UNIVERSIDAD DE GRANADA

Servidores Web de Altas Prestaciones

Práctica 4 – Asegurar la Granja Web

Arturo Alonso Carbonero

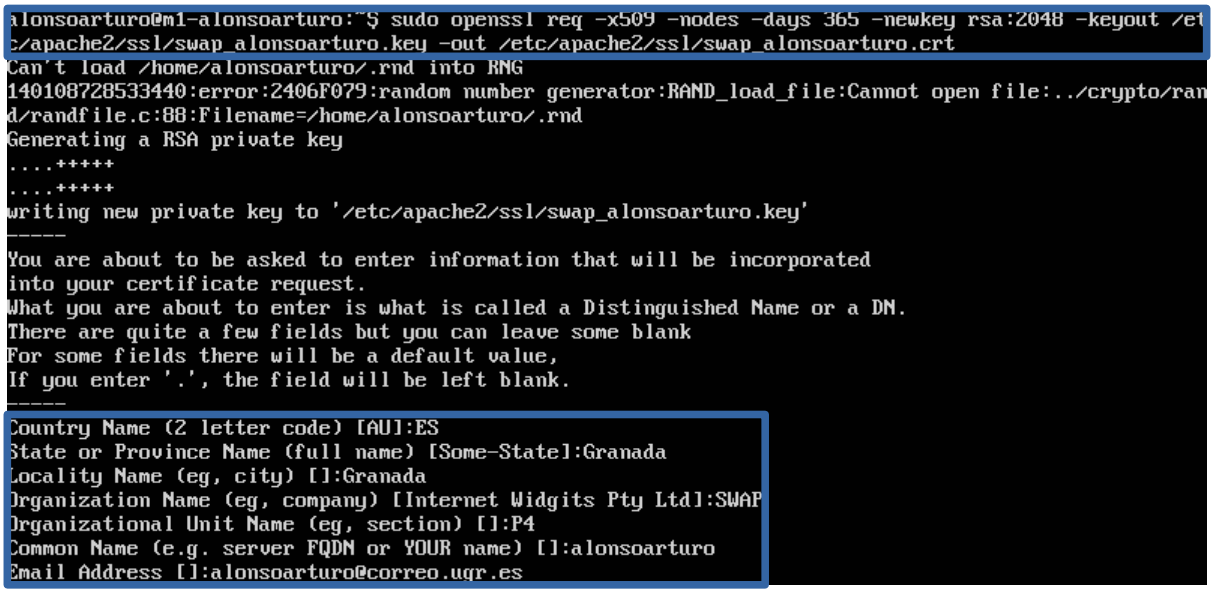
ÍNDICE

1. SSL

Para crear un certificado autofirmado hay que activar el módulo para SSL de Apache. Para ello, ejecutamos el comando **sudo a2enmod ssl** y reiniciamos el servicio de Apache mediante **systemctl** tal y como hemos hecho hasta ahora. Es necesario crear un directorio para almacenar el certificado y la llave privada que debemos crear. En este caso será el directorio **/etc/apache2/ssl**. Una vez hemos preparado el entorno, ejecutamos el siguiente comando: **sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/swap_alonsoarturo.crt -out /etc/apache2/ssl/swap_alonsoarturo.key**. Dicho comando crea el certificado (.crt) y la llave (.key) mencionados anteriormente. Las opciones utilizadas son las siguientes:

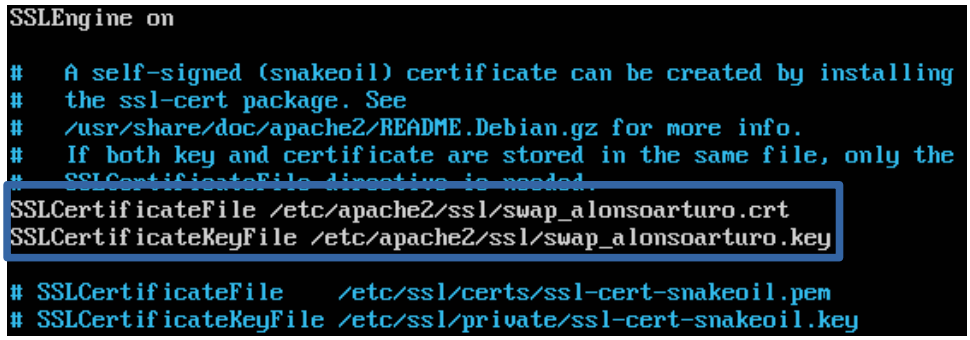
- **req** → Solicitud de firma de certificado.
- **-x509** → Crea una estructura X.509 en lugar de una petición.
- **-days** → Número de días que un certificado creado con x509 es válido.
- **-newkey rsa:2048** → Crear nueva llave encriptada usando RSA.
- **-out <fichero>** → Guarda la secuencia en el fichero indicado.

Una vez ejecutamos el comando, nos pedirá información sobre nuestro certificado, la cual rellenaremos como se indica en el guion de la práctica.



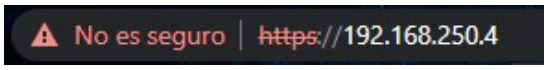
Creación del certificado SSL autofirmado

A continuación, editamos el fichero de configuración **/etc/apache2/sites-available/default-ssl.conf** haciendo uso del comando **nano, vi o vim** y agregamos la ruta del certificado creado tal y como se muestra en la imagen siguiente.

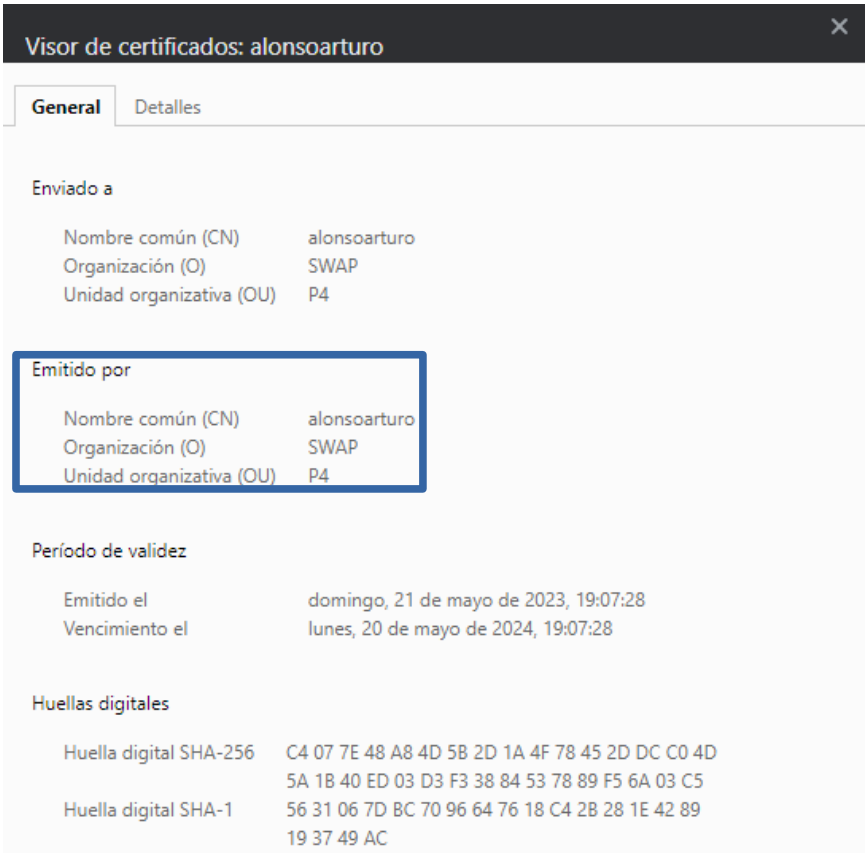


Adición del certificado creado

Es necesario comentar las líneas pertenecientes al certificado anterior. A continuación, activamos el sitio **default-ssl** usando el comando **sudo a2ensite default-ssl** y recargamos el servicio de Apache con **suso systemctl apache2 reload**. Si el proceso ha sido el adecuado, al acceder a la máquina desde un navegador encontraremos un mensaje de advertencia como el de la imagen.



Si pulsamos en el mensaje 'No es seguro' podemos seleccionar ver el certificado, el cual no es válido ya que está autofirmado.



Certificado

Los ficheros del certificado y a la llave privada deben copiarse al resto de máquinas mediante **sudo scp swap_alonsoarturo.crt/key alonsoarturo@IPmáquina:/home/alonsoarturo/-swap_alonsoarturo.crt/key**. Tras esto, en M2, repetimos el proceso. Dicho proceso solo es aplicable a las máquinas finales M1 y M2. En el caso de M3, el balanceador de carga, el proceso difiere. Para esta máquina, debemos añadir en el fichero **/etc/nginx/conf.d/default.conf** un nuevo *server* como se muestra en la siguiente imagen.

```
server{
  listen 443 ssl;
  ssl on;
  ssl_certificate /home/alonsoarturo/ssl/swap_alonsoarturo.crt;
  ssl_certificate_key /home/alonsoarturo/ssl/swap_alonsoarturo.key;
}
```

Server para M3

Si el proceso se ha llevado a cabo correctamente, deberíamos poder lanzar peticiones https a todas las máquinas haciendo uso del comando **curl -k https://IPmáquina** desde el anfitrión.

```
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl -k https://192.168.250.3
<html>
        <body>
                Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
                Correo: alonsoarturo@correo.ugr.es
        </body>
</html>
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl -k https://192.168.250.4
<html>
        <body>
                Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
                Correo: alonsoarturo@correo.ugr.es
        </body>
</html>
```

Peticiones HTTPS a M1 y M2

```
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl -k https://192.168.250.7
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Petición HTTPS a M3.

2. Iptables

2.1. Objetivos básicos

Para realizar la acción de cortafuegos se hará uso de **iptables**. Para comprobar el estado del mismo basta con ejecutar **sudo iptables -L -n -v**. Para realizar las acciones pertinentes, se ejecutará un script llamado **scriptP4.sh** situado en **/home/alonsoarturo**, mediante el comando **bash**, que contendrá los comandos con las reglas del cortafuegos. A continuación se muestra la configuración para permitir únicamente peticiones HTTP y HTTPS en la máquina M1.

```
# Eliminar todas las reglas
iptables -F
iptables -X

# Políticas por defecto (denegar todo el tráfico)
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Permitir acceso desde localhost
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Permitir solo HTTP y HTTPS
# Entrada
iptables -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT

# Salida
iptables -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
```

Script para iptables

Las opciones utilizadas son las siguientes:

- **-A** → Nombre de la cadena.
- **-p** → Protocolo.
- **--sport** → Puerto fuente.
- **--dport** → Puerto destino.
- **-m** → Añadir módulo.
- **--state** → Estado de la petición.
- **-j** → Acción.

Para copiar el script en el resto de máquinas basta con hacerlo a través de **scp**.

```
alonsoarturo@m1-alonsoarturo:~$ sudo scp scriptP4.sh alonsoarturo@192.168.250.3:/home/alonsoarturo/s
criptP4.sh
alonsoarturo@192.168.250.3's password:
scriptP4.sh                                   100%  955   369.4KB/s   00:00
alonsoarturo@m1-alonsoarturo:~$ sudo scp scriptP4.sh alonsoarturo@192.168.250.7:/home/alonsoarturo/s
criptP4.sh
alonsoarturo@192.168.250.7's password:
scriptP4.sh                                   100%  955   348.5KB/s   00:00
```

Copiar el script a M2 y M3

Por último, si se desea reiniciar el cortafuegos, ejecutamos los comandos de la siguiente imagen.

```
227  sudo scp scriptP4.sh alonsoarturo@192.168.250.3:/home/alonsoarturo/scriptP4.sh
228  iptables -F
229  sudo iptables -F
230  sudo iptables -X
231  sudo iptables -Z
232  sudo iptables -t nat -F
233  sudo iptables -P INPUT ACCEPT
234  sudo iptables -P OUTPUT ACCEPT
235  sudo iptables -P FORDWARD ACCEPT
236  sudo iptables -P FORWARD ACCEPT
237  sudo scp scriptP4.sh alonsoarturo@192.168.250.3:/home/alonsoarturo/scriptP4.sh
```

Reconfigurar iptables

- Eliminar todas las reglas.
  - iptables -F
  - iptables -X
  - iptables -Z
  - iptables -t nat -F

- Aceptar todas las peticiones.
  - Iptables -P INPUT ACCEPT
  - Iptables -P OUTPUT ACCEPT
  - Iptables -P FORWARD ACCEPT

Para comprobar que el resultado ha sido el deseado, basta con ejecutar un comando que utilice un protocolo distinto a HTTP o HTTPS como **ping**, que utiliza **icmp**, y vemos como no es aceptado.

```
alonsoarturo@DESKTOP-UKJ4G5Q:~$ ping 192.168.250.4
PING 192.168.250.4 (192.168.250.4) 56(84) bytes of data.
^C
--- 192.168.250.4 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2001ms

alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl -k https://192.168.250.4
<html>
        <body>
                Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
                Correo: alonsoarturo@correo.ugr.es
        </body>
</html>
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl http://192.168.250.4
<html>
        <body>
                Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
                Correo: alonsoarturo@correo.ugr.es
        </body>
</html>
```

Resultado para M1

```
alonsoarturo@DESKTOP-UKJ4G5Q:~$ ping 192.168.250.3
PING 192.168.250.3 (192.168.250.3) 56(84) bytes of data.
^C
--- 192.168.250.3 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1000ms

alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl -k https://192.168.250.3
<html>
        <body>
                Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
                Correo: alonsoarturo@correo.ugr.es
        </body>
</html>
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl http://192.168.250.3
<html>
        <body>
                Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
                Correo: alonsoarturo@correo.ugr.es
        </body>
</html>
```

Resultado para M2

```
alonsoarturo@DESKTOP-UKJ4G5Q:~$ ping 192.168.250.7
PING 192.168.250.7 (192.168.250.7) 56(84) bytes of data.
^C
--- 192.168.250.7 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl http://192.168.250.7
<html>
        <body>
                Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
                Correo: alonsoarturo@correo.ugr.es
        </body>
</html>
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl -k https://192.168.250.7
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
```

Resultado para M3

2.2. Objetivos extra

Permitir solo SSH, PING y DNS

A continuación se muestra cómo debe estar configurado el script anterior para permitir únicamente SSH, PING y DNS.

```
# Eliminar todas las reglas
iptables -F
iptables -X

# Políticas por defecto (denegar todo el tráfico)
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Permitir acceso desde localhost
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Permitir solo HTTP y HTTPS
# Entrada
# iptables -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
# iptables -A INPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT

# Salida
# iptables -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
# iptables -A INPUT -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
```

Reconfigurar iptables

```
# Permitir SSH
# Entrada
iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT

# Salida
iptables -A INPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

# Permitir PING
iptables -A INPUT -p icmp --icmp-type 8 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type 0 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# Permitir DNS
# UDP
iptables -A INPUT -p udp --sport 53 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p udp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT

# TCP
iptables -A INPUT -p tcp --sport 53 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT
```

Permitir SSH, PING y DNS

A continuación se muestra el resultado de la configuración anterior para la máquina M1. Como se puede observar, las peticiones HTTP y HTTPS no son aceptadas, mientras que PING se ejecuta con normalidad y es posible realizar conexión SSH. En cuanto a DNS, como se puede observar en el resultado de ejecutar **ping**, el servidor resuelve correctamente la dirección IP.

```
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl -k https://192.168.250.4
^C
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl http://192.168.250.4
^C
alonsoarturo@DESKTOP-UKJ4G5Q:~$ ping 192.168.250.4 -c 3
PING 192.168.250.4 (192.168.250.4) 56(84) bytes of data.
64 bytes from 192.168.250.4: icmp_seq=1 ttl=64 time=1.00 ms
64 bytes from 192.168.250.4: icmp_seq=2 ttl=64 time=1.55 ms
64 bytes from 192.168.250.4: icmp_seq=3 ttl=64 time=1.36 ms

--- 192.168.250.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 1.008/1.309/1.554/0.230 ms
```

PING y DNS

```
alonsoarturo@DESKTOP-UKJ4G5Q:~$ ping 192.168.250.4
PING 192.168.250.4 (192.168.250.4) 56(84) bytes of data.
^C
--- 192.168.250.4 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1000ms

alonsoarturo@DESKTOP-UKJ4G5Q:~$ ssh alonsoarturo@192.168.250.4
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.15.0-156-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon May 22 15:54:50 CEST 2023
```

SSH

Permitir solo HTTP y HTTPS de M3 en M1 y M2

Para permitir que las máquinas M1 y M2 acepten peticiones HTTP y HTTPS provenientes de M3 únicamente, hay que añadir la IP de origen (**-s**) y de destino (**-d**) en la configuración de las reglas para M1 y M2 tal y como se muestra en la siguiente imagen. En M3, basta con permitir peticiones HTTP y HTTPS normalmente.

```
# Permitir solo HTTP y HTTPS
# Entrada
iptables -A INPUT -s 192.168.250.7 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -d 192.168.250.7 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -s 192.168.250.7 -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -d 192.168.250.7 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT

# Salida
iptables -A INPUT -s 192.168.250.7 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -d 192.168.250.7 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -s 192.168.250.7 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -d 192.168.250.7 -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
```

Configuración de M1 y M2

```
# Permitir solo HTTP y HTTPS
# Entrada
iptables -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT

# Salida
iptables -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
```

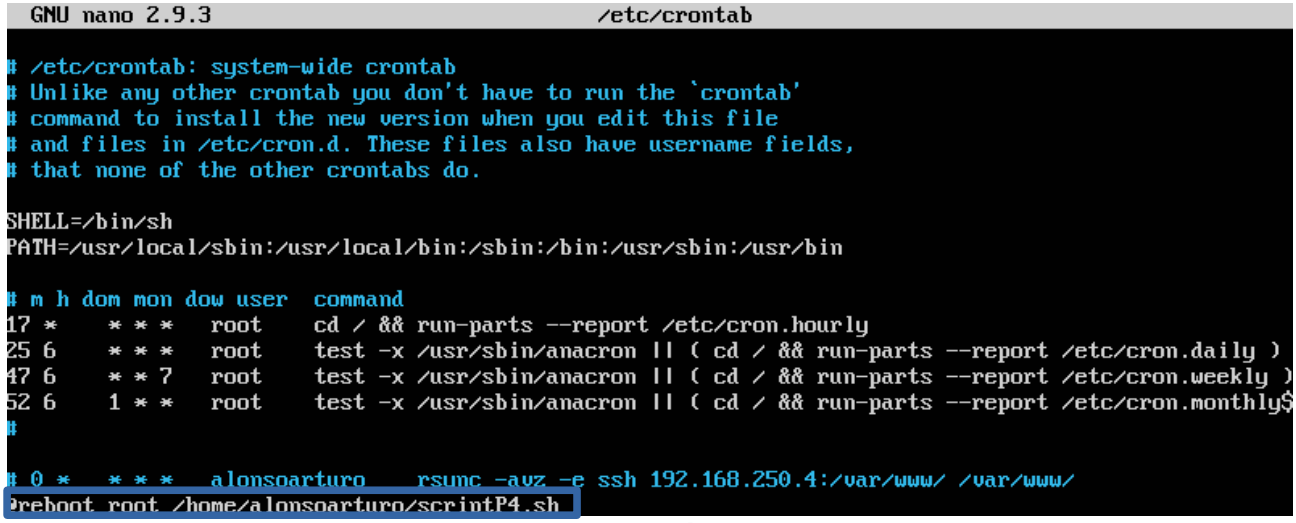
Configuración de M3

```
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl http://192.168.250.4
^C
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl -k https://192.168.250.4
^C
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl http://192.168.250.3
^C
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl -k https://192.168.250.3
^C
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl http://192.168.250.7
<html>
        <body>
                Web de ejemplo en la máquina m1 de <b>alonsoarturo</b> para SWAP
                Correo: alonsoarturo@correo.ugr.es
        </body>
</html>
alonsoarturo@DESKTOP-UKJ4G5Q:~$ curl http://192.168.250.7
<html>
        <body>
                Web de ejemplo en la máquina m2 de <b>alonsoarturo</b> para SWAP
                Correo: alonsoarturo@correo.ugr.es
        </body>
</html>
```

Resultado

Activar configuración en el aranque

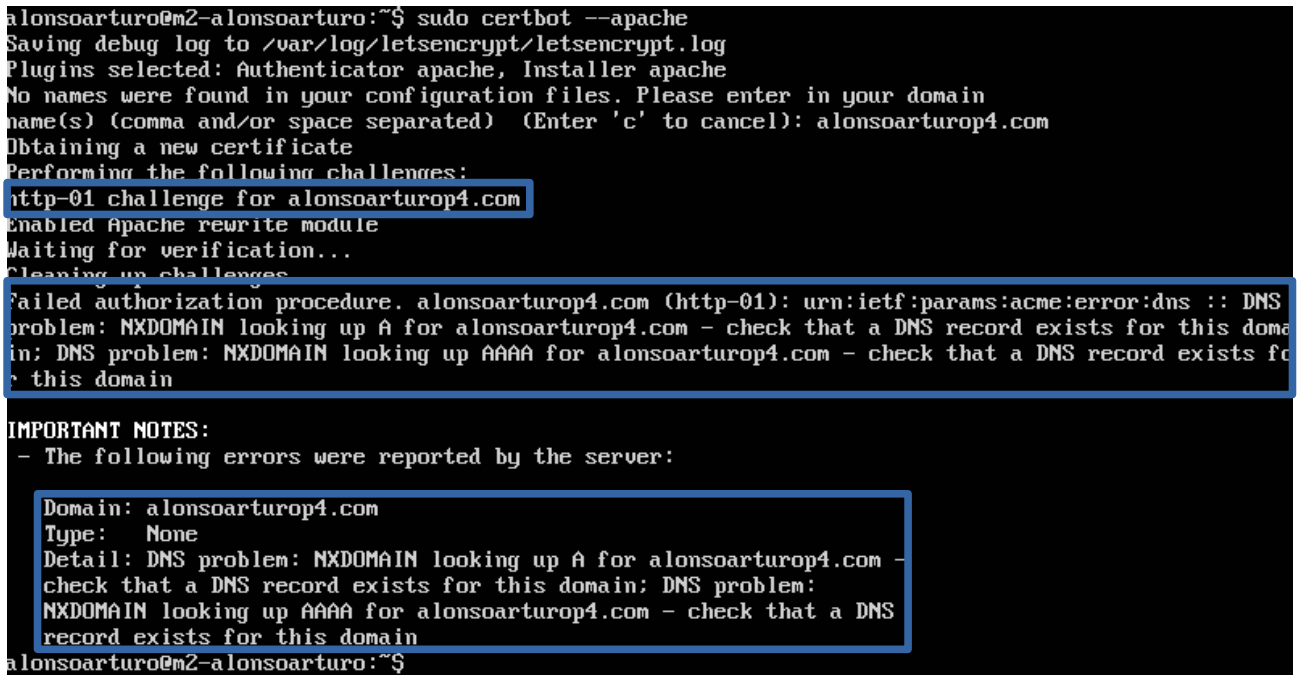
Para hacer que el script con la configuración del cortafuegos se ejecute en el arranque basta con hacer uso de la herramienta **crontab**. Para ello, ejecutamos **crontab -e** y añadimos la línea de la siguiente imagen en el fichero. La opción temporal **@reboot** ejecuta el comando en el arranque y la opción **root** la ejecuta como superusuario, lo cual es necesario ya que se está haciendo uso del cortafuegos.



Crontab

3. SSL con Certbot

Para crear un certificado con Certbot hay que instalar la herramienta. Para ello, ejecutamos el comando **sudo snap install--classic certbot** y ejecutamos con **sudo certbot --apache**. Rellenamos la información solicitada para intentar crear el certificado. En este caso, como no disponemos de ningún dominio, no podemos completar este paso, por lo que introducimos un dominio cualquiera. Es por eso que el certificado no puede ser creado con éxito usando Certbot.



Certbot

4. Referencias

SSL

- http://stuff.gpul.org/2004_cripto/doc/chuleta_openssl.pdf
- https://www.openssl.org/docs/man1.0.2/man1/openssl.html

IPtables

- https://www.ionos.es/digitalguide/servidores/herramientas/iptables-conoce-las-reglas-para-crear-paquetes-de-datos/
- https://linux.die.net/man/8/iptables

Certbot

- https://weblinus.com/instalar-certificado-para-https-con-certbot/
- https://certbot.eff.org/
- https://help.clouding.io/hc/es/articles/360021481640-Cómo-adquirir-certificados-Let-s-Encrypt-con-Certbot