

PRACTICA 4 REPORT

Antonios Matzakos Chorianopoulos

Theodoros Katsikeros

The aim of this practice is to protect the web farm we have created in the past exercises by installing an SSL certificate to configure HTTPS access to the servers and to configure firewall rules.

First, we generate and install an SSL certificate. We execute the following commands:

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

And then we complete the form that is generated:

```
root@ubuntuwap:/home/foris# 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:
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
root@ubuntuwap:/home/foris# service apache2 restart
bash: service: command not found
root@ubuntuwap:/home/foris# service apache2 restart
root@ubuntuwap:/home/foris# mkdir /etc/apache2/ssl
root@ubuntuwap:/home/foris# 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_UGR
Organizational Unit Name (eg, section) []:Christoforos
Common Name (e.g. server FQDN or YOUR name) []:Christoforos
Email Address []:foris95@yahoo.gr
```

And we add the lines, as instructed under the SSLEngine on, in this file:

/etc/apache2/sites-available/default-ssl

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
    ServerAdmin webmaster@localhost

    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
```

After restarting apache, if we access the server from a browser using its IP, the address bar will be red.

After that, we configure our firewall to block any incoming traffic that doesn't meet our safety criteria.

To do this we use the iptables application:

```

root@ubuntu:~# iptables -L -n -v
iptables v1.6.0
root@ubuntu:~# 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
root@ubuntu:~# iptables -P INPUT DROP
root@ubuntu:~# iptables -P FORWARD DROP
root@ubuntu:~# iptables -P OUTPUT ACCEPT
root@ubuntu:~# iptables -P INPUT -m state --state NEW,ESTABLISHED -j ACCEPT
iptables v1.6.0: -P requires a chain and a policy
Try `iptables -h' or 'iptables --help' for more information.
root@ubuntu:~# iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT
root@ubuntu:~# iptables -L -n -v
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,ESTABLISHED

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
root@ubuntu:~#

```

After we test some commands to learn about the iptables application, we have to open certain ports such as port 22 to allow SSH access,

```

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in     out     source                   destination
root@ubuntu:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables v1.6.0: unknown option "--dport22"
Try `iptables -h' or 'iptables --help' for more information.
root@ubuntu:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
root@ubuntu:~# iptables -A OUTPUT -p udp --dport 22 -j ACCEPT
root@ubuntu:~# iptables -A OUTPUT -p udp --sport 22 -j ACCEPT
root@ubuntu:~# iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target      prot opt in     out     source                   destination
 60 15072 ACCEPT      all  --  *      *        0.0.0.0/0                0.0.0.0/0
  0      0 ACCEPT      tcp  --  *      *        0.0.0.0/0                0.0.0.0/0

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      udp  --  *      *        0.0.0.0/0                0.0.0.0/0
  0      0 ACCEPT      udp  --  *      *        0.0.0.0/0                0.0.0.0/0
root@ubuntu:~#

```

```

root@ubuntu:~# iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
root@ubuntu:~# iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT
root@ubuntu:~# iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source               destination
  90 22608 ACCEPT     all  --  *      *        0.0.0.0/0            0.0.0.0/0            state NEW,ESTABLISHED
  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 tcp dpt:80
  0      0 ACCEPT     tcp  --  *      *        0.0.0.0/0            0.0.0.0/0            state NEW 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     udp  --  *      *        0.0.0.0/0            0.0.0.0/0            udp dpt:22
  0      0 ACCEPT     udp  --  *      *        0.0.0.0/0            0.0.0.0/0            udp spt:22
root@ubuntu:~#

```

```

root@ubuntu:~# iptables -A INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT
root@ubuntu:~# iptables -A INPUT -m state --state NEW -p tcp --dport 53 -j ACCEPT
root@ubuntu:~# iptables -L -n -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source               destination
  120 30144 ACCEPT     all  --  *      *        0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED
    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 tcp dpt:80
    0      0 ACCEPT     tcp  --  *      *        0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:443
    0      0 ACCEPT     udp  --  *      *        0.0.0.0/0            0.0.0.0/0           state NEW udp dpt:53
    0      0 ACCEPT     tcp  --  *      *        0.0.0.0/0            0.0.0.0/0           state NEW tcp 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 ACCEPT     udp  --  *      *        0.0.0.0/0            0.0.0.0/0           udp dpt:22
    0      0 ACCEPT     udp  --  *      *        0.0.0.0/0            0.0.0.0/0           udp spt:22

```

to check if our ports are open. For our commands to run on startup, we made a script that runs on startup.