Práctica 4

Asegurar la granja web

Duración: 3 sesiones

1. Objetivos de la práctica

El objetivo de esta práctica es llevar a cabo la configuración de seguridad de la granja web. Para ello, llevaremos a cabo las siguientes tareas:

- Instalar un certificado SSL para configurar el acceso HTTPS a los servidores.
- Configurar las reglas del cortafuegos para proteger la granja web.

2. Instalar un certificado SSL autofirmado para configurar el acceso por HTTPS

Un certificado SSL sirve para brindar seguridad al visitante de su página web, una manera de decirles a sus clientes que el sitio es auténtico, real y confiable para ingresar datos personales.

El protocolo SSL (Secure Sockets Layer) es un protocolo de comunicación que se ubica en la pila de protocolos sobre TCP/IP. SSL proporciona servicios de comunicación segura entre cliente y servidor, como por ejemplo autenticación (usando certificados), integridad (mediante firmas digitales), y privacidad (mediante encriptación).

La versión actual es la SSLv3, que se considera insegura. El nuevo estándar se llama TLS (Transport Layer Security).

Existen diversas formas de obtener un certificado SSL e instalarlo en nuestro servidor web para poder servir páginas mediante el protocolo HTTPS, para ello, lo principal es conseguir un certificado que podremos conseguir de las siguientes formas:

- Mediante una autoridad de certificación.
- Crear nuestros propios certificados SSL auto-firmados usando la herramienta openssl.
- Utilizar certificados del proyecto Certbot (antes Let's Encrypt).

Generar e instalar un certificado autofirmado

Para generar un certificado SSL autofirmado en Ubuntu Server solo debemos activar el módulo SSL de Apache, generar los certificados e indicarle la ruta a los certificados en la configuración. Así pues, como root ejecutaremos en la máquina M1:

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

Nos pedirá una serie de datos para configurar el certificado del dominio. Deberéis crear el certificado con los siguientes campos:
- Nombre de país: *ES*
- Provincia: *Granada*
- Localidad: *Granada*
- Organización: *SWAP*
- Organización sección: *P4*
- Nombre: "*usuario_ugr*"
- Email: "*email_ugr*"

Editamos el archivo de configuración del sitio **default-ssl**:

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

Y agregamos la ruta de los certificados debajo del parámetro **SSLEngine on**:

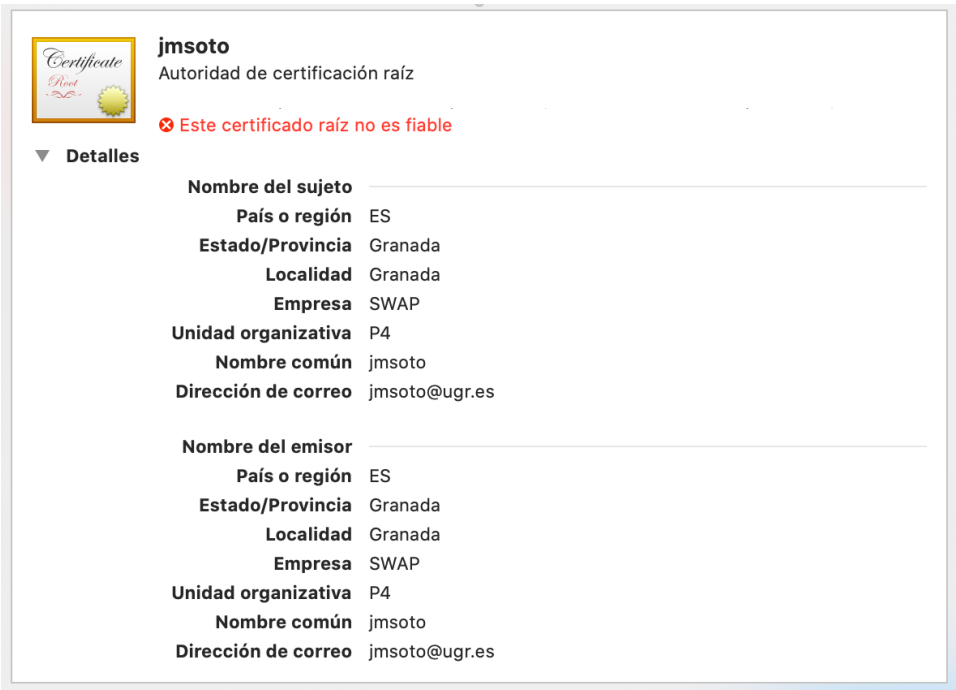
```
SSLCertificateFile /etc/apache2/ssl/swap_usuarioUGR.crt
SSLCertificateKeyFile /etc/apache2/ssl/swap_usuarioUGR.key
```

Activamos el sitio **default-ssl** y reiniciamos apache:

```
a2ensite default-ssl
service apache2 reload
```

Una vez reiniciado Apache, accedemos al servidor web mediante el protocolo HTTPS y veremos, si estamos accediendo con un navegador web, que en la barra de dirección sale en rojo el https, ya que se trata de un certificado autofirmado.

Debéis de comprobar que el certificado es el que hemos instalado mostrándolo desde el navegador web.



Para hacer peticiones por HTTPS utilizando la herramienta curl, podemos ejecutar:

```
curl -k https://ipM1/index.html
```

Por último, y como queremos que la granja nos permita usar el HTTPS, debemos configurar el balanceador para que también acepte este tráfico (puerto 443). Para hacer esto, copiaremos la pareja de archivos (el .crt y el .key) a todas las máquinas de la granja web. **No debemos generar más certificados**, sino que los archivos *swap_usuarioUGR.crt* y *swap_usuarioUGR.key* que generamos en el servidor M1 en el paso anterior vamos a copiarlos al otro servidor (M2) y al balanceador (M3). Para copiarlos podemos usar scp o rsync.

```
sudo scp swap_usuarioUGR.crt
usuario@ipm2:/home/usuario/swap_usuarioUGR.crt
sudo scp swap_usuarioUGR.key
usuario@ipm2:/home/usuario/swap_usuarioUGR.key
```

Al igual que en la máquina M1, en la máquina M2 debemos crear directorio /etc/apache2/ssl, copiar los archivos .crt y .key, configurar default-ssl.conf, activar el sitio default-ssl y reiniciar apache.

```
sudo mkdir /etc/apache2/ssl
sudo mv /home/usuario/apache* /etc/apache2/ssl
sudo a2enmod ssl & sudo service apache2 restart
sudo nano /etc/apache2/sites-available/default-ssl.conf
      "SSLCertificateFile /etc/apache2/ssl/swap_usuarioUGR.crt
      SSLCertificateKeyFile /etc/apache2/ssl/swap_usuarioUGR.key"
sudo a2ensite default.ssl
Sudo service apache2 reload
```

En el balanceador (M3) pondremos la ruta a la carpeta donde hayamos copiado el *apache.crt* y el *apache.key*. Después, en el balanceador nginx debemos añadir nuevo **server** al archivo /etc/nginx/conf.d/default.conf igual que el configurado en la práctica anterior pero ahora añadiendo los siguientes datos:

```
listen 443 ssl;
ssl on;
ssl_certificate      /home/usuario/ssl/swap_usuarioUGR.crt;
ssl_certificate_key  /home/usuario/ssl/swap_usuarioUGR.key;
```

Ahora ya podremos hacerle peticiones por HTTPS a la IP del balanceador.

3. Configuración del cortafuegos

Un cortafuegos es un componente esencial que protege la granja web de accesos indebidos. Son dispositivos colocados entre subredes para realizar diferentes tareas de manejo de paquetes. Actúa como el guardián de la puerta al sistema web, permitiendo el tráfico autorizado y denegando el resto.

En general, todos los paquetes TCP/IP que entren o salgan de la granja web deben pasar por el cortafuegos, que debe examinar y bloquear aquellos que no cumplan los criterios de seguridad establecidos. Estos criterios se configuran mediante un conjunto de reglas, usadas para bloquear puertos específicos, rangos de puertos, direcciones IP, rangos de IP, tráfico TCP o tráfico UDP.

Configuración del cortafuegos *iptables* en Linux

iptables es una herramienta de cortafuegos, de espacio de usuario, con la que el superusuario define reglas de filtrado de paquetes, de traducción de direcciones de red,

y mantiene registros de log. Esta herramienta está construida sobre Netfilter, una parte del núcleo Linux que permite interceptar y manipular paquetes de red.

Se basa en establecer una lista de reglas con las que definir qué acciones hacer con cada paquete en función de la información que incluye. La sintaxis del comando iptables está documentada en su página de manual (teclear el comando "man iptables" en el shell), aunque también se pueden encontrar multitud de tutoriales y páginas de ayuda en Internet.

Para configurar adecuadamente *iptables* en una máquina Linux, conviene establecer como reglas por defecto la denegación de todo el tráfico, salvo el que permitamos después explícitamente. Una vez hecho esto, a continuación definiremos nuevas reglas para permitir el tráfico solamente en ciertos sentidos necesarios, ya sea de entrada o de salida. Por último, definiremos rangos de direcciones IP a los cuales aplicar diversas reglas, y mantendremos registros (logs) del tráfico no permitido y de intentos de acceso para estudiar más tarde posibles ataques.

Uso de la aplicación iptables

A continuación mostraremos cómo utilizar la herramienta para establecer ciertas reglas y filtrar algunos tipos de tráfico, o bien controlar el acceso a ciertas páginas:

Toda a información sobre la herramienta está disponible en su página de manual y usando la opción de ayuda:

```
man iptables
iptables –h
```

Para comprobar el estado del cortafuegos, debemos ejecutar:

```
iptables –L –n –v
```

También se puede parar el cortafuegos y eliminar al mismo tiempo todas sus reglas:

```
iptables -F
iptables -X
iptables –t nat -F
iptables –t nat -X
iptables –t mangle -F
iptables –t mangle -X
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
```

Para denegar cualquier tráfico de información, podemos hacer:

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables –L –n –v
```

Para bloquear el tráfico de entrada, podemos hacer:

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

Abrir 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
```

Abrir 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
```

Abrir 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
```

Bloquear todo el tráfico de entrada/salida para una IP específica:

```
iptables -I INPUT -s 150.214.13.13 -j DROP
iptables -I OUTPUT -s 31.13.83.8 -j DROP
```

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

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

En algunas ocasiones, en lugar de repetir conjuntos de reglas para diferentes puertos, conviene usar reglas que usen la opción multipuerto (aviso: son órdenes largas que no han cabido en este guion en una sola línea):

```
iptables -A INPUT  -i eth0 -p tcp -m multiport --dports 22,80,443
      -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp -m multiport --sports 22,80,443
      -m state --state ESTABLISHED -j ACCEPT
```

Por último, conviene comprobar el funcionamiento del cortafuegos recién configurado. Para ello, pediremos al sistema que nos muestre qué puertos hay abiertos y qué demonios o aplicaciones los tienen en uso. Para ello, utilizaremos la orden netstat como se muestra a continuación:

```
      netstat -tulpn
```

Por ejemplo, para asegurarnos del estado (abierto/cerrado) del puerto 80, podemos ejecutar:

```
      netstat -tulpn | grep :80
```

Lo habitual es crear un script que se ejecute en el arranque del sistema. Veamos a continuación un ejemplo de script para la configuración básica de una máquina Linux:

```
# (1) se eliminan todas las reglas que hubiera
# para hacer la configuración limpia:
iptables -F
iptables -X

# (2) establecer las políticas por defecto (denegar todo el tráfico):
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# (3) permitir cualquier acceso desde localhost (interface lo):
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# (4) permitir la salida del equipo (output) con conexiones nuevas que
# solicitemos, conexiones establecidas y relacionadas. Permitir la
# entrada (input) solo de conexiones establecidas y relacionadas:
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
```

En cualquier momento, si hubiéramos cometido algún error, podemos poner la configuración que tenía la máquina inicialmente (permitir todo el tráfico):

```
# (1) Eliminar todas las reglas (configuración limpia)
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

# política por defecto: aceptar todo
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT

iptables -L -n -v
```

Cuestiones a resolver

El objetivo de esta práctica es configurar aspectos relativos a la seguridad de la granja web. Se debe añadir *usuarioUGR* en las distintas configuraciones e ilustrarlo con capturas de pantalla.

En esta práctica **se llevarán a cabo las siguientes tareas básicas**:

1. 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.
2. Copiar al resto de máquinas servidoras (M2) y al balanceador de carga (M3) el certificado autofirmado creado en M1 (archivos .crt y .key) y configurarlas para que acepten tráfico HTTP y HTTPS.
3. Denegar todo el tráfico entrante a las máquinas M1, M2 y M3 a excepción de tráfico HTTP y HTTPS.
4. Configurar y documentar las reglas del cortafuegos con IPTABLES a través de un script en cada máquina con las reglas creadas.

Como **tareas avanzadas**:

1. Permitir SSH, PING y DNS a las máquinas M1, M2 y M3 así como el tráfico consigo misma (localhost). El resto de servicios y/o peticiones debe denegarse.
2. Configurar M3 estableciendo reglas de iptables para que sólo M3 sea quien acepte peticiones HTTP y HTTPS mientras que M1 y M2 no acepten peticiones a no ser que sean peticiones provenientes de M3.
3. Hacer que la configuración del cortafuegos se ejecute al arranque del sistema en todas las máquinas.
4. **Adicional**: Crear, instalar y configurar un certificado SSL con Cerbot u otro

NOTA: El correcto funcionamiento de todas las configuraciones debe mostrarse y justificarse documentalmente

Normas de entrega

La práctica se realizará de manera individual.

Se entregará un documento .pdf con el desarrollo de la práctica según el guion detallando, en su caso, los aspectos básicos y avanzados realizados. Se deja a libre elección la estructura del documento el cual reflejará el correcto desarrollo de la práctica a modo de diario/tutorial. En el documento de texto a entregar se describirá cómo se han realizado las diferentes configuraciones (así como comandos de terminal a ejecutar en cada momento).

Para la entrega se habilitará una tarea en PRADO donde se entregará el documento desarrollado siguiendo OBLIGATORIAMENTE el formato **ApellidosNombreP4.pdf**

Evaluación

La práctica se evaluará mediante el uso de rúbrica específica (accesible por el estudiante en la tarea de entrega) y una defensa final de prácticas.

Tiene un peso del 20% del total de prácticas

La detección de prácticas copiadas implicará el suspenso inmediato de todos los implicados en la copia (tanto del autor del original como de quien las copió).
OBLIGATORIO ACEPTAR LICENCIA EULA DE TURNITIN
Si la memoria supera un 40% de copia Turnitin —> suspenso
del 1-10% -> 0
del 11-20% -> -1
del 20-30% —> -2
del 30-40% —> -3
40% —> suspenso

Las faltas de ortografía se penalizarán con hasta 1 punto de la nota de la práctica.

Referencias

- https://en.wikipedia.org/wiki/Transport_Layer_Security
- https://en.wikipedia.org/wiki/HTTPS
- https://en.wikipedia.org/wiki/OpenSSL
- https://github.com/certbot/certbot
- https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-load-balancing-with-ssl-termination
- https://www.linuxtotal.com.mx/?cont=info_seyre_002
- http://es.tldp.org/Manuales-LuCAS/doc-iptables-firewall/doc-iptables-firewall.pdf
- http://www.thegeekstuff.com/2011/06/iptables-rules-examples
- https://www.digitalocean.com/community/tutorials/how-to-forward-ports-through-a-linux-gateway-with-iptables
- https://unix.stackexchange.com/questions/322879/port-forward-why-is-iptables-with-postrouting-rule-required
- http://www.ubuntuleon.com/2016/10/cargar-un-script-al-inicio-del-sistema.html
- http://rm-rf.es/etc-rc-local-ejecutar-comandos-o-scripts-en-el-arranque-de-nix/
- https://www.digitalocean.com/community/tutorials/how-to-test-your-firewall-configuration-with-nmap-and-tcpdump