Tarefa Práctica 4: Securización dun Servidor LAMP Nativo con HTTPS

Obxectivos da Práctica

- Comprender os conceptos básicos de SSL/TLS e a necesidade de cifrar o tráfico web con HTTPS.
- Aprender a xerar un certificado SSL/TLS autofirmado usando OpenSSL.
- Configurar un Virtual Host de Apache para servir contido a través do porto 443 (HTTPS).
- Activar os módulos de Apache necesarios para a xestión de SSL.
- Axustar as regras do firewall para permitir o tráfico HTTPS.
- Implementar unha redirección permanente de HTTP a HTTPS, unha práctica estándar de seguridade.

Contexto

Na primeira práctica, montamos un servidor LAMP completamente funcional, pero toda a comunicación entre o navegador e o servidor viaxaba en texto plano (HTTP). Calquera persoa nunha rede intermedia podería interceptar e ler os datos. Para calquera aplicación real, isto é inaceptable.

Nesta práctica, imos securizar o noso servidor. Como non dispoñemos dun dominio público real para obter un certificado dunha Autoridade de Certificación (CA) como Let's Encrypt, crearemos o noso propio **certificado autofirmado**. Este tipo de certificado ofrece o mesmo nivel de cifrado, pero non é recoñecido automaticamente polos navegadores, o que nos ensinará unha lección valiosa sobre a "cadea de confianza" en internet.

Requisitos Previos

- Ter completada a Práctica 1 ("Instalación e Configuración dun Servidor LAMP..."), incluíndo a creación do Virtual Host `proxecto.daw`.
- Unha máquina virtual con Ubuntu 24.04 coa pila LAMP nativa funcionando.
- Un número de equipo asignado (`XX`).

Fases da Práctica

Fase 1: Xeración do Certificado SSL

Usaremos a ferramenta `openssl` para crear simultaneamente a nosa clave privada e o noso certificado público.

1. **Crear a Clave e o Certificado:**
   Executa o seguinte comando na terminal. Este comando creará unha clave RSA de 2048 bits e un certificado válido por 365 días.

   ```
   sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
   ```

   O comando solicitarache varios datos. A maioría podes deixalos por defecto premendo `Intro`, pero o máis importante é o último:

   - `Common Name (e.g. server FQDN or YOUR name) []`: Aquí debes escribir o dominio exacto que queres asegurar. Para esta práctica, será `proxectoXX.daw` (substitúe `XX` polo teu número de equipo).

2. **Verificar os Ficheiros Creados:**
   Comproba que os dous ficheiros existen nas súas respectivas localizacións:

   - A clave privada: `sudo ls -l /etc/ssl/private/apache-selfsigned.key`
   - O certificado público: `sudo ls -l /etc/ssl/certs/apache-selfsigned.crt`

Fase 2: Configuración de Apache para Usar SSL

Agora debemos dicirlle a Apache como usar estes ficheiros.

1. **Activar o Módulo SSL:**
   Apache necesita o seu módulo SSL para entender HTTPS. Tamén activaremos o módulo de cabeceiras, que é unha boa práctica.

   ```
   sudo a2enmod ssl
   sudo a2enmod headers
   ```

2. **Crear un Virtual Host para HTTPS:**
   A mellor práctica é ter un ficheiro de configuración separado para o tráfico seguro. Copiaremos o noso Virtual Host existente para crear unha versión para o porto 443.

   ```
   sudo cp /etc/apache2/sites-available/proxecto.daw.conf /etc/apache2/sites-available/proxecto.daw-ssl.conf
   ```

3. **Editar o Novo Ficheiro de Virtual Host SSL:**
   Abre o novo ficheiro co teu editor.

   ```
   sudo nano /etc/apache2/sites-available/proxecto.daw-ssl.conf
   ```

   Modificao para que teña o seguinte contido, prestando especial atención ás liñas novas ou

modificadas:

```
<VirtualHost *:443>  # Modificado para escoitar no porto 443
    ServerAdmin admin@proxecto.daw
    ServerName proxecto.daw
    ServerAlias www.proxecto.daw
    DocumentRoot /var/www/proxecto.daw

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # --- Sección SSL Engadida ---
    SSLEngine on
    SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile   /etc/ssl/private/apache-selfsigned.key

    # --- Cabeceiras de Seguridade Recomendadas (Boa práctica) ---
    <FilesMatch "\.(cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>
    Header always set Strict-Transport-Security "max-age=63072000"
</VirtualHost>
```

Fase 3: Redirixir o Tráfico de HTTP a HTTPS

Non queremos que os usuarios poidan acceder á versión insegura. Imos forzar a todos a usar HTTPS.

1. **Editar o Virtual Host Orixinal (Porto 80):**
   Abre o ficheiro de configuración orixinal.
   `sudo nano /etc/apache2/sites-available/proxecto.daw.conf`
2. Modifica o seu contido para que simplemente redirixa todo o tráfico. Substitúe `XX` polo teu número de equipo.

```
<VirtualHost *:80>
    ServerName proxecto.daw
    ServerAlias www.proxecto.daw

    Redirect permanent / https://proxectoXX.daw/
</VirtualHost>
```

Fase 4: Activación da Nova Configuración

Agora aplicamos todos os cambios.

1. **Activar o Novo Sitio SSL:**
   `sudo a2ensite proxecto.daw-ssl.conf`
2. **Axustar o Firewall:**
   Debemos permitir o tráfico a través do porto 443. O perfil `Apache Full` abre tanto o 80 como o 443.
   `sudo ufw allow 'Apache Full'`

3. **Verificar a Sintaxe da Configuración:**
   Este é un paso crucial para detectar erros antes de reiniciar o servizo.
   `sudo apache2ctl configtest`
   Se todo está correcto, deberías ver "Syntax OK".
4. **Recargar Apache:**
   Para aplicar todos os cambios sen cortar conexións activas, usamos `reload`.
   `sudo systemctl reload apache2`

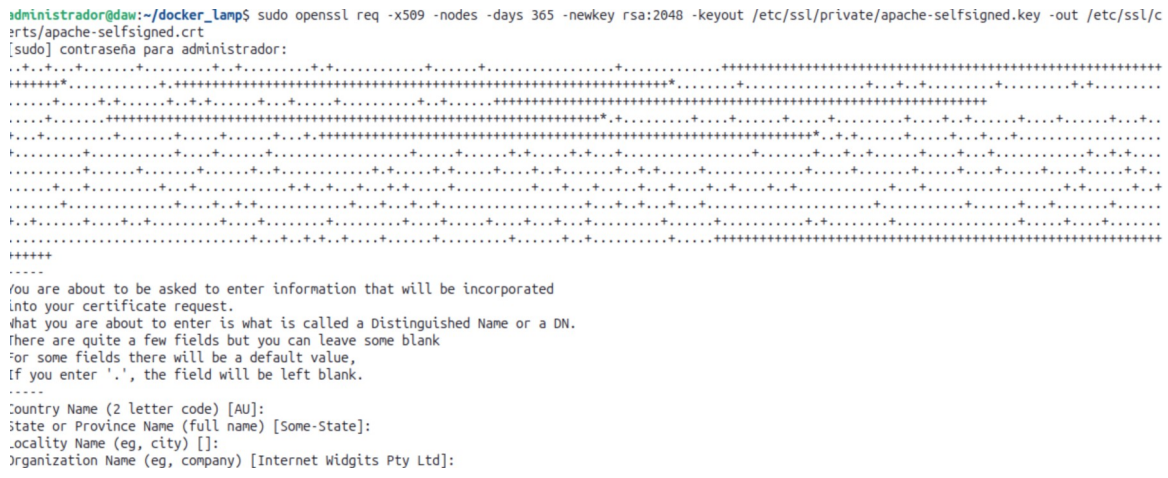
Fase 5: Verificación Final no Navegador

1. **Modifica o Ficheiro `hosts`:** Asegúrate de que o teu ficheiro `hosts` na túa máquina física (anfitrioa) aínda apunta o dominio `proxectoXX.daw` á IP da túa VM.
2. **Proba a Redirección:** Abre o teu navegador e visita `http://proxectoXX.daw` (con **http**). Deberías ser redirixido automaticamente a `https://proxectoXX.daw`.
3. **Acepta a Advertencia de Seguridade:** O teu navegador amosará unha advertencia de seguridade a toda páxina ("A súa conexión non é privada", "Potencial risco de seguridade"). **Isto é esperado e correcto!** Ocorre porque o certificado foi asinado por nós mesmos, non por unha autoridade na que o navegador confíe.
   - Fai clic en "Avanzado" ou "Configuración avanzada".
   - Busca a opción "Aceptar o risco e continuar" ou "Continuar a proxectoXX.daw (non seguro)".
4. **Verifica o Resultado:** Unha vez aceptada a excepción de seguridade, deberías ver a túa páxina de "Éxito!" de prácticas anteriores. No canto do enderezo, deberías ver un **icona de cadeado**, probablemente tachado ou cunha advertencia, indicando que a conexión está cifrada pero a confianza non puido ser verificada.

Instrucións de Entrega

Deberás entregar un ficheiro **PDF** que inclúa:

1. **Capturas de pantalla** que demostren o correcto funcionamento:
   - A saída do comando `openssl` durante a creación do certificado.



- O contido do teu ficheiro de configuración `proxecto.daw-ssl.conf`.

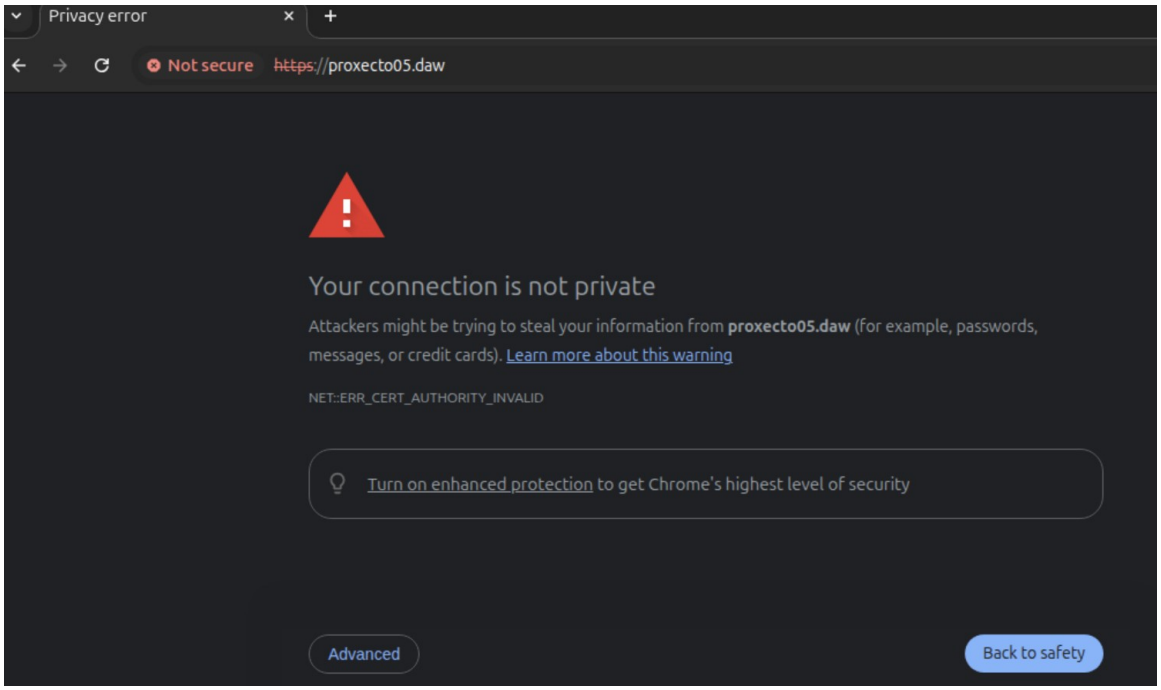
```
GNU nano 7.2                                    /etc/apache2/sites-available/proxecto.daw-ssl.conf
    <VirtualHost *:443>  # Modificado para escoitar no porto 443
        ServerAdmin admin@proxecto.daw
        ServerName proxecto.daw
        ServerAlias www.proxecto.daw
        DocumentRoot /var/www/proxecto.daw

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # --- Sección SSL Engadida ---
        SSLEngine on
        SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
        SSLCertificateKeyFile   /etc/ssl/private/apache-selfsigned.key

        # --- Cabeceiras de Seguridade Recomendadas (Boa práctica) ---
        <FilesMatch "\.(cgi|shtml|phtml|php)$">
            SSLOptions +StdEnvVars
        </FilesMatch>
        Header always set Strict-Transport-Security "max-age=63072000"
    </VirtualHost>
```

- O navegador amosando a advertencia de seguridade.



- O navegador amosando a páxina final con HTTPS na barra de enderezos e o cadeado visible.

2. **Análise e Reflexión:**
   - Coas túas propias palabras, que é un certificado autofirmado e cal é a súa principal limitación en comparación cun de Let's Encrypt?

     Que no se trata de una certificación ofical entonces esta mas limitada porque o navegador non a considera cifrada.

   - Por que é importante forzar a redirección de todo o tráfico de HTTP a HTTPS?

     Porque todas as paxinas web oficais usan este estandar de seguridade que se volto indispensable.