

Práctica 4

En primer lugar generamos los certificados SSL autofirmados, para ello activamos el módulo SSL de apache, generamos los certificados y les especificamos la ruta de configuración. Ejecutamos:

```
a2enmod ssl
```

```
service apache2 restart
```

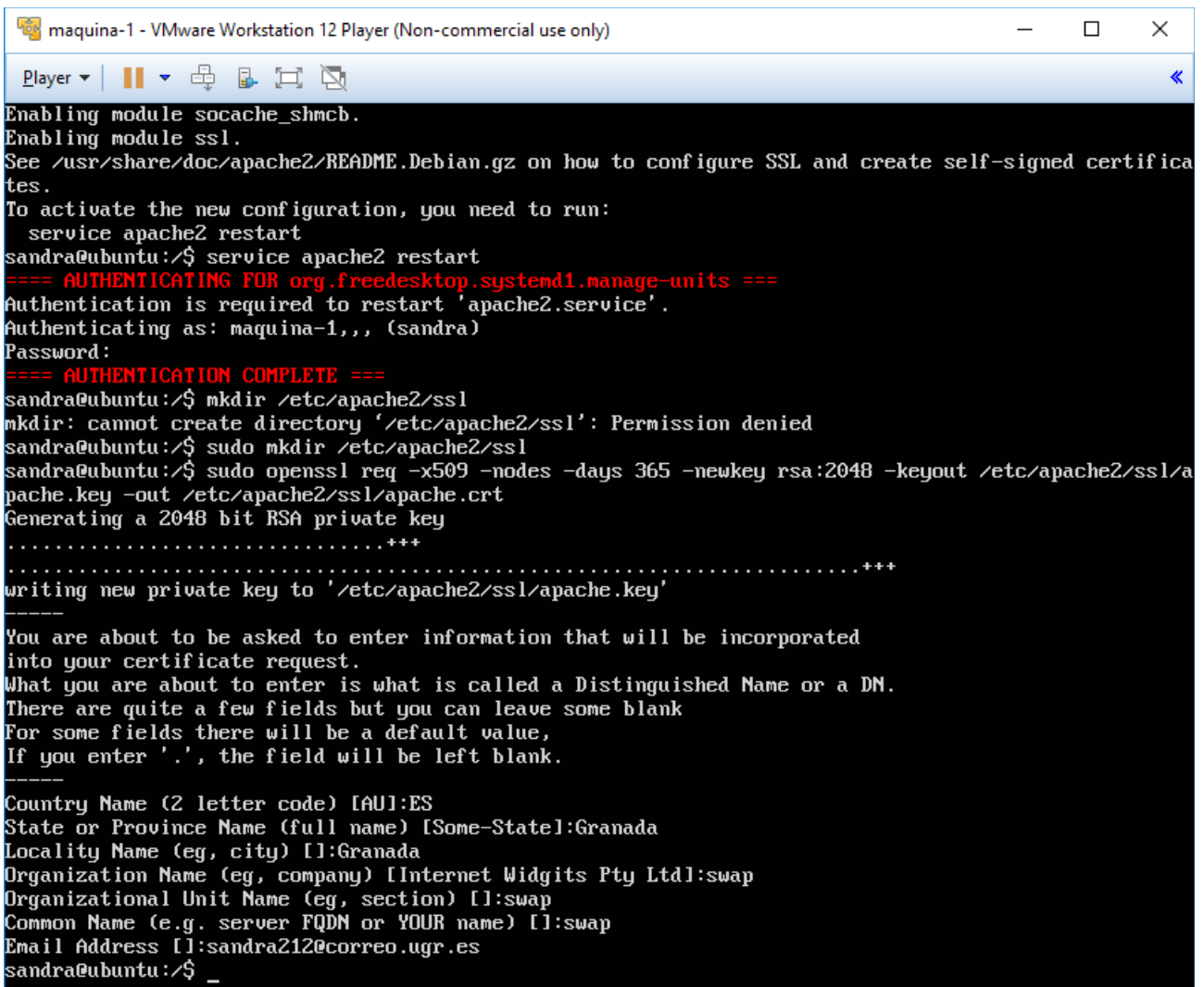
creamos la carpeta ssl dentro de la carpeta apache2 con:

```
mkdir /etc/apache2/ssl
```

y generamos los certificados con:

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

y rellenamos los datos que se nos piden:



```
maquina-1 - VMware Workstation 12 Player (Non-commercial use only)
Player ▾ | [Icons]
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
sandra@ubuntu:/$ service apache2 restart
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'apache2.service'.
Authenticating as: maquina-1,,, (sandra)
Password:
==== AUTHENTICATION COMPLETE ====
sandra@ubuntu:/$ mkdir /etc/apache2/ssl
mkdir: cannot create directory '/etc/apache2/ssl': Permission denied
sandra@ubuntu:/$ sudo mkdir /etc/apache2/ssl
sandra@ubuntu:/$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /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) []:swap
Common Name (e.g. server FQDN or YOUR name) []:swap
Email Address []:sandra212@correo.ugr.es
sandra@ubuntu:/$ _
```

Ahora editamos el archivo de configuración del sitio default-ssl con:

```
nano /etc/apache2/sites-available/default-ssl.conf
```

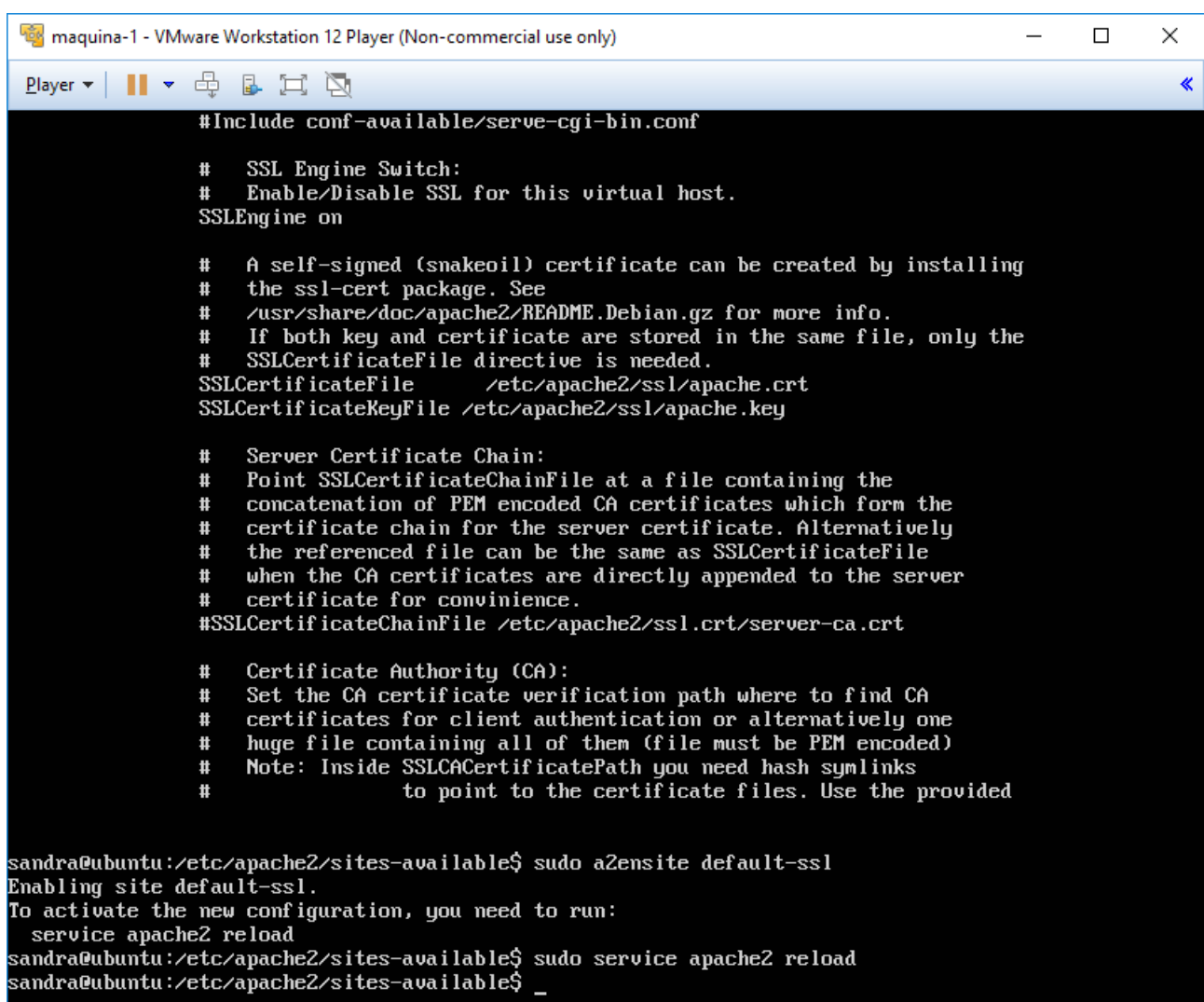
y añadimos debajo de SSL Engine on las siguientes líneas:

```
SSLCertificateFile /etc/apache2/ssl/apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/apache.key
```

Activamos el sitio default-ssl y reiniciamos apache:

```
a2ensite default-ssl
```

```
service apache2 reload
```



```
maquina-1 - VMware Workstation 12 Player (Non-commercial use only)
Player ▾ | [Icons]
#Include conf-available/serve-cgi-bin.conf

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSL Engine on

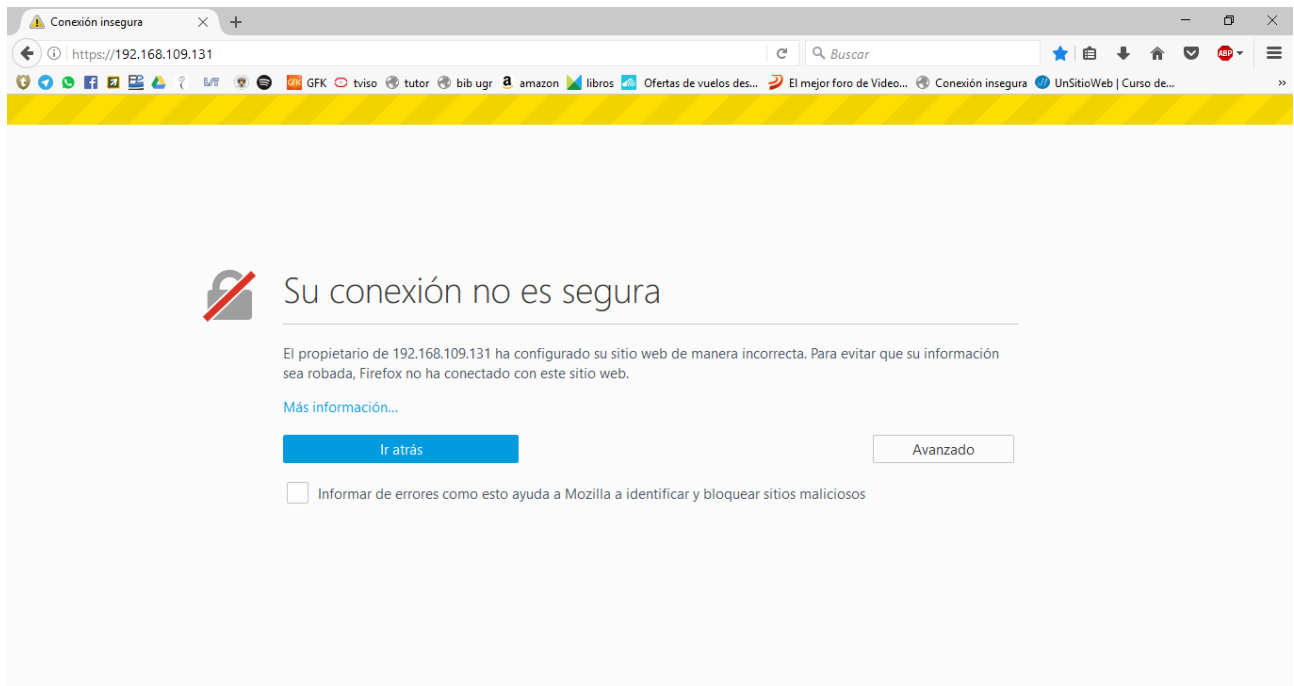
# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/apache2/ssl/apache.crt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

# 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
# certificate for convinience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)
# Note: Inside SSLCACertificatePath you need hash symlinks
# to point to the certificate files. Use the provided

sandra@ubuntu:/etc/apache2/sites-available$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
service apache2 reload
sandra@ubuntu:/etc/apache2/sites-available$ sudo service apache2 reload
sandra@ubuntu:/etc/apache2/sites-available$ _
```

Comprobamos que la conexión no es segura:



Configuración del cortafuegos

En primer lugar comprobamos el estado del cortafuegos con:

```
iptables -L -n -v
```

```
maquina-1 - VMware Workstation 12 Player (Non-commercial use only)
Player
sandra@ubuntu:/etc$ sudo iptables -L -n -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination

Chain FORWARD (policy ACCEPT 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
sandra@ubuntu:/etc$ _
```

Denegamos cualquier tráfico de información con:

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Bloqueamos el tráfico de entrada con:

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT
```

Bloqueamos todo el tráfico ICMP (ping), para evitar ataques como el del ping de la muerte:

```
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

Abrimos el puerto 22 para permitir el acceso por SSH:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p udp --sport 22 -j ACCEPT
```

Abrimos los puertos HTTP/HTTPS (80 y 443) para configurar un servidor web:

```
iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT
```

Abrimos el puerto 53 para permitir el acceso a DNS:

```
iptables -A INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT
iptables -A INPUT -m state --state NEW -p tcp --dport 53 -j ACCEPT
```

Bloqueamos todo el tráfico de entrada desde una IP (máquina 2: 192.168.109.132):

```
iptables -I INPUT -s 192.168.109.132 -j DROP
```

Bloqueamos todo el tráfico de salida hacia una IP (máquina 2: 192.168.109.132):

```
iptables -I OUTPUT -s 192.168.109.132 -j DROP
```

Evitamos el acceso a www.facebook.com especificando el nombre de dominio:

```
iptables -A OUTPUT -p tcp -d www.facebook.com -j DROP
```

```
maquina-1 - VMware Workstation 12 Player (Non-commercial use only)
Player
root@ubuntu:/etc# iptables -P INPUT DROP
root@ubuntu:/etc# iptables -P OUTPUT DROP
root@ubuntu:/etc# iptables -P FORWARD DROP
root@ubuntu:/etc#
root@ubuntu:/etc# iptables -P INPUT DROP
root@ubuntu:/etc# iptables -P FORWARD DROP
root@ubuntu:/etc# iptables -P OUTPUT ACCEPT
root@ubuntu:/etc# iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT
root@ubuntu:/etc#
root@ubuntu:/etc# iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
root@ubuntu:/etc#
root@ubuntu:/etc# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
root@ubuntu:/etc# iptables -A OUTPUT -p udp --sport 22 -j ACCEPT
root@ubuntu:/etc#
root@ubuntu:/etc# iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
root@ubuntu:/etc# iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT
root@ubuntu:/etc#
root@ubuntu:/etc# iptables -A INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT
root@ubuntu:/etc# iptables -A INPUT -m state --state NEW -p tcp --dport 53 -j ACCEPT
root@ubuntu:/etc#
root@ubuntu:/etc# iptables -I INPUT -s 192.168.109.132 -j DROP
root@ubuntu:/etc# iptables -I OUTPUT -s 192.168.109.132 -j DROP
root@ubuntu:/etc#
root@ubuntu:/etc# iptables -A OUTPUT -p tcp -d www.facebook.com -j DROP
root@ubuntu:/etc# _
```

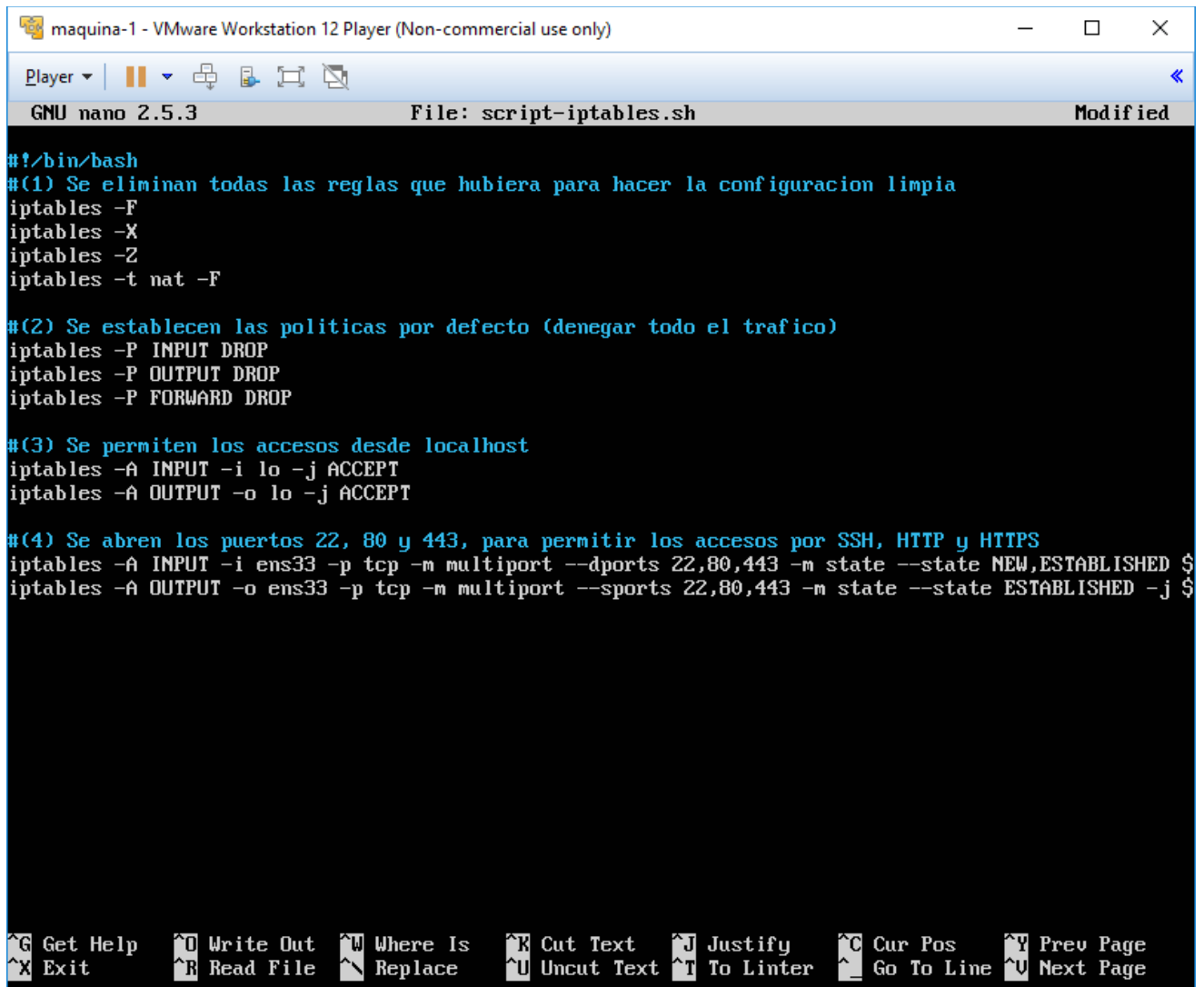
Y volvemos a comprobar el estado del cortafuegos:

```
maquina-1 - VMware Workstation 12 Player (Non-commercial use only)
Player
root@ubuntu:/etc# iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source               destination
    0      0 DROP      all  --  *      *        192.168.109.132      0.0.0.0/0
 329 26062 ACCEPT    all  --  *      *        0.0.0.0/0            0.0.0.0/0          state NEW,E
STABLISHED
    0      0 ACCEPT    all  --  *      *        0.0.0.0/0            0.0.0.0/0          state NEW,E
STABLISHED
    0      0 DROP      icmp  --  *      *        0.0.0.0/0            0.0.0.0/0          icmptype 8
    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          state NEW t
cp dpt:80
    0      0 ACCEPT    tcp   --  *      *        0.0.0.0/0            0.0.0.0/0          state NEW t
cp dpt:443
    0      0 ACCEPT    udp   --  *      *        0.0.0.0/0            0.0.0.0/0          state NEW u
dp dpt:53
    0      0 ACCEPT    tcp   --  *      *        0.0.0.0/0            0.0.0.0/0          state NEW t
cp dpt:53

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 DROP      all  --  *      *        192.168.109.132      0.0.0.0/0
    0      0 ACCEPT    udp   --  *      *        0.0.0.0/0            0.0.0.0/0          udp spt:22
    0      0 DROP      tcp   --  *      *        0.0.0.0/0            31.13.77.36
root@ubuntu:/etc#
```

Ahora hacemos un script, “script-iptables.sh”, que nos configure las reglas del cortafuegos:



```
maquina-1 - VMware Workstation 12 Player (Non-commercial use only)
Player
GNU nano 2.5.3 File: script-iptables.sh Modified

#!/bin/bash
#(1) Se eliminan todas las reglas que hubiera para hacer la configuracion limpia
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

#(2) Se establecen las politicas por defecto (denegar todo el trafico)
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

#(3) Se permiten los accesos desde localhost
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

#(4) Se abren los puertos 22, 80 y 443, para permitir los accesos por SSH, HTTP y HTTPS
iptables -A INPUT -i ens33 -p tcp -m multiport --dports 22,80,443 -m state --state NEW,ESTABLISHED $
iptables -A OUTPUT -o ens33 -p tcp -m multiport --sports 22,80,443 -m state --state ESTABLISHED -j $

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^Y Prev Page
^X Exit ^R Read File ^_ Replace ^U Uncut Text ^T To Linter ^_ Go To Line ^U Next Page
```

e indicamos en “/etc/rc.local” que se ejecute, para que la ejecución se realice al iniciar el sistema:

maquina-1 - VMware Workstation 12 Player (Non-commercial use only)

Player ▾ | [Icons] <<

GNU nano 2.5.3 File: rc.local Modified

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does not
./etc/apache2/ssl/script-iptables.sh_
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos ^V Prev Page
^X Exit ^R Read File ^_ Replace ^U Uncut Text ^T To Linter ^_ Go To Line ^U Next Page