GUIA NO FALLO RETO 1

INDEX

FIREWALL

Configuración de UFW

Para configurar el UFW para abrir los puertos lo primero que tenemos que hacer es instalar el firewall con el comando **apt install UFW**.

Para abrir los puertos que nos interesan, en este caso serían el 22 (SSH), el 80 (HTTP) y el 443 (HTTPS), usaremos los comandos:

- **ufw allow 'Nginx Full'** para el 80 y el 443
- **ufw allow 'Nginx HTTP**' para el 80
- **ufw allow 'Nginx HTTPS'** para el 443
- **ufw allow 'OpenSSH'** para el 22

```
root@debian:~# ufw allow 'Nginx Full'
Rules updated
Rules updated (v6)
root@debian:~# ufw allow 'Nginx HTTP'
Rules updated
Rules updated (v6)
root@debian:~# ufw allow 'Nginx HTTPS'
Rules updated
Rules updated (v6)
root@debian:~# ufw allow 'OpenSSH'
Rules updated
Rules updated (v6)
```

Usamos el comando **ufw status** para comprobar que los puertos estan abiertos.

```
root@debian:~# ufw status
Status: active

To                         Action      From
--                         ------      ----
Nginx Full                 ALLOW       Anywhere
Nginx HTTP                 ALLOW       Anywhere
Nginx HTTPS                ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
Nginx Full (v6)            ALLOW       Anywhere (v6)
Nginx HTTP (v6)            ALLOW       Anywhere (v6)
Nginx HTTPS (v6)           ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)
```

Si el output del status es como el siguiente:

```
root@debian:~# ufw status
Status: inactive
```

Utiliza el comando **ufw enable** para activarlo.

```
root@debian:~# ufw enable
Firewall is active and enabled on system startup
```

SFTP

Configuración y preparación de usuarios, permisos y carpetas

1. Usuarios

Para crear un usuario con posibilidad de almacenar ssh-keys ha de tener directorio en /home/ para ello haremos **useradd -m "usuario"** y seguidamente un **passwd "usuario"**

```
root@hosting:~# useradd -m alumnat
root@hosting:~# passwd alumnat
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
root@hosting:~#
```

2. Carpetas a crear

Las carpetas necesarias a crear son las de /var/www/ éstas se crearán con el nombre del dominio escogido por el cliente ej**. alumnat.com** en este caso crearíamos la carpeta **alumnat.com** con subcarpeta **html,** con un comando como
mkdir -p /var/www/alumnat.com/html

```
root@hosting:~# mkdir -p /var/www/alumnat.com/html
root@hosting:~#
```

3. Permisos de carpetas

Estas carpetas recién creadas han de tener ciertos permisos estos son:

- Carpeta **alumnat.com** usuario propietario root
- Carpeta **alumnat.com** grupo propietario el nombre del usuario asignado
- Permisos de **alumnat.com** 755 para evitar broken pipe
  (el primer 5 debe ser 5 ya que elimina la escritura para el grupo propietario, alumnat en este caso, si pusieramos 775 dariamos escritura a grupo alumnat y surgiría broken pipe ya que es la carpeta chroot)

```
root@hosting:~# chown root:alumnat /var/www/alumnat.com/
root@hosting:~#
root@hosting:~# chmod 755 /var/www/alumnat.com/
root@hosting:~#
```

- Subcarpeta **html** usuario propietario root
- Subcarpeta **html** grupo propietario el nombre del usuario asignado
- Permisos de subcarpeta html ha de ser 775 ya que el usuario sftp ha de poder tener escritura dentro de la subcarpeta ya que en la carpeta chroot no se puede tener escritura (broken pipe)

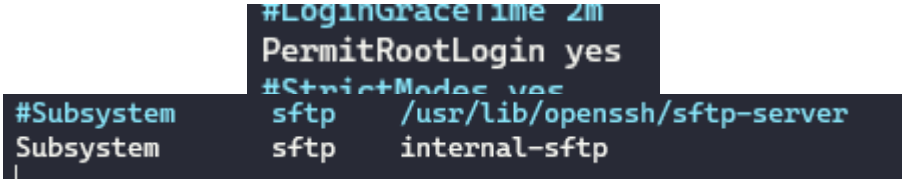
```
root@hosting:~# chown root:alumnat /var/www/alumnat.com/html/
root@hosting:~#
root@hosting:~# chmod 775 /var/www/alumnat.com/html/
root@hosting:~#
```

Configuración sshd_config

1. Configuración superficial sobre el archivo sshd_config

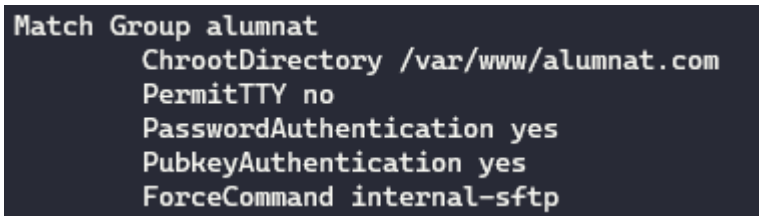
Para entrar a editar el archivo hay que hacer el comando **nano /etc/ssh/sshd_config**

- Primero establecemos el **PermitRootLogin** como **yes**
- Cambiaremos el **Subsystem sftp /usr/../..** a **Subsystem sftp internal-sftp**



2. Reconfiguración del sshd_config

Añadimos la configuración para enjaular un grupo en un ChrootDirectory:
(es importante enjaular antes de la subcarpeta **html,** de momento ponemos passwordauth en yes mas adelante ya la quitaremos para el publickeyauth)



Configuración de Public Key Auth

Tendremos que acceder al cliente para poder hacer la gran mayoría de este proceso

1. Parte Servidor **1**

- Accederemos al sshd_config y comentamos la conf jail entera
- Guardamos y **systemctl restart sshd**

```
#Match Group alumnat
#       ChrootDirectory /var/www/alumnat.com
#       PermitTTY no
#       PasswordAuthentication yes
#       PubkeyAuthentication yes
#       ForceCommand internal-sftp
root@hosting:~# systemctl restart sshd
root@hosting:~#
```

2. Parte de Cliente **1**

- Crearemos un par de keys con el comando **ssh-keygen**
- Haremos ssh-copy-id alumnat@ip-o-dominio(recordar el dominio se asigna en /etc/hosts)

```
root@debian:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:srXPLfsk5MwdywujicEXmDAH4AjpsR9RRSjQDRQPbTs root@debian
The key's randomart image is:
+---[RSA 3072]----+
|.o=*B.+o         |
|o+ +o*           |
|o + *.o          |
| o . E o         |
|  . . = S . .    |
|   . . + B o o   |
|      + o B =    |
|       + =.* .   |
|      . o +++    |
+----[SHA256]-----+
root@debian:~#
```

```
root@debian:~# ssh-copy-id alumnat@alumnat.com
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'alumnat.com (192.168.0.100)' can't be established.
ED25519 key fingerprint is SHA256:lkoTQeTptmFy9r/gZsI/torvYL2Z+HIwhoYMCrRVoHQ.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out a
 that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted no
it is to install the new keys
alumnat@alumnat.com's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'alumnat@alumnat.com'"
and check to make sure that only the key(s) you wanted were added.

root@debian:~#
```

3. Parte Servidor **2**

- Descomentar la jail y denegar PasswordAuth y permitir PubkeyAuth(restart sshd)

```
Match Group alumnat
        ChrootDirectory /var/www/alumnat.com
        PermitTTY no
        PasswordAuthentication no
        PubkeyAuthentication yes
        ForceCommand internal-sftp
root@hosting:~# systemctl restart sshd
root@hosting:~#
```

4. Parte Cliente **2**

- Probamos la conexión con Publickey

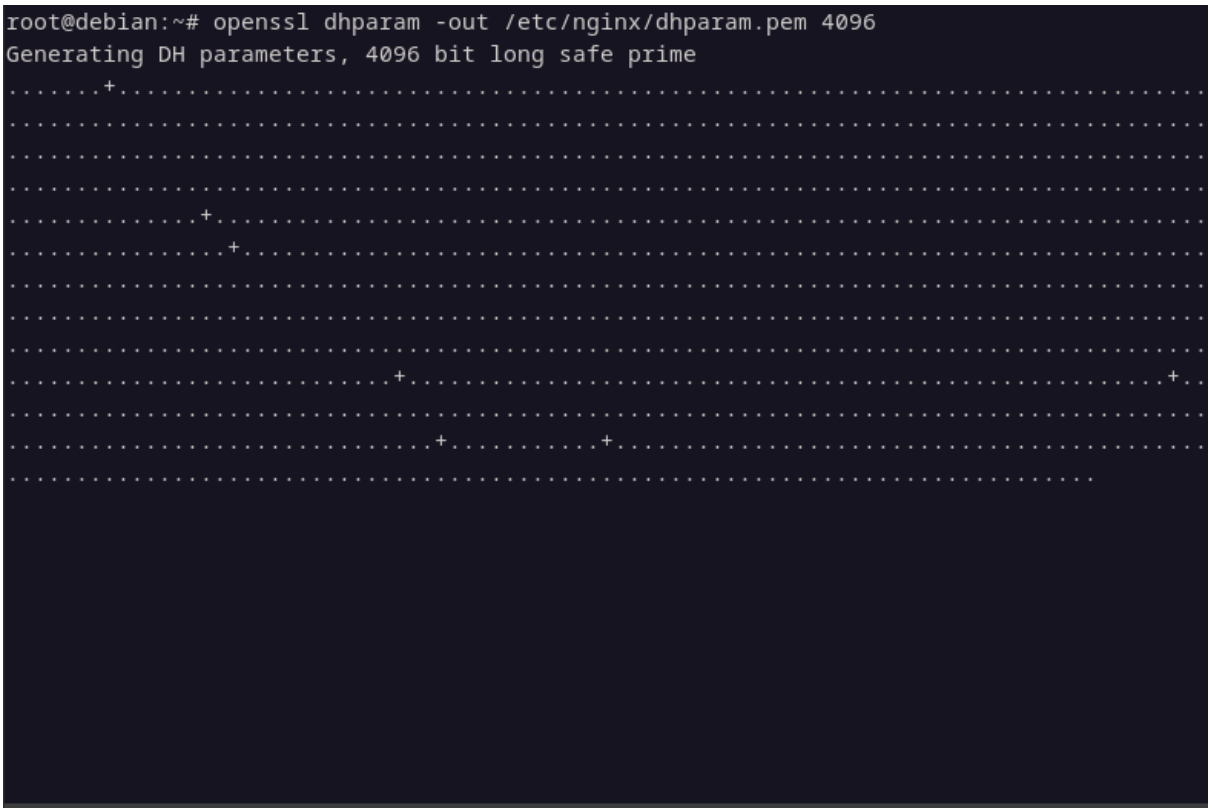
```
root@debian:~# sftp alumnat@alumnat.com
Connected to alumnat.com.
sftp> cd html/
sftp> put index.html
Uploading index.html to /html/index.html
index.html                                    100%    0     0.0KB/s   00:00
sftp>
```

SSL

Creación de DH, SSL-CRT y Snippets

Para crear la configuración base del cifrado SSL, lo primero que vamos a necesitar es crear el grupo y el archivo de parámetros de DH, para hacer eso usaremos el comando:

- **openssl dhparam -out /etc/nginx/dhparam.pem 4096**



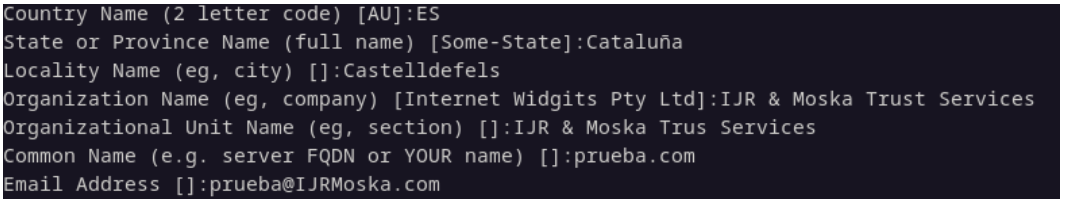
Este proceso puede tardar un rato, espera hasta que acabe por completo.

Para configurar el SSL tendremos que crear un certificado (CRT) y una clave (KEY), para crear ambas usaremos el comando:
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/nginx-selfsigned.key -out /etc/ssl/certs/nginx-selfsigned.crt

Donde editaremos el directorio y nombre de la key y el crt para poder diferenciarlas entre si, es decir:
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/alumnat.com.key -out /etc/ssl/certs/alumnat.com.crt

Lo subrayado es un ejemplo de lo que podemos editar.

Después de ejecutar este comando tendremos que responder unas preguntas:



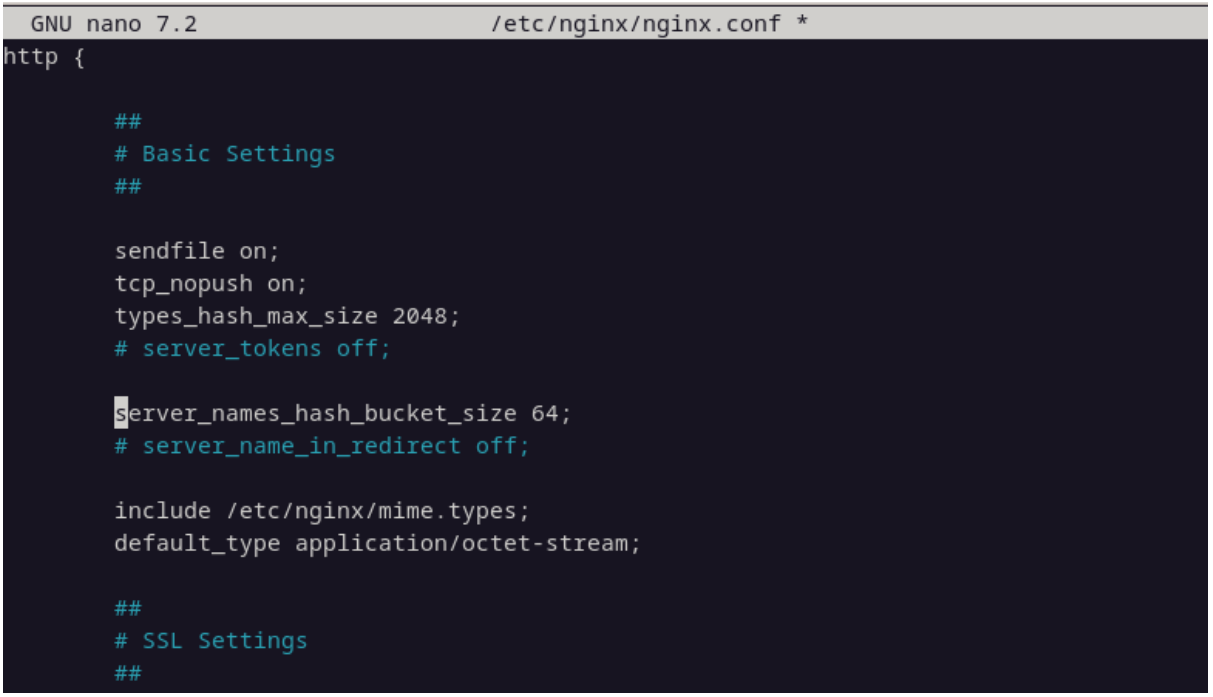
Lo siguiente es la configuración del snippets, usamos el comando **nano /etc/nginx/snippets/ssl-params.conf**
Una vez dentro de la configuración, escribimos las siguientes líneas:

ssl_protocols TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_dhparam /etc/nginx/dhparam.pem;
ssl_ciphers
ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AE
S256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384;
ssl_ecdh_curve secp384r1; # Requires nginx >= 1.1.0
ssl_session_timeout 10m;
ssl_session_cache shared:SSL:10m;
ssl_session_tickets off; # Requires nginx >= 1.5.9
ssl_stapling on; # Requires nginx >= 1.3.7
ssl_stapling_verify on; # Requires nginx => 1.3.7
resolver 8.8.8.8 8.8.4.4 valid=300s;
resolver_timeout 5s;
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";

VHOSTS w/ SSL

Correcta configuracion del nginx.conf

Para configurar el servidor nginx para que funcione perfectamente, lo primero que vamos a hacer es abrir el archivo de configuración con el comando **nano /etc/nginx/nginx.conf** y descomentamos la linea que ponga **server_names_hash_bucket_size 64;**



Configuración de VHOSTS en sites-available y enabled

Para configurar VHOSTS sin SSL, lo primero que vamos ha hacer es crear un archivo en **/etc/nginx/sites-available/** dentro de ese archivo de configuración vamos a escribir las siguientes líneas:

```
server {
        listen 80;
        listen [::]:80;

        root /var/www/tudominio.com/html;
        index index.html index.htm index.nginx-debian.html;

        server_name alumnat.com www.alumnat.com;

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

Estas líneas están creando un servidor que escuche por el puerto 80 (HTTP) y redirige el tráfico de alumnat.com a /var/wwwalumnat.com/html/index.html.

Este archivo una vez configurado tendrás que copiar el archivo de configuración de **/etc/nginx/sites-available/** a **/etc/nginx/sites-enabled/**

En el caso que quieras un servidor con encriptación SSL tendrás que crear un archivo con las siguientes líneas:

```
server {
        listen 443 ssl;
        listen [::]:443 ssl;
        ssl_certificate /etc/ssl/certs/alumnat.com.crt;
        ssl_certificate_key /etc/ssl/private/alumnat.com.key;
        include snippets/ssl-params.conf;

        root /var/www/alumnat.com/html;
        index index.html index.htm index.nginx-debian.html;

        server_name alumnat.com www.alumnat.com;

        location / {
                try_files \$uri \$uri/ =404;
        }
    }
    server {
        listen 80;
        listen [::]:80;

        server_name alumnat.com www.alumnat.com;

        return 302 https://\$server_name\$request_uri;
}
```