Practica 4

Autor: Antonio Jesús Heredia Castillo

Configuración SSL

Aunque en el guión pide que se haga en **M1**, me di cuenta que lo hice primero en **M2**. No obstante el proceso es exactamente igual. Ademas como me encargo de copiar las claves tanto a **M1** y **M3** el resultado final es el mismo.
Lo primero que hacemos es activar el modulo SSL de apache, reiniciarlo y crear la capreta donde vamos a guardar los ficheros.

```
antoni-heredia@m2:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
antoni-heredia@m2:~$ sudo systemctl restart apache2
antoni-heredia@m2:~$ sudo mkdir /etc/apache2/ssl
antoni-heredia@m2:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/s
sl/apache.key -out /etc/apache2/ssl/apache.crt
Can't load /home/antoni-heredia/.rnd into RNG
139699448943040:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/ran
d/randfile.c:88:Filename=/home/antoni-heredia/.rnd
Generating a 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]:
```

Cuando estemos generando el certificado introduciremos los siguientes datos:

```
antoni-heredia@m2:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/s
sl/apache.key -out /etc/apache2/ssl/apache.crt
Can't load /home/antoni-heredia/.rnd into RNG
140575389426112:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/ran
d/randfile.c:88:Filename=/home/antoni-heredia/.rnd
Generating a 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) []:antonioheredia
Email Address []:antonioheredia@correo.ugr.es
antoni-heredia@m2:~$ _
```

Una vez generado el certificado tenemos que indicar a apache donde se encuentra el mismo, ya que la ruta que trae por defecto no es la de nuestro certificado:

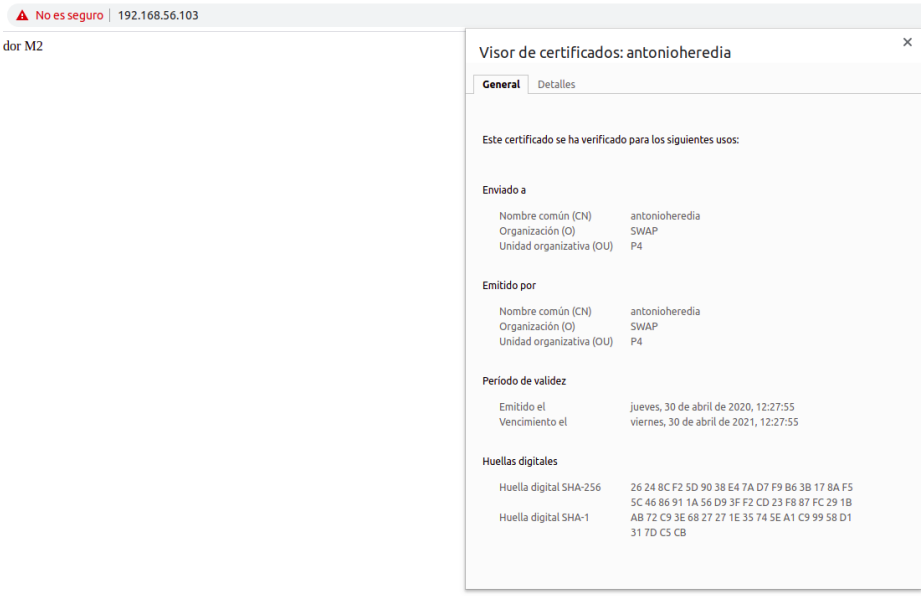
```
#   /usr/share/doc/apache2/README.Debian.gz for more info.
#   If both key and certificate are stored in the same file, only the
#   SSLCertificateFile directive is needed.
SSLCertificateFile      /etc/apache2/ssl/apache.crt
SSLCertificateKeyFile   /etc/apache2/ssl/apache.key_

#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
```

Ahora solo tenemos que activar el SLL por defecto y recargar la configuración:

```
antoni-heredia@m2:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
antoni-heredia@m2:~$ sudo systemctl reload apache2
antoni-heredia@m2:~$ _
```

Cuando entramos con el navegador nos dice si “confiamos en el sitio” tendremos que decir que si, esto se debe a que el certificado no esta firmado por alguna empresa reconocida. Y al ver el certificado podemos ver que son los datos que nosotros añadimos.



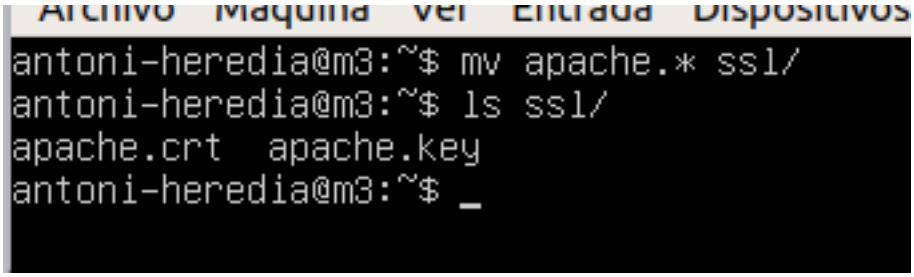
Ahora para poder tener el mismo certificado en **M1** y **M3**, haremos uso de scp. Con ella copiaremos el certificado y la clave a ambas maquinas como se puede ver en la siguiente imagen:

```
antoni-heredia@m2:~$ sudo scp  /etc/apache2/ssl/* antoni-heredia@192.168.56.101:~/.
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ECDSA key fingerprint is SHA256:jEQ8dX+zUskbOiTdw9S27pz2KlecT6GZuFpO6d8k+Mg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.101' (ECDSA) to the list of known hosts.
antoni-heredia@192.168.56.101's password:
apache.crt                                    100% 1460   752.8KB/s   00:00
apache.key                                    100% 1704   975.4KB/s   00:00
antoni-heredia@m2:~$ sudo scp  /etc/apache2/ssl/* antoni-heredia@192.168.56.103:~/.
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established.
ECDSA key fingerprint is SHA256:jEQ8dX+zUskbOiTdw9S27pz2KlecT6GZuFpO6d8k+Mg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.56.103' (ECDSA) to the list of known hosts.
antoni-heredia@192.168.56.103's password:
apache.crt                                    100% 1460   689.2KB/s   00:00
apache.key                                    100% 1704   918.1KB/s   00:00
antoni-heredia@m2:~$
```

En la maquina **M1** deberemos mover ambos ficheros al siguiente directorio:

```
antoni-heredia@m1:~$ sudo mv apache.* /etc/apache2/ssl/
antoni-heredia@m1:~$ ls /etc/apache2/ssl/
apache.crt  apache.key
antoni-heredia@m1:~$ _
```

En el caso de la maquina **M3** lo pondremos en el siguiente directorio:



Podemos ver que si intentamos acceder a la maquina **M3** antes de la configuración, aunque **M1** y **M2** esten ya configurados para usar SSL, no nos va a dejar acceder.

En la maquina **M1** el proceso para configurar Apache es el mismo que el que hemos descrito anteriormente. En el caso de **M3** como usa NGINX para balancear la carga si cambia la configuración. Tendremos que modificar el fichero "/etc/nginx/conf.d/default.conf" y añadir un servidor nuevo. Quedando la configuración algo tal que asi:

```
server{
        listen 443 ssl;
        ssl on;
        ssl_certificate /home/antoni-heredia/ssl/apache.crt;
        ssl_certificate_key /home/antoni-heredia/ssl/apache.key;
        server_name balanceador;
        access_log /var/log/nginx/balanceador.access.log;
        error_log /var/log/nginx/balanceador.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 "";
        }
}
```

Indicamos que escuche en el puerto 443, el por defecto para usar HTTPs, y donde están ubicados los ficheros del certificado. Ahora lo unico que nos queda es reiniciar NGINX para que coja la nueva configuración y listo.

```
antoni-heredia@m3:~$ sudo systemctl restart nginx
antoni-heredia@m3:~$ _
```

Ahora ya podemos acceder tanto a **M1** como a **M2** desde la maquina **M3**.

Cortafuegos

Primero, como pide en la practica vamos a realizar una configuración muy básica con IPTABLES en la maquina **M1**. Solo permitiremos el acceso web por el puerto 80 y 430 y ademas el acceso por SSH, ya que ese puerto también se suele dejar abierto para poder configurar la maquina de forma remota. Para ello usaremos el siguiente script:

```
 GNU nano 2.9.3                          cortafuego.sh

#eliminamos reglas preexistentes
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
# denegamos todo el trafico por defecto
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT
#permito cualquier acceso desde el localhost
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -i lo -j ACCEPT
#permitimos el puerto 22 para ssh
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT
#permitirmos trafico por el puerto 80
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
#permitimos trafico por el puerto 443
iptables -A INPUT -p tcp --dport 443 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT

iptables -L -n -v
```

Le damos permisos de ejecución al script y lo ejecutamos como administrador:

```
antoni-heredia@m1:~$ chmod 755 cortafuego.sh
antoni-heredia@m1:~$ sudo ./cortafuego.sh
iptables v1.6.1: Can't use -i with OUTPUT

Try `iptables -h' or 'iptables --help' for more information.
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 DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    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
antoni-heredia@m1:~$ _
```

Y podemos ver si esta funcionando bien con el comando netstat:

```
antoni-heredia@m1:~$ netstat -tulpn
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp6       0      0 :::443                  :::*                    LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
udp        0      0 127.0.0.53:53           0.0.0.0:*                           -
antoni-heredia@m1:~$
```

Para el extra que se pide en la practica lo he conseguido a medias. Ya si he realizado que **M1** y **M2** reciban solo datos desde **M3** que es el balanceador, pero no he conseguido que el balanceador, recibiendo peticiones solo por el puerto 80 o 443, pueda servir las paginas web de las otras dos maquinas.

Por lo tanto lo que he realizado es que **M3** reciba y envíe a todas las maquinas

con el siguiente script:

```
 GNU nano 2.9.3                          cortafuego_defecto.sh

#eliminamos reglas preexistentes
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT

iptables -L -n -v
```

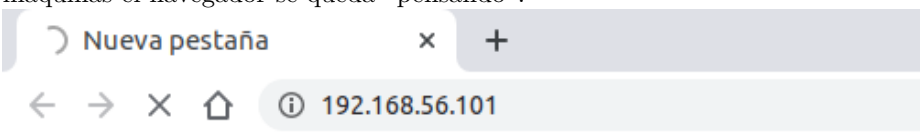
Y para que **M1** y **M2** solo reciban desde **M3** el script es el siguiente:

```
 GNU nano 2.9.3                              cortafuego.sh

#eliminamos reglas preexistentes
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
# denegamos todo el trafico por defecto
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT
#permito cualquier acceso desde el localhost
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
#permitimos el puerto 22 para ssh
iptables -A INPUT -p tcp --dport 22 -s 192.168.56.103 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -d 192.168.56.103  -j ACCEPT
#permitirmos trafico por el puerto 80
iptables -A INPUT -p tcp --dport 80 -s 192.168.56.103 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 -d 192.168.56.103 -j ACCEPT
#permitimos trafico por el puerto 443
iptables -A INPUT -p tcp --dport 443 -s 192.168.56.103 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 443 -d 192.168.56.103 -j ACCEPT


iptables -L -n -v
```

Y como podemos ver si intentamos acceder directamente a cualquiera de las maquinas el navegador se queda “pensando”:



Y en cambio si intentamos acceder desde el balanceador funciona perfectamente: