
<VirtualHost *:80>
   ServerName mi-aplicacion.local
   # Redirige todo a HTTPS de forma permanente
   Redirect permanent / https://localhost/wordpress/
</VirtualHost>
```

B. Bloque para HTTPS (Puerto 443) - Servidor Seguro

Este bloque configura Apache para usar los certificados.

Apache

```
<VirtualHost *:443>
   ServerName localhost:443
   DocumentRoot "C:/xampp/htdocs"

   # Activar el motor SSL/TLS y apuntar a los archivos generados
   SSLEngine on
   SSLCertificateFile "C:/xampp/apache/conf/ssl/wordpress.crt"
   SSLCertificateKeyFile "C:/xampp/apache/conf/ssl/wordpress.key"
```

```
  # Permite la lectura del .htaccess (Necesario para Fase II)
  <Directory "C:/xampp/htdocs">
    AllowOverride All
    Require all granted
  </Directory>
</VirtualHost>
```

3. Reinicio de Apache

Reinicie el servicio **Apache** desde el Panel de Control de XAMPP/WAMP.

FASE II: Autenticación Básica HTTP (Protección de wp-admin)

Esta fase asegura la zona de administración.

1. Creación del Archivo de Contraseñas (.htpasswd)

Utilice el ejecutable htpasswd (ubicado en C:\xampp\apache\bin\) para generar el archivo de credenciales.

1. **Navegue a la carpeta bin**: cd C:\xampp\apache\bin
2. **Genere el archivo:**

   Bash
   ```
   # Se usa -c (create) solo la primera vez. Guardar fuera de htdocs.
   htpasswd -c C:\xampp\apache\.htpasswd admin
   # Introduzca la contraseña
   ```

2. Creación del Archivo .htaccess en wp-admin

Cree un nuevo archivo llamado **.htaccess** en la carpeta **wp-admin** de su instalación de WordPress:

- **Ruta:** C:\xampp\htdocs\wordpress\wp-admin\.htaccess

Contenido del .htaccess (con la excepción de WordPress):

Apache

```
# 1. Definicion de la autenticacion
AuthType Basic
AuthName "Acceso Restringido - Area de Administracion WP"
# 2. Ruta ABSOLUTA al archivo de contraseñas (usar barras inclinadas)
AuthUserFile "C:/xampp/apache/.htpasswd"
# 3. Requerir cualquier usuario valido
Require valid-user

# EXCEPCION ESENCIAL: Evita que el frontend de WordPress se rompa
<Files admin-ajax.php>
  Allow from all
  Satisfy any
</Files>
```

3. Verificación Final del Reto

1. Acceda a: http://localhost/wordpress (Debe redirigir a HTTPS).
2. Acceda a: https://localhost/wordpress/wp-admin/ (Debe pedirle primero las credenciales de **Apache** (admin), y luego las credenciales de **WordPress**).

Con estos pasos, habrá completado con éxito la configuración de seguridad para el **RA2** sobre la arquitectura XAMPP/WordPress.

Anexo: Eliminación de la Alerta de Seguridad (Opcional)

Al usar un certificado autofirmado, los navegadores muestran por defecto una advertencia de seguridad porque no reconocen nuestra autoridad de certificación personal. Para simular un entorno de producción real y ver el **candado de seguridad** cerrado, debemos instalar nuestro certificado en el sistema.

A. Para Google Chrome, Microsoft Edge y Opera (Almacén de Windows)

Estos navegadores utilizan el almacén de certificados del sistema operativo Windows.

1. Localizar el certificado:

   Navegue a la carpeta donde generó el certificado: C:\xampp\apache\conf\ssl\.

   Haga doble clic sobre el archivo localhost.crt.

2. Iniciar el Asistente:

   Se abrirá una ventana con los detalles del certificado. Pulse el botón "Instalar certificado...".

3. **Ubicación del Almacén:**
   - Seleccione **"Equipo local"** (recomendado) o "Usuario actual".
   - Pulse **Siguiente**.
4. **Selección de la Carpeta (PASO CRÍTICO):**
   - No elija la opción automática. Marque la casilla: **"Colocar todos los certificados en el siguiente almacén"**.
   - Pulse **"Examinar..."**.
   - Seleccione la carpeta: **"Entidades de certificación raíz de confianza"** (*Trusted Root Certification Authorities*).
   - Pulse **Aceptar** y luego **Siguiente**.
5. Finalizar:

   Pulse Finalizar. Windows le mostrará un aviso de seguridad preguntando si realmente confía en este certificado. Pulse Sí.

6. Comprobación:

   Cierre completamente el navegador y vuelva a abrirlo. Al entrar en https://localhost/wordpress, debería ver el candado cerrado sin advertencias.

B. Para Mozilla Firefox

Firefox gestiona su propio almacén de certificados independiente de Windows, por lo que la configuración se hace dentro del navegador.

1. Abra Firefox y vaya a **Ajustes** (Settings).
2. En el buscador de ajustes (arriba a la derecha), escriba: **Certificados**.
3. Pulse el botón **"Ver certificados..."**.
4. En la pestaña **"Autoridades"**, pulse el botón **"Importar..."**.
5. Busque y seleccione su archivo C:\xampp\apache\conf\ssl\localhost.crt.
6. Aparecerá una ventana emergente ("Descargando certificado"). Marque la casilla:
    - **Confiar en esta CA para identificar sitios web.**
7. Pulse **Aceptar**.

Al recargar la página https://localhost/wordpress, Firefox confiará en su servidor local.

3. Guía Práctica: Despliegue en AWS Academy (Paso a Paso)

FASE 1: Acceso y Arranque del Laboratorio (Sesión 04/12)

A diferencia de una cuenta AWS normal, en Academy el entorno es controlado.

1. **Login:** El alumno accede al portal de estudiantes de AWS Academy (Canvas).
2. **Módulo Learner Lab:** Entrar en "Modules" -> "Learner Lab - Foundational Services".
3. **Iniciar Laboratorio (CRÍTICO):**
    - Pulsar **"Start Lab"**. El círculo de estado pasará de rojo a amarillo y finalmente a verde.
    - *Explicación:* Esto enciende la cuenta temporal de AWS y asigna el presupuesto (usualmente $100).
4. **Descargar Clave SSH:**
    - En el panel de instrucciones del laboratorio (derecha o arriba), buscar **"AWS Details"**.
    - Descargar la **SSH Key** (suele llamarse labsuser.pem o hay que crear una vockey).
    - *Nota:* En los laboratorios modernos, se recomienda usar la clave pre-generada **vockey**. Si no, crearán una en el lanzamiento.
5. **Entrar a la Consola:** Pulsar el enlace rojo **"AWS"** (punto verde activo). Esto abre la consola real de AWS en una nueva pestaña.

FASE 2: Lanzamiento de la Instancia EC2 (El Servidor)

1. **Región:** Verificar arriba a la derecha que están en **N. Virginia (us-east-1)**. Learner Lab suele restringir el uso a esta región.
2. **Servicio EC2:** Buscar "EC2" y pulsar en "Lanzar instancia" (*Launch Instance*).
3. **Configuración de la Máquina:**
    - **Nombre:** Servidor-Web-Alumno
    - **Imagen (AMI):** Seleccionar **Ubuntu** (Ubuntu Server 22.04 LTS o 24.04).
    - **Tipo de Instancia: t2.micro** (Apta para la capa gratuita/educativa).
    - **Par de claves (Login):** Seleccionar **vockey** (Esta clave se inyecta automáticamente por el Learner Lab).
4. **Configuración de Red (Security Groups):**
    - En "Configuración de red", pulsar "Editar".
    - **Crear grupo de seguridad**. Nombre: Permisos-Web.
    - Añadir reglas de entrada (Inbound rules):

- **SSH (22):** Origen Anywhere (0.0.0.0/0) - *Para administrar.*
- **HTTP (80):** Origen Anywhere (0.0.0.0/0) - *Para ver la web.*
- **HTTPS (443):** Origen Anywhere (0.0.0.0/0) - *Para el futuro SSL.*

5. **Lanzar:** Pulsar "Lanzar instancia".

FASE 3: Conexión SSH y Preparación del Entorno

Aquí el alumno aprenderá a administrar un servidor sin interfaz gráfica.

Opción A: Windows con PuTTY (Clásico)

Requiere convertir la clave .pem a .ppk con PuTTYgen. Es engorroso.

Opción B: Windows con PowerShell / CMD (Recomendado)

Windows 10/11 ya trae cliente SSH nativo.

1. Ir a la carpeta donde descargaron la clave labsuser.pem (desde AWS Details).
2. Abrir terminal allí.
3. **Comando de conexión:**

   PowerShell

   ```
   # El usuario por defecto en Ubuntu es 'ubuntu'
   # La IP pública se ve en la consola AWS seleccionando la instancia.
   ssh -i labsuser.pem ubuntu@<DIRECCION_IP_PUBLICA>
   ```

   (Aceptar la huella digital escribiendo "yes").

FASE 4: Instalación de la Pila LAMP (Sesión 04/12 y 11/12)

Una vez dentro de la terminal negra de Linux (AWS), ejecutarán:

Bash

```
# 1. Actualizar repositorios
sudo apt update

# 2. Instalar Apache
sudo apt install apache2 -y

# 3. Instalar PHP y módulos necesarios
sudo apt install php libapache2-mod-php php-mysql -y

# 4. Instalar MySQL (Servidor de Base de datos)
sudo apt install mysql-server -y
```

Comprobación Rápida:

El alumno copia su IP Pública de AWS y la pega en el navegador del móvil o del PC: http://54.x.x.x.

- *Resultado:* Debe ver la página "Apache2 Ubuntu Default Page". ¡Ya tienen servidor en la nube!

FASE 5: Despliegue de la Aplicación (Sesión 11/12)

Aquí conectamos con lo aprendido en las semanas anteriores.

1. Clonar/Subir código:

   En lugar de FTP (que requiere configurar servidor FTP), usaremos Git o Wget para bajar WordPress.

   Bash

   ```
   cd /var/www/html
   sudo rm index.html
   sudo wget https://wordpress.org/latest.tar.gz
   sudo tar -xzvf latest.tar.gz
   sudo mv wordpress/* .
   ```

2. Permisos (Concepto clave de Linux):

   Apache (usuario www-data) necesita permiso para escribir.

   Bash

   ```
   sudo chown -R www-data:www-data /var/www/html/
   sudo chmod -R 755 /var/www/html/
   ```

3. Base de Datos:

   Crear la base de datos para WordPress en MySQL.

   Bash

   ```
   sudo mysql
   # Dentro de MySQL:
   CREATE DATABASE mi_web;
   CREATE USER 'alumno'@'localhost' IDENTIFIED BY 'password_segura';
   GRANT ALL PRIVILEGES ON mi_web.* TO 'alumno'@'localhost';
   EXIT;
   ```

4. Instalación Web:

   Acceder a http://<IP_PUBLICA>/ y completar la instalación de WordPress.