**Unidad 1**

Internet= red de redes

Red de computadoras – hardware y software conectado mediante dispositivos físicos, operan a través de un canal físico y un código común llamado protocolo

TCP – transmisión control protocol (conjunto de reglas), ip- internet protocol

Medios físicos: par trenzado: 3 cables de cobre, hasta 1/10gbps

Medio guiado – coaxial, cobre, fibra

No guiados: infra rojo, radio

El coaxial tiene dos conductores concéntricos de cobre con aislante entre ellos – no es bueno para distancias largas como la fibra

La fibra óptica es fibra de vidrio q transporta pulso de luz

Niveles de isp: nivel 1 – cobertura nacional/internacional (hay pocos en el mundo) – 15

Nivel 2: los regionales, se conectan a 1 o mas isp de nivel 1 y a otros de nivel 2

Nivel3: son los isp locales, los mas cercanos a sistemas terminales.

Pila de protocolos:

Aplicación: aplicaciones de red : ftp, smtp, http, dns, Messenger, Skype, etc

Transporte: transferencia de datos tcp, udp, identificado por puerto

Red: ruteo de datagramas, identificado por ip, protocolos

Enlace: ethernet, wifi, identificado por mac

El tcp tiene capa link, network, transport y application

1GBytes son 1000MBytes, 1MB son 1000KB, 1KB son 8KB

1gbps 1000mbps

SWITCH – operan en la capa de enlace, analizan y modifican frames MAC, realizan filtradio y reenvio de paquetes – dividen una red en varios canales aislados

ROUTER – reenvía paquetes entre redes y trabaja a nivel red, usa direcciones lógicas y tablas de ruteo

Se usan para encapsular dispositivos también

STORE AND FORWARD – almacenamiento y reenvio: significa q el equipo conmutador recibe el paquete completo antes de empezar a transmitirlo

Cada router tiene en memoria una cola de paquetes

Demora de propagación- tiempo q tarda la señal desde q comienza hasta que llega al otro extremo, tiempo de bit, velocidad de transmisión, demora de transmisión, de encolamiento y de procesamiento

Tiempo de transmisión = tamaño del archivo en bits/velocidad de transmisión en bps

**Unidad 2**

Mac. Dirección de cada dispositivo física.

Ipv4 son direcciones de 32 digitos binarios (4 bytes)

Mascara= ip/potencia de 2 para alcanzar los requerimientos

Ej: 255.255.255.0/24 = 11111111.11111111.11111111.00000000

Switch: operan en la capa de enlace de datos y no producen colision como los hubs ya que dividien varios dominios de colisión.

Routeo:

Ruta con menos saltos, puede ser dinámico o estático

Tabla de routeo: red de destino, directa o indirecta, próximo router (via), interfaz de salida, si para llegar a muchas redes hay q pasar x el mismo router se simplifica poniendolo como default

En core:

Ip route add default via<dir-ip<dest>dev<output-interface>

Ip route add <red/mask>via<dir-ip-dest>dev<output-interface>

Traceroute: se usa para determinar ruta de un paquete hasta su destino

Traceroute <dir-ip>

VLSM: variable lenght subnet mask

Se utiliza para divider redes, variando el tamaño de la mascara de acuerdo a las necesidades (recordar que son hosts + 1dir base +1 broadcast)

Para pasar de binario a decimal se debe asignar una potencia de 2 a cada numero comenzando desde el 0, de izquierda a derecha y multiplicar cada numero x dicha potencio para luego sumarlos.

En core asignas ip con ip addr add<ip/masc>dev<interface>

ARP:adress resolution protocol – traduce la ip a la dirección mac – la tabla arp tiene información de la mac y de la ip de cada dispositivo.

Netfilter:

Firewall: cortafuegos – separa una maquina o subred del resto de la red filtrando paquetes

Nat: tabla que contiene ip privadas, ip publica de destino y puerto – se usa para dar salida a internet a través de la ip publica

NETFILTER:

Framework Linux para autorizar o rechazar paquetes

Iptables -t filter -A input -p tcp -I eth1 –destination-port 21 -j DROP

(descarta)

Prerouting – paquetes q llegan a la maquina

Input – paquetes destinados a la propia maquina

Forward – paquetes que han llegado a la maquina pero q hay q reenviar a otra

Output – paquetes creados x la maquina

Postrouting- paquetes q salen de la maquina

Snat=source nat – cambia ip o Puerto de origen de paquetes salientes

DNAT=destination nat – cambia ip o Puerto a paquetes entrantes

Masquerade= enmascaramiento ip o NAPT – el servidor identifica q uno de los equipos quiere conectarse a una red de afuera y de esta manera el servidor toma la misma ip del cliente y se enmascara.

WIRESHARK:

Para diagnosticar problemas de red – muestra los paquetes enviados y recibidos en una interfaz- Se pueden filtrar para ver solo paquetes de determinada red o protocolo como http x ejemplo

Traceroute – determina la ruta x la que pasa el paquete hasta alcanzar su destino

DOCKER:

Sirve para deployear apps dentro de contenedores que son un paquete de elementos que permiten ejecutar una ap en cualquier sistema operativo

Imagen: son templates con instrucciones para crear containers – una imagen esta basada en otra imagen y contiene configuración adicional. Vos podes usar una imagen de Ubuntu para crear otra que use Ubuntu y además corra un apache

Dockerfile: archivo que dice como crear la imagen

From: la imagen base

Workdr: carpeta donde vas a trabajar

Copy:copiar archivos locales a la imagen

Run:ejecutar comando en la imagen

Cmd: comando de inicialización de container

Docker hub/registry es en donde se guardan las imágenes

Ej:

FROM Ubuntu

CMD echo “this is a test”

FROM node:latest

WORKDIR /usr/src/app

Copy package.json yarn.lock

RUN yarn

COPY .

RUN yarn build

Docler image pull nginx – descarga la imagen usando el registry por defecto y el tag por defecto

Docker image ls – ver imágenes descargadas

Docker image inspect {container\_id} – ver info de bajo nivel de la imagen

Docker build – construye la imagen desde un Docker file

Docker build –tag cdar24/clasedocker:1.0 –imagen con tag

Docker run -d –rm -p nginx:latest

O Docker run -d –rm -p 80:80 nginx:latest //ejecuta imagen

Docker ps -muestra contenedores ejecutando

Docker ps -a apagados y ejecutados

CREAR SERVER WEB EN DOCKER

Descargar el Docker file y hacerle Docker build

Docker run

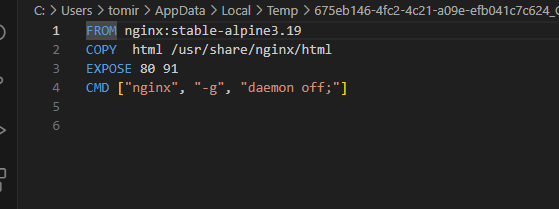
Ingresas al puerto y se va a ver la pagina corriendo

Este dockerfile dice de donde descargar la imagen base nginx

Y copia la carpeta html que esta en el mismo directoria hacia el /usr/share/nginx/html que es la carpeta del nginx

Expose 80 91 hace que esos sean los puertos a usar

Y en la línea cmd le indica a nginx que se ejecute en primer plano y no como demoño



UNIDAD 3

Cifrados:

Simétricos:

Cesar: desplazado

Cesar combinado: en lugar de desplazar, cualquier letra se sustituye por cualquier otra.

Poli alfabético: cada posición tiene un alfabeto diferente y una misma letra que aparece en distintas posiciones puede codificarse de distinta forma

Simétrico moderno:

De flujo: redes inalámbricas

De bloque: se divide el mensaje en componentes de igual longitud y se codifican esas partes

Cifrado asimetrico – se posee una clave publica y una privada

El emisor usa la clave publica ofrecida x el receptor y el mensaje solo puede ser descifrado x la clave privada q conoce el receptor

Algoritmo RSA

Es mas lento q el cifrado simetrico

Hash – toman un input de tamaño arbitrario y producen un output de tamaño fijo llamado hash (como lo q hacíamos en web2)

Un cambio mínimo cambia el hash de manera significativa – propiedad avalancha

Hashes: md5-128bits, es decir 16bytes

Sha-256: 256bits

Firmas digitales – indica el propietario de un documento

Vpn – virtual private network

Permiten usar infraestructura publica en forma privada – es útil para una empresa que tiene muchas oficinas

Túnel de datos – al conectarse a una vpn el trafico va al isp y luego al server vpn y luego al destino q corresponda.

Proxy – intermediario que asegura la seguridad, el control administrativo y el cacheo de la información

Puede ocultar, esconder y mantener tu identificador anonimo

El proxy se diferencia del vpn ya que no cifra todo el trafico, la vpn tambien enmascara la ip y hace creer q estas en otro lado pero su concepto va mas allá de eso

El proxy tiene menos costo de ancho de banda

Cacheo – acelera el servicio de request a través de guardar lo solicitado en un request previo

Proxy web – cachea todo el trafico de internet y es el uso mas común de los web proxy

Filtrado de contenido: provee el control administrativo del contenido q puede ser solicitado o accedido. Se usa en instituciones

Cifrado – conversión de datos de forma legible a un formato codificado

Pila de protocolos:

Aplicación/transporte/red/enlace

La aplicación esta compuesto x protocolos y servicios de red para las aplicaciones del usuario – http, dns

Transporte – transferencia de datos host-host: tcp/udp, se identifica por puerto

Red: ruteo de datagramas desde fuente a destino: ip, protocolso de ruteo – se identifca por ip

Enlace – transferencia de datos entre elementos vecinos en la red – ppp, ethernet, wifi: identificado por mac

Aplicaciones de red – corren en diferentes sistemas y se comunican por la red. No se refiere a software escrito para dispositivos en la red interna (los routers, switchers) no funcionan en la capa aplicación.

Arquitecturas cliente-servidor, p2p, e hibridas

P2p – peer to peer: no se obtienen los datos de un servidor que esta siempre encendido, sino que los pares se conectan intermitentemente

Proceso. – programa q corre en una maquina – dentro de una maquina los procesos se comunican usando comunicación entre procesos intercambiando mensajes.

Direccionamiento de procesos – ip y numero de puerto (16bits, va del 0 al 65535)

Netstat -t tcp

Netstat .u udp

Servicios de la capa transporte necesarios:

Confiabilidad en la entrega(sin perdida) – audio, video, etc

Bajo retardo – telefonía, juegos

Tasa de datos – algunas necesitan un mínimo de ancho de banda para funcionar bien y otras funcionan de forma elástica

Tcp – orientado a coneccion, transporte confiable, control de flujo, controla la congestion, no provee garantías de retardo ni ancho de banda

Udp- transferencia no confiable, no tiene establecimiento de conexión, confiabilidad, control de flujos

http: hypertext transfer protocol

es un protocolo de la capa de aplicación

usa tcp, no guarda “estado”, es decir, el servidor no guarda info de los requerimientos de los clientes

http no persistente

un objeto es enviado x cada conexión tcp, se usaba en http 1.0

http persistente: multiples objetos en una única conexión, a partir de http 1.1

tiempo de respuesta: RTT(round trip time)

un rtt para iniciar conexión, otro por requerimiento http y primeros bytes de respuesta.

Total=2RTT + tiempo de transmisión

http no persistente requiere al menos 2 rtts por objeto, gasta mas recursos xq el navegador abre conexiones paralelas generalmente para traer objetos referenciados.

Persistente – el servidor deja las conexiones abiertas

Con pipeling – default en http 1.1, 1 rtt para todas las transferencias, se envia requerimientos tan pronto encuentra un objeto referenciado

Sin pipeling – cliente envia nuevo requerimiento solo cuando el previo fue recibido, un rtt x cada objeto referenciado

Certificados – ssl – certifica q el sitio es seguro, este certificado se envia al navegador luego de solicitar la conexión e incluye la clave publica del servidor y la firma de una autoridad

Filter

Input output y forward

Nat

Prerouting y postrouting y output

Netfilter

Dnat redirigir puertos

Masquerade salir a internet

Forward para q deje pasar

Input o output para que no llegue algo al router pero es mejor usar forward

* **iptables [-t <tabla>] -A <cadena> <condicion> <accion>**