

DAW Práctica 2.5: Certificados Nginx en host virtual

Elabora un documento donde figuren todos los pasos realizados con las pantallas significativas, explicando cada uno de los pasos.

Procedimiento:

Nginx

1. Creamos un directorio llamado ssl en /etc/nginx
2. Dentro del directorio recientemente creado generamos la clave privada con el cifrado RSA y la longitud 2048 bits. Pedirá introducir el nombre que vamos a generar como clave.

```
sudo openssl genpkey -algorithm RSA -out server.key -pkeyopt  
rsa_keygen_bits:2048
```

3. Completa las funcionalidades del anterior comando

```
genpkey  
-algorithm  
-pkeyopt  
Numbits (2048)
```

4. Generamos el certificado autofirmado:

```
sudo openssl req -x509 -nodes -days 365 -key server.key -out server.crt
```

5. ¿Esta configuración cuenta con passphrase? ¿En la anterior práctica estaba implementada? ¿Qué diferencia hay entre tener o no tener passphrase?

6. Completa las funcionalidades del anterior comando

```
req  
-nodes  
-days  
-key  
-out
```

7. Incluir el certificado en Nginx modificando el fichero de configuración de nuestro host virtual. En el fichero de configuración se deberá añadir una nueva configuración de servidor con, al menos, las siguientes características:

- a. SSL escuche por el puerto 443 (listen 443 ssl).
- b. El nombre del servidor.
- c. Raíz desde donde buscará los archivos.
- d. La referencia a la clave y a los certificados.

```
ssl_certificate /etc/nginx/ssl/server.crt;  
ssl_certificate_key /etc/nginx/ssl/server.key;
```

- e. Location / para que busque las url o muestre el error 404 al no encontrarlas.

8. Confirmar la correcta sintaxis de los ficheros de configuración.

9. Reiniciar el servicio.

10. ¿Qué aparece al intentar acceder con https? ¿Por qué sucede?

11. Ahora que se ha implementado https puede ser interesante que se redirija el posible tráfico de http. Para ello, en la parte del puerto 80 del fichero de configuración, si estamos trabajando con Nginx añadimos la siguiente instrucción.

```
return 301 https://$host$request_uri;
```

12. Reiniciamos el servicio.

13. Probamos la nueva implementación.