Práctica 2.6. Servidor Web Seguro

Trabajo realizado por: Javier Epifanio López

1. **Describe con tus palabras:**
    1. ***¿Qué es un servidor web seguro?***
    Se dice que un servidor web es seguro si las comunicaciones que se establecen desde los navegadores utilizan cifrado para garantizar la confidencialidad y además permite garantizar al usuario la identidad de dicho servidor.

    2. ***¿Qué es el archivo pem?***
    PEM son las siglas de correo de privacidad mejorada. El formato PEM se utiliza a menudo para representar certificados, solicitudes de certificados, cadenas de certificados y claves.

    3. ***¿Qué es el archivo csr y qué contiene?***
    El CSR contiene información que será incluida finalmente en el certificado SSL, como por ejemplo tu nombre o el de a la empresa, la dirección, el país de residencia o el common name (dominio para el que es generado el SSL), además de estos datos también incluirá una clave pública que será incluida también en tu certificado.

    4. ***¿Qué es archivo key?***
    El archivo key representan ambas partes de un certificado, siendo la clave la clave privada del certificado.

    5. ***¿Cuál/es de todos estos tiene el servidor? ¿Y los clientes?***
    El servidor tiene todos, y el cliente tiene el archivo: pem.

2. **Genera tu propio certificado con OpenSSL.**
    2.1 Instala previamente el openSSL : apt install openssl
    El paquete openssl puede venir instalado por defecto en Ubuntu, en mi caso ya venia instalada y nos damos cuenta que a la hora de instalarlo nos dice que esta en su versión más reciente.

```
javiepi@ubusrv-jepi:~$ sudo apt-get install openssl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openssl ya está en su versión más reciente (1.1.1f-1ubuntu2.10).
fijado openssl como instalado manualmente.
```

    2.2 Modifica el fichero openssl.conf con información de la empresa, pais, etc.
    Modifico el fichero opens.cnf y le añado la información necesaria, en la siguiente imagen lo podemos comprobar.

```
  GNU nano 4.8                        /etc/ssl/openssl.cnf                  Modified
# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = ES
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Cordoba

localityName                    = Locality Name (eg, city)

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = epifanioDAW

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
#organizationalUnitName_default =

commonName                      = epifanio.epifanioDAW2.org
commonName_max                  = 64

emailAddress                    = a19eploja@iesgrancapitan.org
emailAddress_max                = 64

# SET-ex3                       = SET extension number 3

[ req_attributes ]
challengePassword               = A challenge password_
```

2.3 Genera el .pem

Ej: openssl req -x509 -newkey rsa:2048 -days 3650 -keyout misiteKey.key -out misitecert.pem

Nota: es recomendable mover el fichero .key a "etc/ssl/private" y el fichero .pem a "etc/ssl/certs"

Aquí generamos el archivo .pem que es el que va a recebir el cliente.

```
javiepi@ubusrv-jepi:~$ sudo openssl req -x509 -newkey rsa:2048 -days 3650 -keyout /etc/ssl/private/e
pifaniodaw2.key -out /etc/ssl/certs/epifaniodaw2.pem
Generating a RSA private key
.....................................+++++
.............+++++
writing new private key to '/etc/ssl/private/epifaniodaw2.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [ES]:ES
State or Province Name (full name) [Cordoba]:Cordoba
Locality Name (eg, city) []:Cordoba
Organization Name (eg, company) [epifanioDAW]:epifanioDAW
Organizational Unit Name (eg, section) []:
epifanio.epifanioDAW2.org []:epifanio.epifaniodaw2.org
a19eploja@iesgrancapitan.org []:a19eploja@iesgrancapitan.org
javiepi@ubusrv-jepi:~$
```

Aquí vamos a generar el archivo .key

```
javiepi@ubusrv-jepi:/etc/nginx/certificate$ sudo openssl req -new -newkey rsa:4096 -x509 -sha256 -da
ys 365 -nodes -out nginx-certificate.crt -key
-key      -keyform  -keyout
javiepi@ubusrv-jepi:/etc/nginx/certificate$ sudo openssl req -new -newkey rsa:4096 -x509 -sha256 -da
ys 365 -nodes -out nginx-certificate.crt -keyout nginx.key
Generating a RSA private key
.....................................................................................................
.....................................................................................+++++
....++++
writing new private key to 'nginx.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [ES]:ES
State or Province Name (full name) [Cordoba]:Cordoba
Locality Name (eg, city) []:Cordoba
Organization Name (eg, company) [epifanioDAW]:epifanioDAW
Organizational Unit Name (eg, section) []:
epifanio.epifanioDAW2.org []:epifanio.epifanioDAW2.org
a19eploja@iesgrancapitan.org []:a19eploja@iesgrancapitan.org
javiepi@ubusrv-jepi:/etc/nginx/certificate$
```

3. Configuración del sitio web seguro y acceso desde un cliente.

3.1 Sigue los pasos del curso para conseguir acceder también en modo seguro (https) a tu página web de prueba.

- Nginx seguro

Seguimos los pasos que del curso para acceder de modo seguro a mi página web

3.2 Configurar el servidor web para acceder por https.

Muestra los cambios en el fichero de configuración del sitio web.

Aquí os muestro el fichero de conf de mi site.

```
server{
        listen 443 ssl default_server;
        listen [::]:443 ssl default_server;

        # Certificados web segura (key y crt)
        ssl_certificate /etc/nginx/certificate/nginx-certificate.crt;
        ssl_certificate_key /etc/nginx/certificate/nginx.key;

        root /var/www/html/epifanio/;
        index index.html;

        location / {
                try_files $uri $uri/ =404;
        }

}
```

3.3 Prueba el servidor web seguro:

3.3.1 Haz una captura de pantalla, tanto de forma segura (https) https://IPdelservidorwebvirtual

Aquí os muestro como funciona el https:// (web segura).

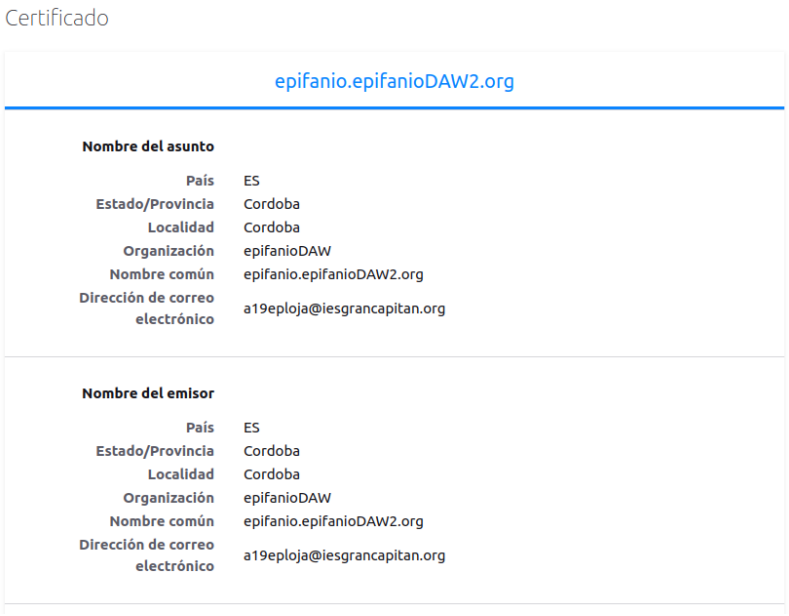
3.3.2 ....como de forma 'insegura' (http): http://IPdelservidorwebvirtual ¿Qué ocurre? ¿Cuál está funcionando correctamente y cuál no?

Cuando intento acceder al http me redirige automáticamente al https:// (web segura), ya que en mi fichero de conf de mi sitio web le he puesto un return a la web segura.

```
# Return para redirigir al https://epifanio.epifaniodaw2.org
return 301 https://$host$request_uri;
```

3.3.3 *muestra el certificado que se ha añadido a tu lista de certificados del navegador. En lugar de "Añadir la excepción" cada vez que accedamos, lo que haremos es instalar el certificado raíz Cacert en nuestro navegador para que lo reconozca.*

Aquí os muestro la información de mi certificado que he creado anteriormente en mi navegador web.



3.4 *Forzar ahora que lo anterior funcione, es decir, aunque se acceda de modo no seguro (http) hacer que se pase a modo seguro (https).*

https://techexpert.tips/es/nginx-es/nginx-redirigir-http-a-https/
Usar para ello "RETURN" (REDIRECT en Apache) de modo que al acceder por http automaticamente te redirija a https.
*Ahora la configuración del *:80 se elimina y sólo queda la mínima para que se acceda e inmediatamente se redirija al site seguro (return).*
*La configuración del site completa estará en el *:443*

```
# Return para redirigir al https://epifanio.epifaniodaw2.org
return 301 https://$host$request_uri;
```