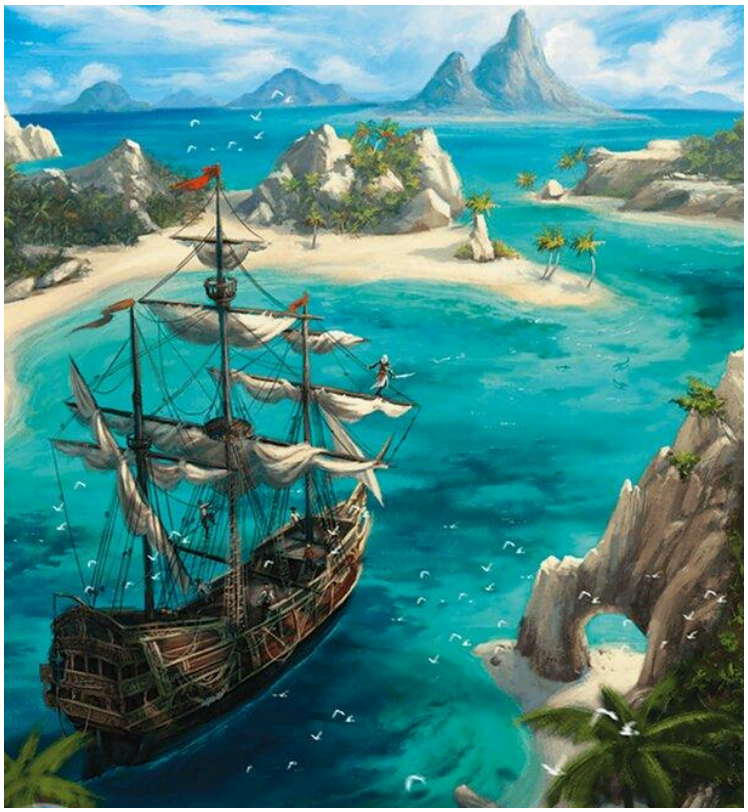
M01UF2 Gestión de la información y los recursos en una red

Apuntes y Anotaciones de clase

Ejercicios Continuos



Precaución al navegar por la web,
en algunos sitios todavía hay dragones…

Alexandre Fernandez Estebanm
1r DAMVIOD

Conceptos Básicos

Los protocolos son las reglas que permiten que los dispositivos en una red se comuniquen de manera clara y eficiente. Estos definen cómo enviar y recibir datos asegurando su seguridad y correcta interpretación.

Elementos clave de los protocolos:

- **Formato (Sintaxis):** Cómo deben estructurarse los mensajes.
- **Significado (Semántica):** Qué significan los mensajes que se intercambian.
- **Sincronización:** Cuándo deben enviarse y recibirse los mensajes.
- **Control de errores:** Métodos para detectar y corregir problemas en la transmisión.
- **Seguridad:** Cifrado y autenticación para proteger los datos.

Tipos de Protocolos

HTTP y HTTPS

- **HTTP:** Protocolo para transferir páginas web (puerto 80).
- **HTTPS:** Versión segura que utiliza cifrado TLS/SSL (puerto 443).

SSH

- **Propósito:** Administrar servidores de forma remota.
- **Puerto:** 22.
- **Características:** Usa cifrado para proteger la conexión.

SMTP

- **Propósito:** Envía correos electrónicos.
- **Puerto:** 25.
- **Nota:** Permite la comunicación entre servidores de correo.

Direcciones IP

IPv4 vs IPv6

- **IPv4:** Direcciones de 32 bits, limitadas a 4.3 mil millones de combinaciones.
- **IPv6:** Direcciones de 128 bits, mucho más abundantes y con soporte para nuevas tecnologías.

Ejemplos:

- IPv4: 192.168.1.1
- IPv6: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

Herramientas de Red

Ncat

Conecta clientes y servidores, útil para depuración.

Modo escucha:
nc -l puerto

Enviar datos:
echo "mensaje" | nc dirección puerto

Comandos en Linux

- **chmod +x archivo:** Permite ejecutar un script.
- **./archivo:** Ejecuta un archivo en el directorio actual.
- **ps aux:** Muestra procesos activos.

Scripts de Cliente y Servidor

Cliente

- Conecta con el servidor.
- Envía:
    1. Cabecera inicial.
    2. Nombre y contenido del archivo.
    3. Hash MD5 del contenido.
- Espera respuestas para confirmar cada paso.

Servidor

- Escucha conexiones.
- Valida:
    1. Cabecera inicial.
    2. Nombre del archivo y su hash MD5.
    3. Contenido recibido y su hash MD5.
- Responde según las verificaciones realizadas.

Conceptos Fundamentales

¿Qué es un puerto?

Es una "puerta numerada" que organiza servicios en un dispositivo. Ejemplos:

- 80: HTTP
- 443: HTTPS
- 22: SSH

Git y GitHub

- **Git:** Control de versiones para rastrear cambios.
- **GitHub:** Plataforma para almacenar proyectos.

Comandos:

- git add: Añadir archivos al área de preparación.
- git commit: Guardar cambios localmente.
- git push: Enviar cambios al repositorio remoto.

Apuntes Complementarios: Redes y Seguridad

Cifrado en Redes

El cifrado es un proceso que transforma datos legibles en un formato ilegible para proteger la información durante su transmisión. Esto asegura que solo las partes autorizadas puedan acceder al contenido.

Tipos de Cifrado

- **Cifrado Simétrico:** Utiliza la misma clave para cifrar y descifrar los datos (ejemplo: AES).
- **Cifrado Asimétrico:** Usa un par de claves, una pública para cifrar y una privada para descifrar (ejemplo: RSA).

Ejemplo de Cifrado en HTTPS

HTTPS utiliza TLS/SSL para proteger la comunicación. Esto incluye:

1. Autenticación del servidor mediante certificados.
2. Establecimiento de una conexión segura usando cifrado.

Ventajas del Cifrado:

- Protección contra ataques de interceptación.
- Garantía de la integridad de los datos.

Redes Privadas y Seguridad

Redes Privadas Virtuales (VPN)

- **Qué es:** Una conexión cifrada que permite navegar de forma segura.
- **Ventajas:**
  - Oculta tu dirección IP.
  - Protege los datos en redes públicas.
  - Permite acceder a recursos restringidos geográficamente.

Firewalls

- **Qué es:** Un sistema de seguridad que monitorea y controla el tráfico de red basado en reglas predefinidas.
- **Tipos:**
    - Firewalls de hardware.
    - Firewalls de software.
- **Ejemplo de Uso:** Bloquear conexiones entrantes no autorizadas.

Diagnóstico de Redes

Herramientas Comunes

1. **Ping:** Verificar si una máquina es alcanzable.

   Comando: ping direccion_ip

2. **Traceroute:** Rastrear la ruta que toma un paquete para llegar a su destino.

   Comando traceroute direccion_ip

3. **Nmap:** Escanear puertos abiertos y detectar servicios en una red.

   Comando: nmap direccion_ip

Buenas Prácticas en Redes

Políticas de Contraseñas

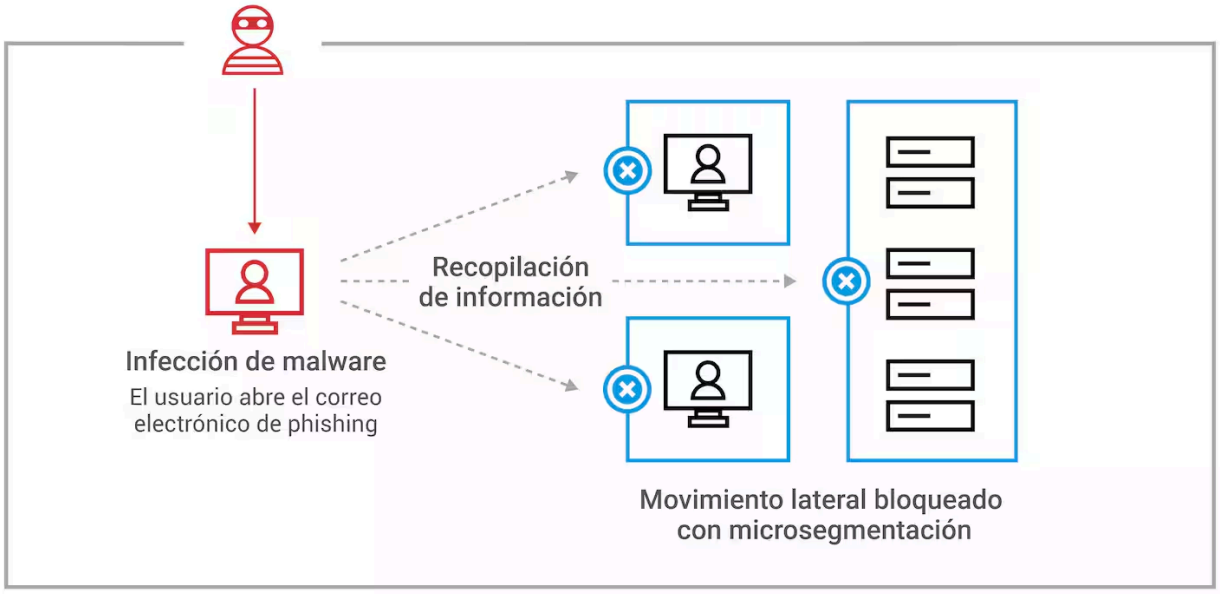
- Utiliza contraseñas largas y complejas.
- Cambia las contraseñas regularmente.

Actualizaciones

- Mantén el sistema operativo y las aplicaciones al día.
- Las actualizaciones incluyen parches de seguridad importantes.

Segmentación de Red

- Divide la red en segmentos para limitar el alcance de posibles ataques.



Cómo evitar el movimiento lateral con microsegmentación

Ejercicios Continuos

UNIX, Libertades y Pastarrufa

1 ¿Qué es, y quien creó UNIX? ¿Por qué es tan importante a día de hoy si es tan antiguo? ¿Qué se creó para crear UNIX? Explica lo más detalladamente que puedas según lo visto en clase y lo que puedas aportar.
2 ¿Quién creó, y qué es, la FSF? ¿Qué es GNU? ¿Y la GPL? Explica las libertades y deberes y porqué no están reñidos con ganar dinero.
3 ¿Qué contemplan las patentes de software y porqué no son válidas en otros países que
no sean EEUU y Japón?
4 ¿Qué es Linux y quién lo creó? ¿Por qué es tan importante a día de hoy? ¿Qué creó el
autor de Linux?
5 ¿Qué es "ping"?

1.
Las bases del sistema operativo fueron diseñadas por Ken Thompson y Dennis Ritchie en 1973. No fue hasta 1975 que el MIT diseñó el primer modelo funcional.

Sigue siendo importante ya que estableció la base sobre la que después se establecieron sistemas más modernos además de al ser una especificación abierta cada individuo puede aportar con tal de hacerla crecer.

2.
FSF significa Free Software Foundation (Fundación del software libre) y fue creada por Richard Stallman en 1985. El mismo Stallman también fundó el proyecto GNU, la creación de un sistema operativo completamente libre y compatible con UNIX. La GPL (GNU Public License) también creada por Richard Stallman es una licencia de software libre con el objetivo de proteger la libertad de los usuarios y desarrolladores de software.

Las libertades de la FSF son las siguientes:
● Libertad para usar el programa: Se refiere a poder usar el programa para cualquier propósito sin restricciones y para cualquier persona.
● Libertad para estudiar cómo funciona el programa: La capacidad de estudiar como esta compuesto el programa y aplicar mejoras
● Libertad para distribuir copias: La capacidad para compartir el software con otra gente.
● Libertad para mejorar el programa y compartir las mejoras: La capacidad de mejorar tu software o el de otro y compartir las mejoras.

3.
Las patentes de software son recursos legales que otorgan a los creadores derechos exclusivos sobre sus diseños relacionados con programas informáticos.
Las patentes de software no son válidas en otros países ya que no se considera una invención per se sinó más bien una forma de solucionar un problema. Linux es un sistema operativo de código abierto basado, que fue creado por Linus Torvalds.

4.

Linux es importante porque permite a los usuarios modificar y distribuir el software, de manera que va creciendo poco a poco de manera natural. Tantos años de progreso hacen que sea un sistema muy resistente y estable y además es aplicado a un montón de sectores y dispositivos distintos.

Linus Torvalds también creó Git, un sistema para controlar versiones.

5.

Ping (literal): es un comando que se utiliza para probar la conectividad entre dos dispositivos en una red. Envía un paquete de datos (ping) al otro dispositivo y cuenta el tiempo que tarda en recibir una respuesta en concreto (pong).

CLIENTES Y SERVIDORES

Este trabajo está hecho gracias a la participación de mis compañeros de clase ya que no pude asistir y me pasaron los apuntes.

¿Por qué el diseño de "internet" tal y como lo conocemos hoy es tan interesante y eficiente? ¿De dónde viene?

Internet es interesante porque permite que millones de dispositivos en todo el mundo se conecten y compartan información súper rápido. Su diseño es eficiente porque usa protocolos que aseguran que los datos lleguen bien, sin importar la distancia o el tipo de red. Viene de investigaciones en los años 60 y 70, originalmente como un proyecto del ejército de Estados Unidos llamado ARPANET.

¿Qué protocolos hemos visto en clase? ¿Por qué algunos de ellos acaban en TP?
Esta es la lista de protocolos que hemos dado en clase.

HTTP: Protocolo base para las comunicaciones en la web.

FTP: Para la transferencia de archivos.

SMTP: Para el envío de correos electrónicos.

HTTPS (443): Versión segura de HTTP.

SFTP (990): FTP con seguridad añadida.

MySQL (3306): Protocolo para bases de datos.

SMTPS (465): Envío seguro de correos electrónicos.

IMAP (143) y IMAPS (993): Protocolo para la sincronización de correos.

POP3 (110 y 995): Protocolo para la descarga de correos.

CIFS (445, 139, 137 y 138): Para compartir carpetas.

TP es transfer protocol, protocolos de transferencia.

¿HTTP y HTML... por qué se dice que son unos de los inventos más importantes de toda
la historia?
Se dice que HTTP y HTML son de los inventos más importantes porque son la base de la web. HTTP es el idioma que permite que los navegadores se comuniquen con los servidores para cargar páginas. HTML es el lenguaje en que se escriben esas páginas para que se vean como sitios web.

¿Qué es un "Puerto" para un protocolo y para qué sirven?
Un puerto es como una puerta numerada en un dispositivo que ayuda a organizar y separar diferentes tipos de conexiones o servicios. Para un protocolo, los puertos permiten que varios servicios puedan funcionar al mismo tiempo sin mezclarse.

¿Qué es un cliente? ¿Y el servidor?
Un cliente es el dispositivo o software que hace una solicitud, y el servidor es el que responde a esa solicitud. Son necesarios porque el cliente pide datos o servicios, y el servidor los provee,como una relación de "pregunta-respuesta" o "solicitud-entrega".

¿Qué protocolos utiliza Whatsapp para hacer sus funcionalidades?

Signal protocol: Para el cifrado de extremo a extremo
HTTPS: No lo entiendo

XMPP: Se utiliza para la gestión de envío de mensajes a tiempo real

RTP: Se usa para las transmisiones de audio y vídeo.

Fuentes:
https://www.seguridadofensiva.com/2013/12/ntendiendo-el-protocolo-de-whatsapp-fun
https://en.wikipedia.org/wiki/XMPP

https://datatracker.ietf.org/doc/html/rfc3550

https://signal.org/blog/whatsapp-complete/

Puertos Abiertos

1. **Explica lo más detalladamente posible todas las respuestas, incluyendo esquemas,**
   capturas y demostraciones de todo lo implicado. Entregad un PDF. La presentación y el
   detalle de las respuestas será valorado.
2. **Hemos visto cómo trabajar con servidores... ¿qué es indispensable que esté activo para iniciar una comunicación en un servicio? ¿Cómo funcionan las conexiones cliente-servidor de diversos protocolos? Utiliza los protocolos HTTP, SSH y SMTP (indica puertos que se utilizan y si puedes dibuja un pequeño esquema de comunicación)**

Es indispensable que el cliente y el servidor (o los dos servidores) estén conectados ya sea por internet o por una red local, hay que conectarse al puerto correcto con el protocolo correspondiente y el servidor tiene que estar activo y escuchando en el puerto en concreto.

- **HTTP** en el puerto 80
- **SSH** en el puerto 22
- **SMTP** en el puerto 25

Referencia: https://ingenieraupoliana.blogspot.com/2010/12/cliente-servidor.html

3. **¿Por qué, si lo intentamos, no podemos abrir la comunicación para escuchar un puerto estándar? ¿Qué debemos hacer para poder hacerlo? ¿Por qué debemos hacerlo de esta manera?**

La mayoría de los puertos estándar están reservados para usuarios administradores o el propio Root, para poder acceder a ellos tendríamos que hacerlo desde estos usuarios, se hace desde esta manera por temas de seguridad. Hay una manera también de redirigir otro puerto para que funcione por un puerto estándar ¿?

Protocolos y prefijos

¿Qué es Git? ¿Y github? Explica para qué sirve y porqué se utiliza tanto en la industria, además de los parámetros básicos de git para el uso habitual de clase. ¿Por qué necesitamos generar las claves SSH?

Git es un sistema de control de versiones, Github es una plataforma en la que puedes almacenar, compartir y distribuir junto con otros usuarios proyectos y código. Sirve sobre todo para mantener un seguimiento de los cambios y navegar por las distintas versiones de un proyecto con facilidad. Es muy útil también para encontrar repositorios, proyectos o personas que te puedan ser de utilidad a ti o a tu proyecto.

Los comandos mas habituales son:git add, git commit, git push, git pull.
Se generan las claves como sistema de seguridad, cada usuario tiene una clave privada y una pública. Al compartir algo intercambiais las claves públicas y de esta manera se consigue un acceso seguro.

¿Por qué los mensajes de error son tan importantes en la comunicación de protocolos? ¿Por qué algunos son... "positivos"? ¿Qué significan los errores 404, 400, 200, 500 y 418 en HTTP?

Los mensajes de error ayudan a comprender que parte del proceso no se ha realizado con tal de entender mejor el problema y poder solucionarlo. Los errores positivos sencillamente comunican que todo se ha realizado correctamente.
Errores y que significan:
404 (Page not found): Este error indica que el URL o la dirección indicada no se ha encontrado.
400 (Bad request): Este error indica que la URL o la dirección no está escrita como tendría que ser y contiene algún error léxico.
200 (ok): Este error indica que todo se ha realizado correctamente.
500: Este error indica que algo malo ha sucedido en el servidor donde se alojan los archivos de tu sitio web.
418 (Soy una tetera): El servidor se rehúsa a preparar café ya que es una tetera .

Con vim haz dos scripts, "client.sh" y "server.sh". En server, escuchad el puerto 2022. Para el cliente, investigad cómo con el comando "nc" podéis enviar un mensaje utilizando echo... y escribid la línea que permita enviar datos a la propia máquina (al fin y al cabo, el server y el cliente están en la misma máquina). Ejecutad dos sesiones de putty conectadas a la misma máquina virtual, en una editad y ejecutad el server y en la otra el cliente y comprobad cómo se comunican entre sí.

La parte de conectarme con Putty en dos sesiones para comprobar si funcionan los scripts no la he podido realizar, he intentado configurar el Putty en mi ordenador pero creo que tengo algo mal.

Referencias:
https://www.ochobitshacenunbyte.com/2021/11/04/uso-del-comando-ncat-nc-en-linux-c-ejemplos/