Práctica 4

Los principales objetivos de la práctica son dos fundamentalmente, instalar los certificados SSL para el acceso a los servidores y configurar las reglas de nuestro firewall para proteger nuestra granja web. Los pasos pertinentes para alcanzar dichos objetivos son los siguientes:

Certificados SSL

Iniciamos las 3 máquinas virtuales, 1 que representa el balanceador de carga y las otras dos que son el servidor objetivo del acceso. Dicho balanceador de carga apropiadamente configurado con **nginx** resultado de la práctica anterior.

```
sudo a2enmod ssl
sudo service apache2 restart
sudo mkdir /etc/apache2/ssl
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
```

Como resultado del último comando se nos pedirá una serie de datos referentes a la configuración del certificado:

```
root@UbuntuServer:/home/laguilarg99# sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyou
t /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
Generating a 2048 bit RSA private key
.....................................................................................+++
.....................................................+++
writing new private key to '/etc/apache2/ssl/apache.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) [AU]:ES
State or Province Name (full name) [Some-State]:Granada
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SWAP
Organizational Unit Name (eg, section) []:P4
Common Name (e.g. server FQDN or YOUR name) []:laguilarg99
Email Address []:laguilarg99@correo.ugr.es
```

Ahora editaremos el archivo de configuración del sitio **default-ssl.conf,**

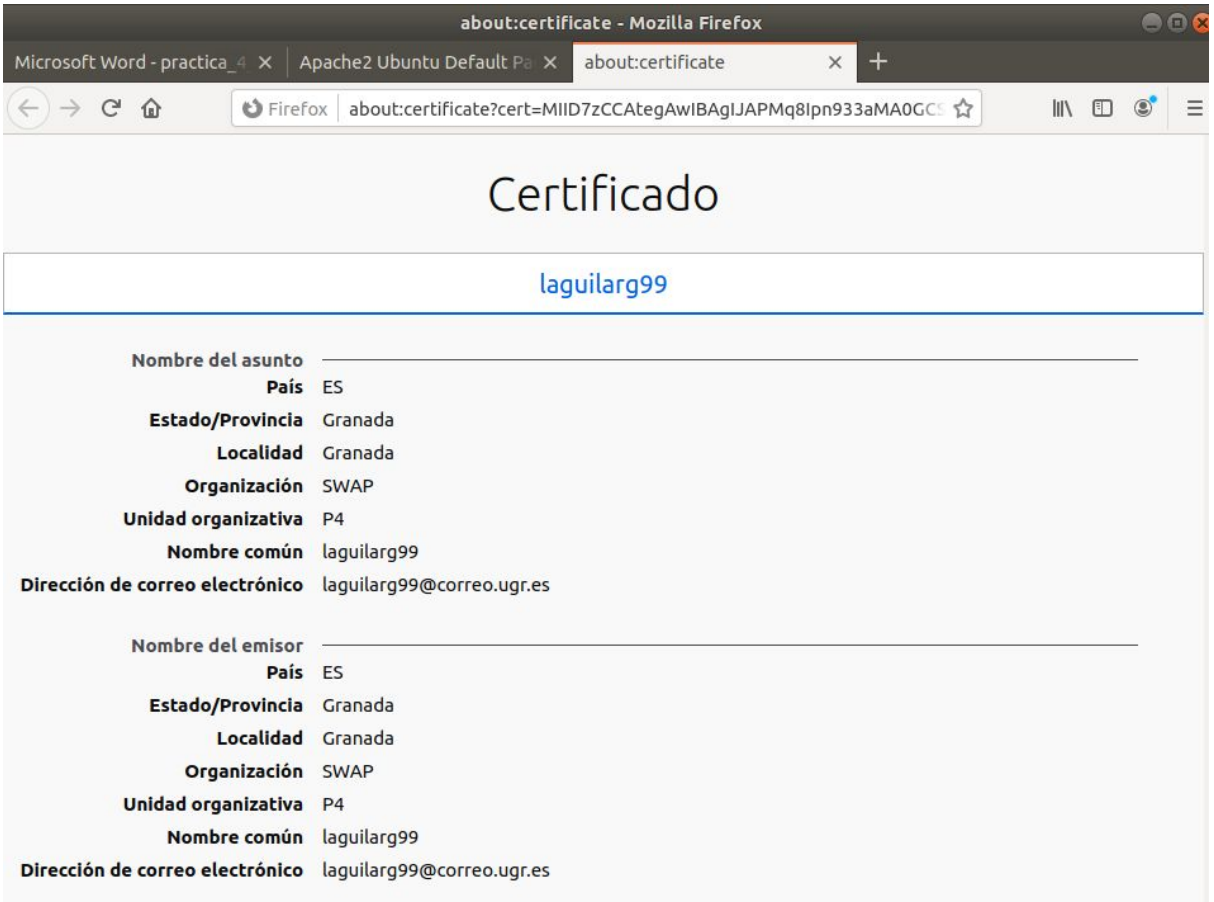
```
        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2/README.Debian.gz for more info.
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/apache2/ssl/apache.crt
        SSLCertificateKeyFile /etc/apache2/ssl/apache.key_
```

Cargamos la nueva configuración y reiniciamos Apache,

```
root@UbuntuServer:/home/laguilarg99# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@UbuntuServer:/home/laguilarg99# service apache2 reload
```

Para comprobar el funcionamiento apropiado del certificado instalados accederemos mediante **HTTPS** en el navegador web a nuestro servidor:

Como se puede observar el sitio web no es seguro pues el certificado ha sido expedido por nosotros mismos y no por una autoridad certificadora, dicho certificado es el siguiente:



Repetiré el proceso hasta ahora mencionado en máquina que resta copiando los certificados.

El siguiente paso es hacer que el balanceador de carga también sea capaz de redirigir el tráfico propio de las comunicaciones HTTPS, cambiando la configuración de nginx **/etc/nginx/conf.d/default.conf** y añadiendo:

```
server{

        listen 443;
        server_name balanceadorHTTPS;

        ssl on;
        ssl_certificate /home/laguilarg99/ssl/apache.crt;
        ssl_certificate_key /home/laguilarg99/ssl/apache.key;

        access_log /var/log/nginx/balanceadorHTTPS.access.log;
        error_log /var/log/nginx/balanceadorHTTPS.error.log;
        root /var/www/;

        location /
        {
                proxy_pass http://servidoresSWAP;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_http_version 1.1;
                proxy_set_header Connection “”;
        }
}
```

Cortafuegos

Para la correcta realización de la práctica hay que conseguir dos objetivos muy concretos, que M3 solo acepte peticiones HTTP/HTTPS y por otro lado M1 y M2 solo pueden aceptar peticiones de M3, para lograrlos hay que crear un script que sea ejecutado con el arranque del sistema para que el cortafuegos se configure adecuadamente.
M3, la máquina encargada del balanceo de carga, tendrá una configuración del cortafuegos básica de un servidor web, es decir, sólo aceptará comunicaciones HTTP/HTTPS:

```
# (1) Eliminar todas las reglas (configuración limpia)
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

# (2) Política por defecto: denegar todo el tráfico entrante
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT

# (3) Permitir cualquier acceso desde localhost (interface lo)
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# (4) Abrir el puerto 22 para permitir el acceso por SSH
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT

# Permitir el tráfico por el puerto 80 (HTTP)
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT

# (6) Permitir el tráfico por el puerto 443 (HTTPS)
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 443 -j ACCEPT

iptables -L -n -v
```

EL script referente a las máquinas M1/M2 permitirá comunicaciones HTTP/HTTPS solo entre estas y M3.

```
# (1) Eliminar todas las reglas (configuración limpia)
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

# (2) Política por defecto: denegar todo el tráfico entrante
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP

# (3) Permitir cualquier acceso desde localhost (interface lo)
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# (4) Abrir el puerto 22 para permitir el acceso por SSH
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT

# Permitir el tráfico por el puerto 80 (HTTP)
iptables -A INPUT -p tcp --dport 80 -s 192.168.56.103  -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 -s 192.168.56.103 -j ACCEPT

# (6) Permitir el tráfico por el puerto 443 (HTTPS)
iptables -A INPUT -p tcp --dport 443 -s 192.168.56.103 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 443 -s 192.168.56.103 -j ACCEPT

iptables -L -n -v
```

El resultado del script en M3:

```
root@UbuntuServer:/home/laguilarg99# ./scriptIPTABLES.sh
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0           state NEW,E
STABLISHED
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:443

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:22
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:80
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp spt:443
```

El resultado del script en M1/M2:

```
root@UbuntuServer:/home/laguilarg99# ./scriptIPTABLES.sh
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     tcp  --  *      *       192.168.56.103       0.0.0.0/0           tcp dpt:80
    0     0 ACCEPT     tcp  --  *      *       192.168.56.103       0.0.0.0/0           tcp dpt:443

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  *      lo      0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     tcp  --  *      *       192.168.56.103       0.0.0.0/0           tcp spt:80
    0     0 ACCEPT     tcp  --  *      *       192.168.56.103       0.0.0.0/0           tcp spt:443
```

Como resultado de esta configuración desde el navegador anfitrión no podremos acceder a las máquinas M1/M2 directamente, tendremos que acceder desde M3 para nos devuelva el contenido de las máquinas M1/M2.

Por otro lado para que el script se ejecute cada vez que iniciamos el sistema, añadimos permisos de ejecución al archivo /etc/rc.local y escribimos la línea que se especifica en la siguiente captura:

chmod +x /etc/rc.local