Práctica 4. Asegurar la granja web

Duración: 2 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 y especificarle la ruta a los certificados en la configuración. Así pues, como root ejecutaremos:

```
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
```

Nos pedirá una serie de datos para configurar el dominio.

```
swap1:~$ 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 []:info@swap.es
swap1:~$
```

Editamos el archivo de configuración del sitio default-ssl:

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

Y agregamos estas lineas debajo de donde pone SSLEngine on:

```
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
```

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.

Para hacer peticiones por HTTPS utilizando la herramienta curl, ejecutaremos:

```
curl -k https://ipmaquina1/index.html
```

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
```

Para lanzar, reiniciar o parar el cortafuegos, y para salvar las reglas establecidas hasta ese momento, ejecutaremos respectivamente:

```
service iptables start
service iptables restart
service iptables stop
service iptables save
```

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
```

Bloquear 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
```

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 desde una IP:

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

Bloquear todo el tráfico de salida hacia una IP:

```
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 guión 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
```

Como segundo ejemplo, veamos cómo realizar una configuración básica para un servidor web:

```
# (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
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) 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) Abrir los puertos HTTP (80) de servidor web
iptables -A INPUT  -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 -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 todos los aspectos relativos a la seguridad de la granja web ya creada.

Hay que **llevar a cabo las siguientes tareas obligatorias**:

1. Crear e instalar en la máquina 1 un certificado SSL autofirmado para configurar el acceso HTTPS a los servidores. Una vez configurada la máquina 1, se debe copiar al resto de máquinas servidoras y al balanceador de carga. Se debe configurar nginx adecuadamente para aceptar y balancear correctamente tanto el tráfico HTTP como el HTTPS.

2. Configurar las reglas del cortafuegos con IPTABLES para asegurar el acceso a uno de los servidores web, permitiendo el acceso por los puertos de HTTP y HTTPS a dicho servidor. Esta configuración se hará en una de las máquinas servidoras finales (p.ej. en la máquina 1), y se debe poner en un script con las reglas del cortafuegos que se ejecute en el arranque del sistema (según la versión de Linux, se llevará a cabo de una forma u otra).

Adicionalmente, y como primera tarea opcional para conseguir una mayor nota en esta práctica, se propone realizar la instalación de un certificado del proyecto Certbot en lugar de uno autofirmado. Es importante tener en cuenta que para obtener este tipo de certificado, es necesario disponer de un dominio real con IP pública (no se puede hacer en máquinas virtuales).

Como segunda tarea opcional para conseguir una mayor nota en esta práctica, se propone realizar la configuración del cortafuegos en una cuarta máquina (M4) que se situará delante del balanceador. Esa M4 sólo tendrá configuradas las iptables, para hacer el filtrado y posterior reencaminamiento del tráfico hacia el balanceador. En esta configuración más compleja sólo a esa M4-cortafuegos se le hará la configuración de iptables (el resto de máquinas de la granja tendrá la configuración por defecto, aceptando todo el tráfico como política por defecto).

Como resultado de la práctica 4 **se mostrará** al profesor el funcionamiento del acceso por HTTPS a las páginas web almacenadas en los servidores finales, haciendo peticiones con la herramienta curl por HTTP/HTTPS tanto a la máquina 1 como al balanceador de carga. También se mostrará el funcionamiento del filtrado del tráfico HTTP/HTTPS con las reglas de iptables configuradas. En el documento de texto a entregar se describirá cómo se han realizado las diferentes configuraciones (tanto configuraciones y comandos de terminal a ejecutar en cada momento).

Normas de entrega

La práctica podrá realizarse de manera individual o por grupos de hasta 2 personas.

Se entregará como un archivo de texto en el que se muestre la información requerida. También se puede utilizar la sintaxis de Markdown para conseguir una mejor presentación e incluso integrar imágenes o capturas de pantalla. La entrega se realizará subiendo los archivos necesarios al repositorio SWAP en la cuenta de GitHub del alumno, a una carpeta llamada “practica4”.

Toda la documentación y material exigidos se entregarán en la fecha indicada por el profesor. No se recogerá ni admitirá la entrega posterior de las prácticas ni de parte de las mismas.

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).

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://servidordebian.org/es/squeeze/intranet/ssl_cert/self_signed
- 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.ubuntu-es.org/node/187790#.WRCAljftZRB
- 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/
- http://www.alvarolara.com/2013/03/20/ejecutar-un-script-al-iniciar-sesion-en-ubuntu/
- https://www.digitalocean.com/community/tutorials/how-to-test-your-firewall-configuration-with-nmap-and-tcpdump