Servidores Web de Altas Prestaciones

Práctica 4

Autor: Juan Ocaña Valenzuela

En esta práctica se deben cumplir los siguientes objetivos obligatorios:

- Crear e instalar en la máquina M1 un certificado SSL autofirmado para configurar el acceso HTTPS al servidor. Se debe comprobar que el servidor acepta tanto el tráfico HTTP como el HTTPS.
- Configurar las reglas del cortafuegos con IPTABLES en uno de los servidores finales para asegurarlo, permitiendo el acceso por los puertos HTTP y HTTPS a dicho servidor.

Además, se proponen los siguientes objetivos opcionales:

- Una vez configurada la máquina M1, copiar el certificado autofirmado al resto de máquinas servidoras y al balanceador, y configurar nginx adecuadamente para aceptar y balancear tráfico HTTP y HTTPS.
- Configurar M3 estableciendo las reglas de IPTABLES para que únicamente M3 acepte peticiones HTTP y HTTPS mientras que M1 y M2 no acepten nada a no ser que provenga de M3. Hacer que la configuración del cortafuegos se ejecute al arrancar el sistema.

Crear e instalar en la máquina M1 un certificado SSL autofirmado para configurar el acceso HTTPS al servidor.

Vamos a configurar un certificado SSL autofirmado en M1. Para ello, activamos el módulo SSL de Apache con `sudo a2enmod ssl` y reiniciamos el servicio:

```
patchispatch@m1:~$ sudo a2enmod ssl
[sudo] password for patchispatch:
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certifica
tes.
To activate the new configuration, you need to run:
  systemctl restart apache2
patchispatch@m1:~$ sudo systemctl restart apache2
patchispatch@m1:~$
```

Ahora generamos los certificados. Para tenerlos a mano, los guardaremos en `/etc/apache2/ssl`, y utilizaremos openssl:

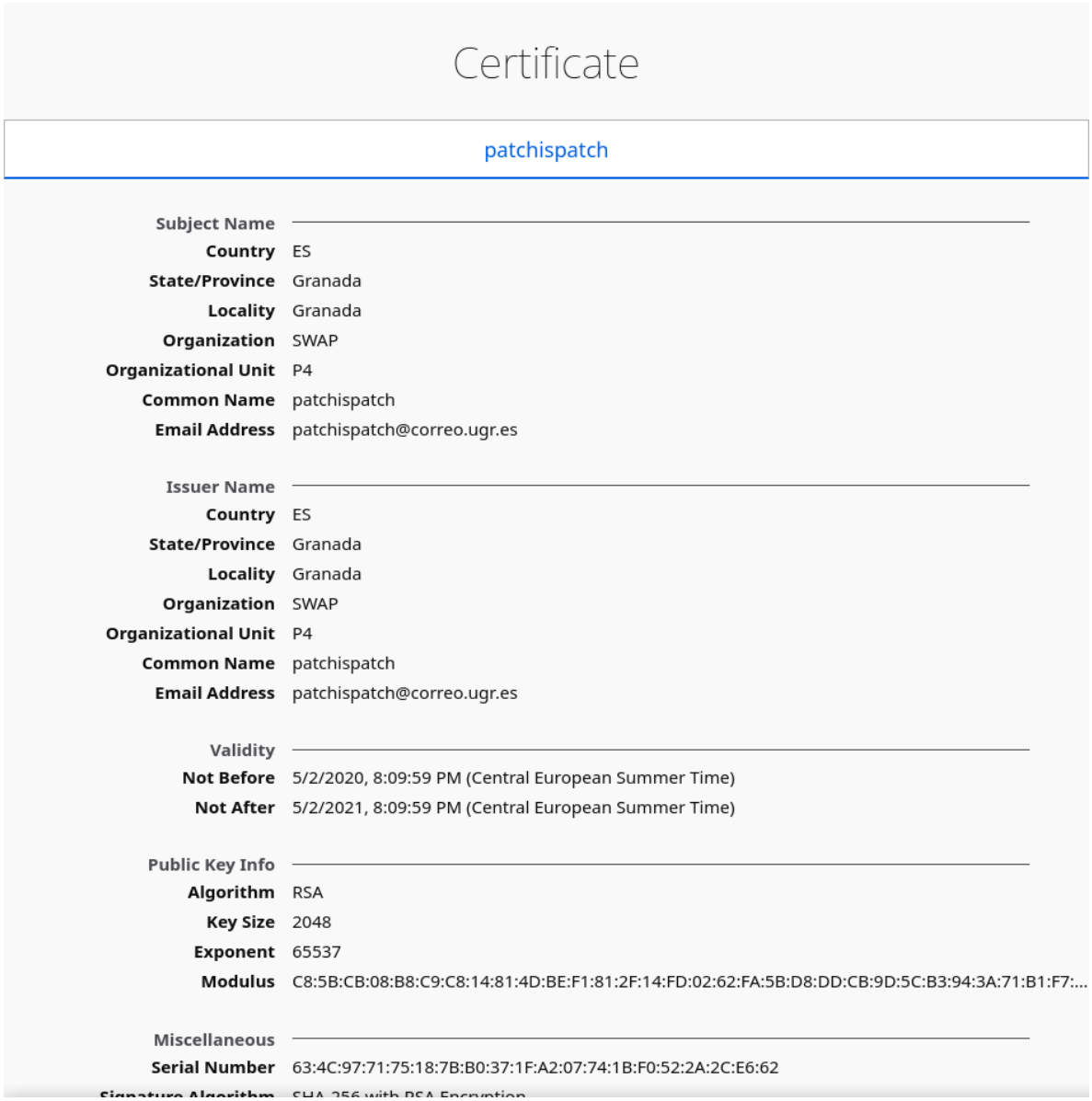
```
patchispatch@m1:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl
/apache.key -out /etc/apache2/ssl/apache.crt
Can't load /home/patchispatch/.rnd into RNG
140125052330432:error:2406F079:random number generator:RAND_load_file:Cannot open file:../crypto/ran
d/randfile.c:88:Filename=/home/patchispatch/.rnd
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) []:patchispatch
Email Address []:patchispatch@correo.ugr.es
patchispatch@m1:~$ _
```

Ahora debemos añadir la ruta de los certificados en el archivo de configuración de SSL de Apache, activarlo y reiniciar el servicio para hacer efectivos los cambios:

```
                DocumentRoot /var/www/html

                # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
                # error, crit, alert, emerg.
                # It is also possible to configure the loglevel for particular
                # modules, e.g.
                #LogLevel info ssl:warn

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                # For most configuration files from conf-available/, which are
                # enabled or disabled at a global level, it is possible to
                # include a line for only one particular virtual host. For example the
                # following line enables the CGI configuration for this host only
                # after it has been globally disabled with "a2disconf".
                #Include conf-available/serve-cgi-bin.conf

                #   SSL Engine Switch:
                #   Enable/Disable SSL for this virtual host.
                SSLEngine on
                SSLCertificateFile /etc/apache2/ssl/apache.crt
                SSLCertificateKeyFile /etc/apache2/ssl/apache.key

                #   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.

                #   Server Certificate Chain:
                #   Point SSLCertificateChainFile at a file containing the
                #   concatenation of PEM encoded CA certificates which form the
                #   certificate chain for the server certificate. Alternatively
                #   the referenced file can be the same as SSLCertificateFile
                #   when the CA certificates are directly appended to the server
"/etc/apache2/sites-available/default-ssl.conf" 134L, 6318C escritos             28,0-1        4%
```

```
patchispatch@m1:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
patchispatch@m1:~$ sudo systemctl reload apache2
patchispatch@m1:~$
```

Ahora vamos a acceder desde un navegador en el host y ver el certificado:



Una vez configurada la máquina M1, copiar el certificado autofirmado al resto de máquinas servidoras y al balanceador, y configurar nginx adecuadamente para aceptar y balancear tráfico HTTP y HTTPS.

Para poder acceder a la granja web mediante el balanceador de carga, debemos copiar el certificado al resto de máquinas.

En M2, puesto que también se trata de un servidor Apache, la configuración es la misma; tan solo tenemos que copiar el certificado mediante, por ejemplo, scp.

```
patchispatch@m1:~$ sudo scp /etc/apache2/ssl/apache.crt patchispatch@192.168.56.101:/home/patchispat
ch/apache.crt
patchispatch@192.168.56.101's password:
apache.crt                                    100% 1448     1.0MB/s   00:00
patchispatch@m1:~$ sudo scp /etc/apache2/ssl/apache.key patchispatch@192.168.56.101:/home/patchispat
ch/apache.key
patchispatch@192.168.56.101's password:
apache.key                                    100% 1704     3.1MB/s   00:00
patchispatch@m1:~$ _
```

En M2, actualizamos la configuración de Apache para que sea idéntica a la de M1. Sin embargo, en M3 debemos configurar Nginx, no Apache, añadiendo un nuevo `server` con los siguientes datos al archivo `/etc/nginx/conf.d/default.conf`:

```
upstream servidoresSWAP {
        server 192.168.56.103;
        server 192.168.56.101;
        keepalive 3;
}

server {
        listen 80;
        server_name balanceador;

        listen 443 ssl;
        ssl on;
        ssl_certificate /home/patchispatch/apache.crt;
        ssl_certificate_key /home/patchispatch/apache.key;

        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 "";
        }
}
~
~
~
~
~
~
~
"/etc/nginx/conf.d/default.conf" 29L, 636C escritos                    14,39-46      Todo
```

Ahora podemos acceder mediante HTTPS a la IP del balanceador, y nos mostrará correctamente (previo aviso de certificado autofirmado) la información correspondiente.

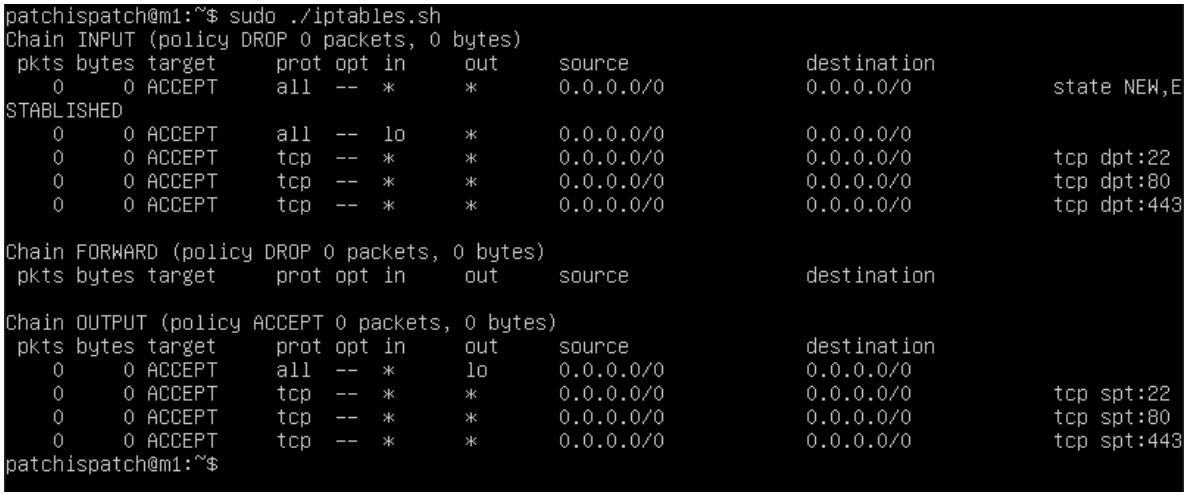
Configurar las reglas del cortafuegos con IPTABLES en uno de los servidores finales para asegurarlo, permitiendo el acceso por los puertos HTTP y HTTPS a dicho servidor.

Vamos a configurar IPTABLES en las máquinas M1 y M2. Para ello, utilizaremos el siguiente script:

```
#!/bin/sh

# (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

# (5) 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 --dport443 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 443 -j ACCEPT

iptables -L -n -v
```

Al ejecutar recibimos la siguiente salida



Configurar M3 estableciendo las reglas de IPTABLES para que únicamente M3 acepte peticiones HTTP y HTTPS mientras que M1 y M2 no acepten nada a no ser que provenga de M3. Hacer que la configuración del cortafuegos se ejecute al arrancar el sistema.

Para permitir que únicamente M3 se comunique con M1 y M2 a través de los puertos 80 y 443 (HTTP y HTTPS), debemos modificar nuestras reglas de iptables de la siguiente forma:

```
# IP permitida
IP=192.168.56.104

# HTTP
iptables -A INPUT -p tcp -s $IP --dport 80 -j ACCEPT
iptables -A OUTPUT -p tcp -s $IP --sport 80 -j ACCEPT

# (6) Permitir el tráfico por el puerto 443 (HTTPS)
iptables -A INPUT -p tcp -s $IP --dport443 -j ACCEPT
iptables -A OUTPUT -p tcp -s $IP --sport 443 -j ACCEPT
```

Para que nuestro script se ejecute al inicio debemos hacer lo siguiente:

1. Mover nuestro script a /etc/init.d/
2. `sudo chmod -x /etc/init.d/iptables.sh`
3. `sudo chown root.root /etc/init.d/iptables.sh`
4. `sudo update-rc.d test.sh defaults`