

PRÁCTICA 5

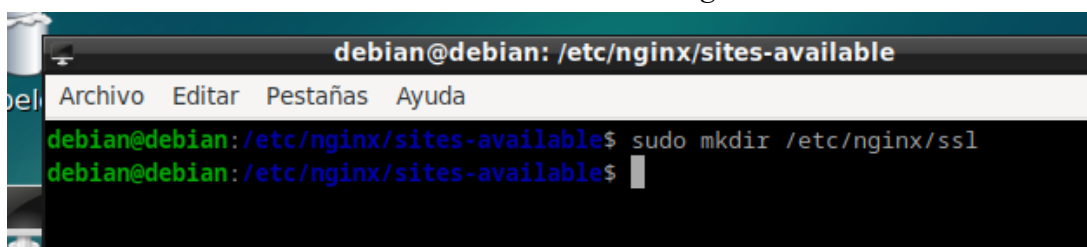
Nginx como Proxy Inverso/Balanceador con cifrado

En esta práctica última práctica de la unidad 2, modificaremos lo que llevamos hecho hasta ahora de manera que la comunicación que se lleva en nuestro sistema sea cifrada, con el objetivo de añadir una mayor seguridad.

Puesto que nuestros servidores (los 2 servidores origen y el proxy) se encuentran en una red privada, no sería necesario que la comunicación entre éstos se cifrara. Por tanto, el cifrado solo se llevará a cabo entre el proxy y las peticiones de clientes que se reciban. De esta forma, también vamos a conseguir liberar a los servidores origen de tareas como el cifrado/descifrado, quedando totalmente disponibles para servir páginas web.

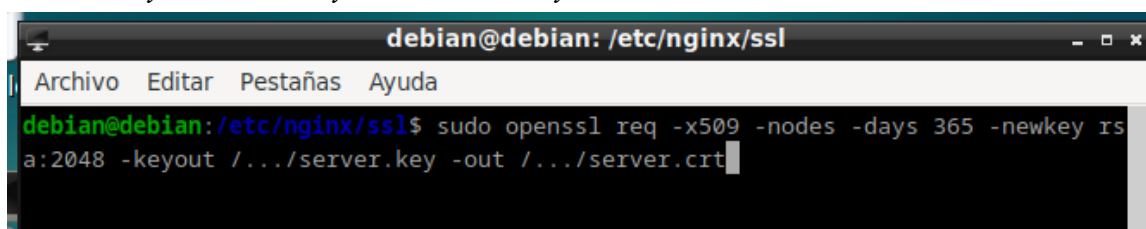
Vamos a utilizar el protocolo HTTPS, que se basa en el uso de certificados digitales. Por tanto, lo primero que haremos será crearnos un certificado (que autofirmaremos):

1. Nos creamos un directorio llamado *ssl* dentro de */etc/nginx/*.



```
debian@debian: /etc/nginx/sites-available
Archivo Editar Pestañas Ayuda
debian@debian:/etc/nginx/sites-available$ sudo mkdir /etc/nginx/ssl
debian@debian:/etc/nginx/sites-available$
```

2. Nos creamos el certificado, junto con el par de claves (pública y privada que lleva asociadas), con el siguiente comando: *sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout ../../server.key -out ../../server.crt*

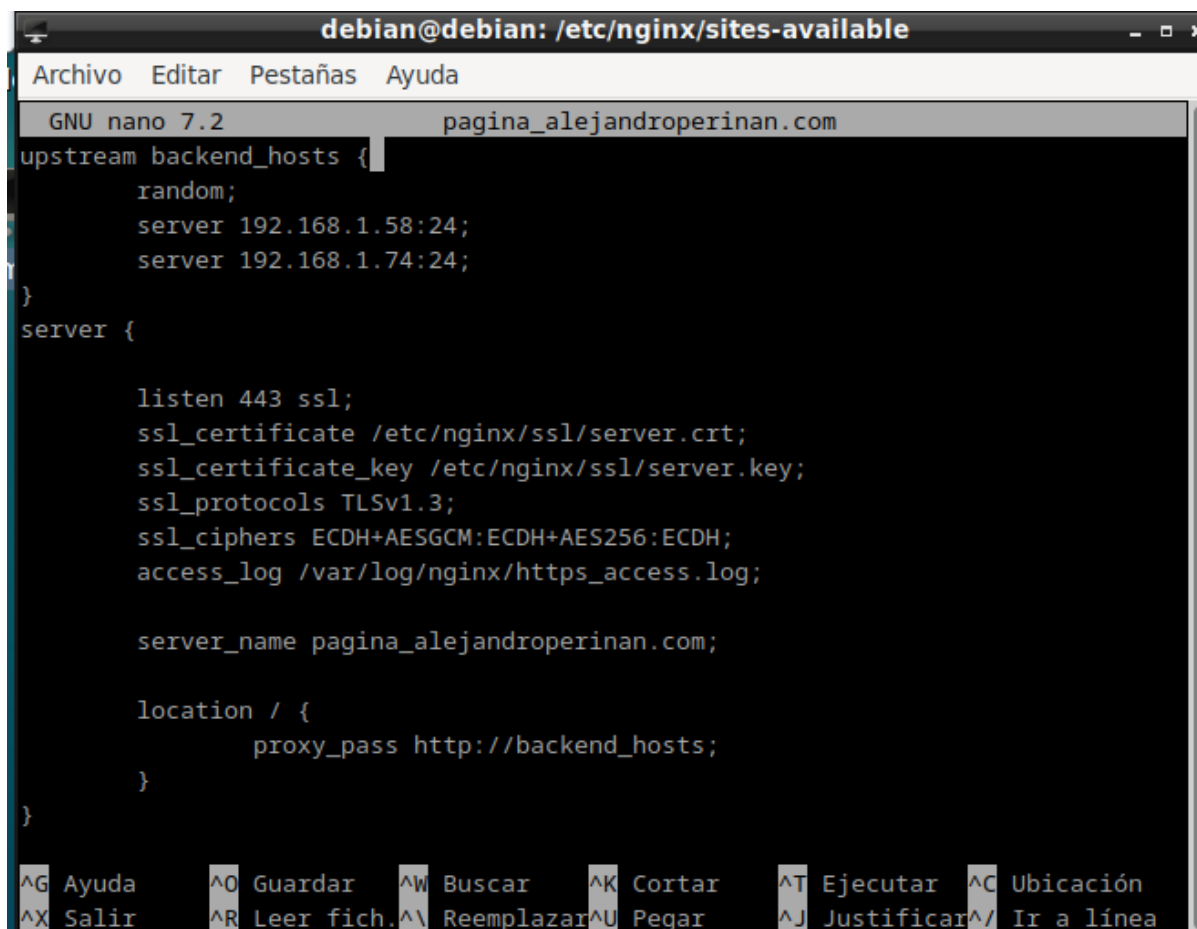


```
debian@debian: /etc/nginx/ssl
Archivo Editar Pestañas Ayuda
debian@debian:/etc/nginx/ssl$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout ../../server.key -out ../../server.crt
```

3. Investiga qué especifica cada una de las partes del comando anterior. Al ejecutar el comando anterior, se nos pedirán una serie de datos que iremos rellenando. Comprobar que tenemos los ficheros de clave y certificado en el directorio correspondiente.
req -x509: Sirve para generar un CSR X509.
-days 365: Sirve para que esté activo el certificado durante 365 días.
-newkey rsa:2048: Sirve para generar una nueva clave privada RSA de 2048 bits.

4. Ahora nos queda modificar el archivo de configuración del proxy para adaptarlo al protocolo HTTPS. Se debe añadir lo siguiente en el bloque del servidor:

```
listen 443 ssl;
ssl_certificate /etc/nginx/ssl/server.crt;
ssl_certificate_key /etc/nginx/ssl/server.key;
ssl_protocols TLSv1.3;
ssl_ciphers ECDH+AESGCM:ECDH+AES256:ECDH;
server_name 192.168.18.20;
access_log /var/log/nginx/https_access.log;
```



The screenshot shows a terminal window titled "debian@debian: /etc/nginx/sites-available". The window contains the nano 7.2 editor with a file named "pagina_alejandroperinan.com". The configuration file content is as follows:

```
upstream backend_hosts {
    random;
    server 192.168.1.58:24;
    server 192.168.1.74:24;
}

server {

    listen 443 ssl;
    ssl_certificate /etc/nginx/ssl/server.crt;
    ssl_certificate_key /etc/nginx/ssl/server.key;
    ssl_protocols TLSv1.3;
    ssl_ciphers ECDH+AESGCM:ECDH+AES256:ECDH;
    access_log /var/log/nginx/https_access.log;

    server_name pagina_alejandroperinan.com;

    location / {
        proxy_pass http://backend_hosts;
    }

}
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts: ^G Ayuda, ^O Guardar, ^W Buscar, ^K Cortar, ^T Ejecutar, ^C Ubicación, ^X Salir, ^R Leer fich., ^\ Reemplazar, ^U Pegar, ^J Justificar, ^_ Ir a línea.

5. Indica para qué sirven las líneas anteriores.
- listen 443 ssl: sirve para escuchar por el puerto 443 que es el del protocolo https y también por ssl
 - ssl_certificate /etc/nginx/ssl/server.crt: Indica dónde está el certificado.
 - ssl_certificate_key /etc/nginx/ssl/server.key: Indica la clave del certificado.
 - ssl_protocols TLSv1.3: indica la versión del protocolo ssl.
 - ssl_ciphers ECDH+AESGCM:ECDH+AES256:ECDH: son suites de ssl.
 - access_log /var/log/nginx/https_access.log: Indica la ruta del log del https.

6. Accede ahora al servidor e indica qué te aparece en el navegador. ¿Por qué aparece ese mensaje?

Aparece un error 400 debido a que se intenta acceder mediante el protocolo http en lugar de https.

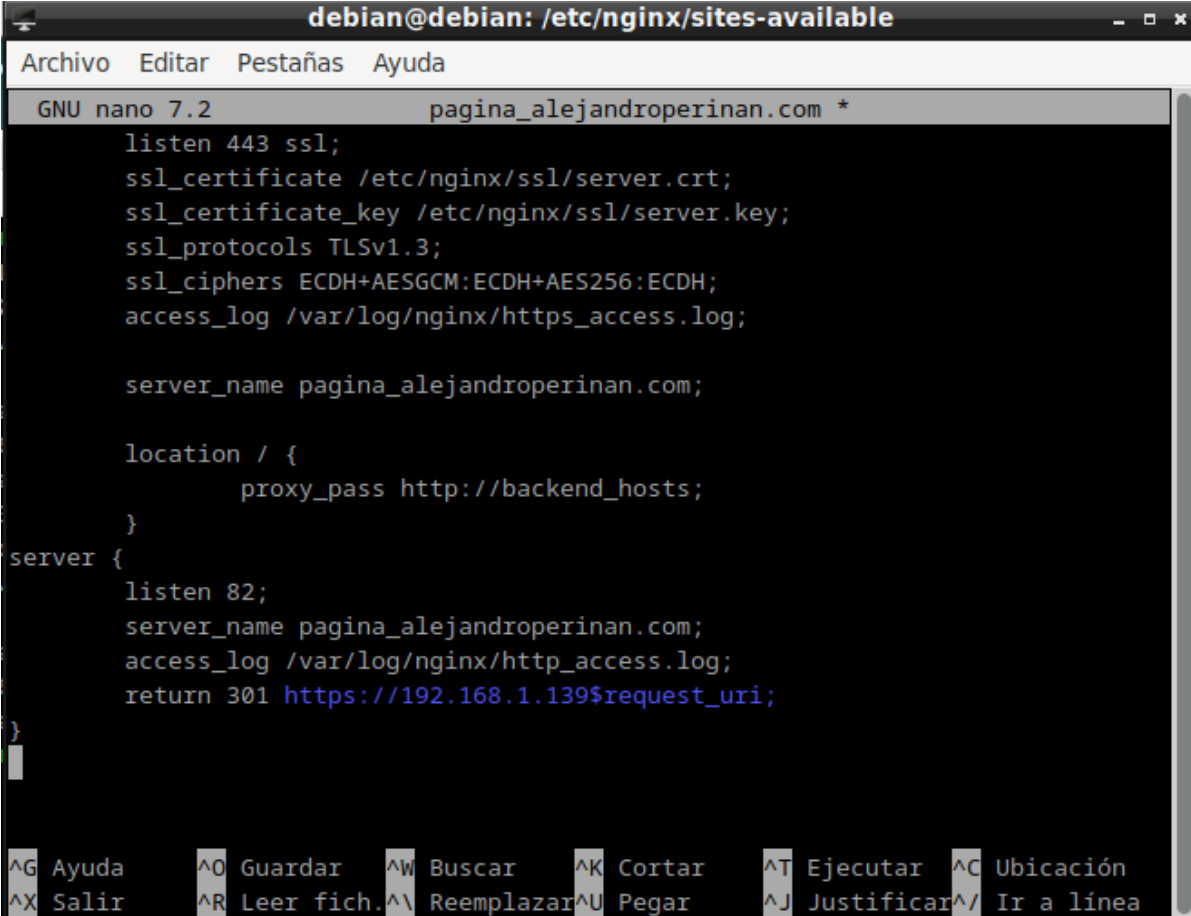
7. Accede al certificado desde el navegador del cliente y muestra una captura de éste.

10.1.4.202:443

92:C3:A9:F1:5D:A1:C2:17:B7:6C:88:B0:18:02:EC:9C:...

8. Vamos a hacer una última modificación al archivo de configuración. El objetivo es que si intentamos acceder a las páginas con HTTP (es decir, sin cifrado), el proxy nos redirija automáticamente a la versión segura. Añade lo siguiente al archivo de configuración:

```
server {  
    listen 82;  
    server_name 192.168.18.20;  
    access_log /var/log/nginx/http_access.log;  
    return 301 https://192.168.18.20$request_uri;  
}
```



The screenshot shows a terminal window titled "debian@debian: /etc/nginx/sites-available". Inside, the nano 7.2 editor is open, editing a file named "pagina_alejandroperinan.com *". The configuration file contains the following content:

```
listen 443 ssl;  
ssl_certificate /etc/nginx/ssl/server.crt;  
ssl_certificate_key /etc/nginx/ssl/server.key;  
ssl_protocols TLSv1.3;  
ssl_ciphers ECDH+AESGCM:ECDH+AES256:ECDH;  
access_log /var/log/nginx/https_access.log;  
  
server_name pagina_alejandroperinan.com;  
  
location / {  
    proxy_pass http://backend_hosts;  
}  
  
server {  
    listen 82;  
    server_name pagina_alejandroperinan.com;  
    access_log /var/log/nginx/http_access.log;  
    return 301 https://192.168.1.139$request_uri;  
}
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts for nano editor operations.

9. Indica qué significa el código HTTP 301. ¿Qué hace la última línea de la imagen?

HTTP 301 indica que la dirección fue movida a otro sitio.

Redirigir a la página que se ponga detrás del return 301.

Incluye capturas de todas las configuraciones realizadas, indicando a qué corresponde cada una. Responde a todas las preguntas planteadas.