Ejercicios Tema 1

1-Buscar información sobre las tareas o servicios web para los que se usan los siguientes programas:

-Apache: Es un servidor web HTTP de código abierto para plataformas Unix, Microsoft, Macintosh y otras, que implementa el protocolo HTTP/1.1.

-Nginx: Es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico. Puede usarse como balanceador de carga.

-thttpd: Es un servidor web de código libre. Se caracteriza por ser simple, pequeño, rápido y seguro, ya que usa los requerimientos mínimos de un servidor HTTP.

-Cherokee: Es un servidor web multiplataforma. Está escrito completamente en c. Puede usarse como un sistema embebido y soporta complementos para aumentar sus funcionalidades. Puede usarse como balanceador de carga.

-node.js: Es un entorno en tiempo de ejecución multiplataforma, para la capa del servidor. Es usado para la creación de programas de red altamente escalables, como por ejemplo, servidores web.

Ejercicios Tema 2

2-Buscar frameworks y librerías para diferentes lenguajes que permitan hacer aplicaciones altamente disponibles con relativa facilidad.

-BlueCove: Librería en Java para usar Bluetooth.
-**Netbeans:** Framework que permite desarrollar aplicaciones en java, c, c++, ruby, etc.
-Django: Framework para phyton.
-Laravel: Framework para php.

3-¿Cómo analizar el nivel de carga de cada uno de los subsistemas en el servidor? Buscar herramientas y aprender a usarlas.

-Podemos usar top y sus diferentes versiones. Por otro lado podemos usar ApacheBenchmark, vmstat, Htop, Nagios, Icinga y otras muchas herramientas de monitorización de software libre.

4- Buscar ejemplos de balanceadores software y hardware (productos comerciales). Buscar productos comerciales para servidores de aplicaciones. Buscar productos comerciales para servidores de almacenamiento.

-Como balanceadores Software tenemos los usados en las prácticas de la asignatura, es decir, Haproxy y Nginx, mientras que como balanceador hardware tenemos los balanceadores de carga distribuidos por CISCO.
-Como productos comerciales para servidores de aplicaciones podemos optar por los distribuidos por IBM.
-Como productos comerciales para almacenamiento, tenemos opciones gratuitas como Dropbox o GoogleDrive, mientras que de pago podemos optar por el Servidor NAS.

Ejercicios Tema 3

1-Buscar con qué órdenes de terminal o herramientas gráficas podemos configurar bajo Windows y bajo Linux el enrutamiento del tráfico de un servidor para pasar el tráfico desde una subred a otra.

-En Linux, debemos indicarle al servidor mediante iptables qué tiene que redireccionar con el siguiente comando:
iptables -t nat -A PREROUTING -p tcp --dport <puerto receptor> -j DNAT --to-destination <ip final>:<puerto de ip final>
También podemos hacerlo añadiento la línea route add-net en el archivo de configuración de la red, que está en /*etc/*network/interfaces.

-En Windows al algo parecido mediante el comando route print. Por lo que para enrutar usamos el siguiente comando:
route add red-destino *mask* máscaraSubred puertaEnlace *metric* métrica *if* interfaz

2-Buscar con qué órdenes de terminal o herramientas gráficas podemos configurar bajo Windows y bajo Linux el filtrado y bloqueo de paquetes.

-El filtrado de bloque y filtrado de paquetes en Linux lo realizamos mediante el comando iptables, por ejemplo si queremos prohibir la salida por el puerto 80 hacemos los siguiente:
iptables -A OUTPUT -p tcp --destination-port 80 -j DROP

-En Windows esta tarea la realiza el propio Firewall de Windows.

Ejercicios Tema 4:

1-Buscar información sobre cuánto costaría en la actualidad un mainframe que tuviera las mismas prestaciones que una granja web con balanceo de carga y 10 servidores finales (p.ej). Comparar precio y potencia entre esa hipotética máquina y la granja web de unas prestaciones similares.

-Lo único que he encontrado en este ejercicio es que IBM en la actualidad comercializa dos tipos de mainframes, que son el IBM z13, el IBM z13s y e IBM z14, siendo el z13 Un mainframe empresarial muy potente y el z13s su versión más económica, minetras que el z14 es un mainframe de almacenamiento. Los precios no los proporciona IBM en su web ya que tienes que ser asesorado por ellos de antemano.

2-Buscar información sobre precio y características de balanceadores hardware específicos. Compara las prestaciones que ofrecen unos y otros.

-La empresa KenpTechnologies tiene una gama de balanceadores de carga, llamada Load Master, que cuenta con 4 modelos: LM-X3, LM-X15, LM-X25 y LM-X40.

El LMX3, con un precio de 4000$ cuenta con un procesador Intel Dual Core, 8 X GbE Auto-negotiating, Full Duplex ports, 500GB HDD, 8 GB RAM

El LM-X15 con un precio de 9800$ cuenta con un Intel Quad Core Processor, 16 x GbE Auto-negotiating, Full Duplex ports, 4 x 10Gb SFP+ ports,2 x 500GB HDD (RAID), 32 GB RAM.

El LM-X25 con un precio de 20000$ cuenta con un 2 x 12 Core Intel Xeon Processor, 12 x 10Gb SFP+ Direct Attach Ports, 64 GB RAM, 2 x 1TB HDD (RAID).

Por último, el LM-X40 con un precio de 30000$ cuenta con un Intel 12 Core Processor, 2 x GbE Auto-negotiating, Full Duplex ports, 12 x 10Gb SFP+ ports, 2 x 1TB HDD (RAID 1), 64 GB RAM.

3-Buscar información sobre los métodos de balanceo que implementan los dispositivos recogidos en el ejercicio 4.2 (o el software que hemos propuesto para la práctica 3).

-Durante la práctica 3 usamos nginx y haproxy. Nginx cuenta con los siguientes métodos de balanceo: Round Robin, leats Connections, Hash e Ip Hash, mientras que haproxy cuenta con los siguientes métodos: Round Robin, Número de conexiones, Fuente de la petición, URL y parámetros en la URL.

5-Probar las diferentes maneras de redirección HTTP.¿Cuál es adecuada y cuál no lo es para hacer balanceo de carga global? ¿Por qué?

-La mejor opción sería la redirección con PHP puesto que con esta redirección se consigue que el ancho de banda sea el mínimo contra el balanceador.

Ejercicios Tema 5

1-Buscar información sobre cómo calcular el número de conexiones por segundo.

-En Ubuntu podemos usar la herramienta Loadtest, con esta herramienta hacemos una petición a la ip del servidor de la siguiente manera:
loadtest -c 10 -n 10000 http://127.0.0.1:8080/
Esto nos devolverá como salida lo siguiente:

Completed requests: 10000
Total time: 4.472977 s
Requests per second: 2236
Mean latency: 4.44 ms

Donde encontramos el número de peticiones por segundo.

3-Buscar información sobre características, funcionalidad, disponibilidad para diversos SO, etc de herramientas para monitorizar las prestaciones de un servidor.

-Top muestra los procesos que están corriendo actualmente en Linux, vmstat devuelve las estadísticas relacionadas con la memoria y netstat devuelve las estadísticas relacionadas con las redes.

-Asimismo podemos encontrar más herramientas para monitorizar las prestaciones, tales como: mpstat, df, free, ps o gprof.

Ejercicios Tema 6

1-Aplicar con iptables una política de denegar todo el tráfico en una de las máquinas de prácticas.Comprobar el funcionamiento. Aplicar con iptables una política de permitir todo el tráfico en una de las máquinas de prácticas. Comprobar el funcionamiento.

-Aceptar Todo:

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

# (2) Politica por defecto: aceptar todo
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -L -n -v
```

-Denegar Todo:

```
#!/bin/bash

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

# (2) Politica por defecto: denegar_todo
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables -L -n -v
```

2-Comprobar qué puertos tienen abiertos nuestras máquinas, su estado, y qué programa o demonio lo ocupa.

```
root@marino-CX61-2PC:/home/marino# netstat -tulpn
Conexiones activas de Internet (solo servidores)
Proto  Recib Enviad Dirección local         Dirección remota       Estado       PID/Program name
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               ESCUCHAR    1119/mysqld
tcp        0      0 127.0.0.53:53           0.0.0.0:*               ESCUCHAR    500/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               ESCUCHAR    1112/sshd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               ESCUCHAR    3161/cupsd
tcp        0      0 0.0.0.0:10050           0.0.0.0:*               ESCUCHAR    1082/zabbix_agentd
tcp        0      0 0.0.0.0:10051           0.0.0.0:*               ESCUCHAR    1092/zabbix_server
tcp6       0      0 :::80                   :::*                    ESCUCHAR    1299/apache2
tcp6       0      0 :::22                   :::*                    ESCUCHAR    1112/sshd
tcp6       0      0 ::1:631                 :::*                    ESCUCHAR    3161/cupsd
tcp6       0      0 :::10050                :::*                    ESCUCHAR    1082/zabbix_agentd
tcp6       0      0 :::10051                :::*                    ESCUCHAR    1092/zabbix_server
udp        0      0 0.0.0.0:43344           0.0.0.0:*                           1056/snmpd
udp        0      0 127.0.0.53:53           0.0.0.0:*                           500/systemd-resolve
udp        0      0 0.0.0.0:68              0.0.0.0:*                           14088/dhclient
udp        0      0 127.0.0.1:161           0.0.0.0:*                           1056/snmpd
udp        0      0 0.0.0.0:631             0.0.0.0:*                           3162/cups-browsed
udp        0      0 0.0.0.0:53994           0.0.0.0:*                           886/avahi-daemon: r
udp        0      0 224.0.0.251:5353        0.0.0.0:*                           13494/chrome --type
udp        0      0 224.0.0.251:5353        0.0.0.0:*                           13446/chrome
udp        0      0 224.0.0.251:5353        0.0.0.0:*                           13446/chrome
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           886/avahi-daemon: r
udp6       0      0 :::5353                 :::*                                886/avahi-daemon: r
udp6       0      0 :::54750                :::*                                886/avahi-daemon: r
root@marino-CX61-2PC:/home/marino#
```

3-Buscar información acerca de los tipos de ataques más comunes en servidores web (p.ej. secuestros de sesión). Detallar en qué consisten, y cómo se pueden evitar.

-**DoS:** En un ataque de denegación de servicio (DoS), un atacante intenta evitar la legitimidad de que los usuarios accedan a información o al servicios. El tipo más común y obvio de ataque DoS ocurre cuando un atacante "inunda" una red con información.
-Ping Flood: Ping flood se basa en enviar a la víctima una cantidad abrumadora de paquetes ping, usualmente usando el comando "ping" de UNIX como hosts
-ICMP Tunneling: El tunneling se usa a menudo para eludir los firewalls que no bloquean los paquetes ICMP, o para establecer un canal de comunicación cifrado y difícil de rastrear entre dos computadoras sin interacción directa de la red.
-Inyección SQL: ejecutar un código debido a la presencia de vulnerabilidad en la capa de la base de datos de la Aplicación. En consecuencia, el código obtendrá datos confidenciales o incluso comprometerá la aplicación en sí.

Ejercicios Tema 7

1-Buscar información sobre los sistemas de ficheros en red más utilizados en la actualidad y comparar sus características. Hacer una lista de ventajas e inconvenientes de todos ellos, así como grandes sistemas en los que se utilicen.

-Los ficheros de ficheros en red más usados actualmente son:

-SMB/CIFS: Sistema nativo de Windows. Permite navegar por los recursos ofrecidos y está orientado al funcionamiento en LAN.

-NFS: Es el sistema nativo de Unix. No está pensado para navegar por los recursos y funciona en WAN.

-Coda: El cliente guarda de forma local los ficheros de trabajo, para asegurar la disponibilidad cuando no existe conexión de red.

-Intermezzo: Inspirado en Coda pero diseñado de nuevo.

-Lustre: Nuevo desarrollo destinado a supercomputación. Para grandes clusters o procesadores masivamente paralelos (MPP).