

## PRÁCTICA 5: FIRMA DE APLICACIONES WEB CON HTTPS (APACHE Y NODE.JS)

### OBJETIVO

Configurar certificados SSL autofirmados en nuestro servidor para servir contenidos de forma segura (HTTPS) tanto en Apache como en Node.js.

### MATERIAL NECESARIO

- Ubuntu Server configurado (Prácticas anteriores completadas).
- Apache2 y Node.js instalados.
- OpenSSL instalado (Ubuntu ya lo tiene por defecto).

### PASOS

#### 1. Crear un certificado SSL autofirmado

1.1. Crear carpeta para certificados:

```
sudo mkdir /etc/ssl/misitio
```

1.2. Generar certificado autofirmado (válido 1 año):

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

Durante la generación te pedirá:

- País, Estado, Ciudad...
- **Common Name:** pon la IP de tu servidor o el dominio (si tuvieras).

#### 2. Configurar HTTPS en Apache2

2.1. Activar los módulos SSL:

```
sudo a2enmod ssl
```

2.2. Crear configuración HTTPS para tu sitio:

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

Contenido:

```
<VirtualHost *:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/misitio
    SSLEngine on
    SSLCertificateFile /etc/ssl/misitio/misitio.crt
    SSLCertificateKeyFile /etc/ssl/misitio/misitio.key
    ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

2.3. Activar el nuevo sitio HTTPS:

```
sudo a2ensite misitio-ssl.conf
```

2.4. Recargar Apache:

```
sudo systemctl reload apache2
```

2.5. Abrir en el navegador:

```
https://IP_DEL_SERVIDOR/
```

#### Notas:

- El navegador te advertirá que es un certificado no confiable (es normal porque es autofirmado).

---

### 3. Configurar HTTPS en Node.js

3.1. Modificar el servidor Node.js para HTTPS:

```
nano ~/miprojectonode/server.js
```

Nuevo contenido:

```
const https = require('https');
const fs = require('fs');

const options = {
  key: fs.readFileSync('/etc/ssl/misitio/misitio.key'),
  cert: fs.readFileSync('/etc/ssl/misitio/misitio.crt')
};

https.createServer(options, (req, res) => {
  res.writeHead(200);
  res.end('¡Hola desde Node.js seguro con HTTPS!');
}).listen(3000, () => {
  console.log('Servidor HTTPS en puerto 3000');
});
```

3.2. Ejecutar el servidor:

```
node server.js
```

3.3. Acceder desde navegador:

```
https://IP_DEL_SERVIDOR:3000
```

---

**Resultado esperado**

- Sitio Apache sirviendo contenido en HTTPS.
- Servidor Node.js sirviendo en HTTPS.
- Se reciben advertencias por certificado autofirmado, pero la conexión es segura.