Actividad 1: Introducción a la Gestión de Servicios.

1.1. Introducción a la Gestión de Servicios en Sistemas Operativos

La **Gestión de Servicios** en sistemas operativos es una función esencial que permite administrar y controlar procesos de fondo (o en segundo plano) que proporcionan funcionalidades específicas en un sistema. Tanto Windows como Linux ofrecen herramientas y conceptos clave para la gestión eficiente de servicios. Esta Actividad aborda los aspectos teóricos y conceptuales de la gestión de servicios en ambos sistemas operativos**.**

1.2. Conceptos Fundamentales de los Servicios

Definición de Servicio

Un servicio es un programa o conjunto de procesos que se ejecutan en segundo plano para realizar tareas específicas, como la gestión de redes, la impresión, o la seguridad del sistema. Los servicios son fundamentales para garantizar el funcionamiento continuo y eficiente del sistema operativo y sus aplicaciones.

Ciclo de Vida de un Servicio

El ciclo de vida de un servicio incluye los estados de inicio, ejecución, detención y reinicio. La gestión adecuada de estos estados es crucial para garantizar la disponibilidad y el rendimiento del sistema.

1.3 Gestión de Servicios en Windows

Herramientas Principales

Administrador de Servicios: Permite a los administradores ver, iniciar, detener y configurar servicios desde una interfaz gráfica.

Línea de Comandos (sc.exe): Ofrece un control avanzado de los servicios mediante comandos.

PowerShell: Proporciona cmdlets específicos como Get-Service, Start-Service y Stop-Service para la administración de servicios.

Características Clave

Tipos de Inicio: Los servicios en Windows pueden configurarse para iniciarse manualmente, automáticamente o estar deshabilitados.

Dependencias: Los servicios pueden depender de otros servicios o recursos, lo que requiere una gestión cuidadosa para evitar conflictos.

Monitoreo y Registro

El Visor de Eventos de Windows permite revisar registros relacionados con el comportamiento y errores de los servicios, facilitando la resolución de problemas.

1.4 Gestión de Servicios en Linux

Herramientas Principales

Systemd: Es el sistema de init más utilizado en distribuciones modernas. Los comandos systemctl permiten gestionar servicios, como iniciar (systemctl start), detener (systemctl stop) o verificar el estado (systemctl status).

SysVinit y Upstart: Sistemas de init anteriores que aún se encuentran en algunas distribuciones más antiguas.

Archivos de Configuración

Los servicios en Linux suelen estar definidos mediante archivos de configuración ubicados en directorios como /etc/systemd/system/. Estos archivos especifican cómo y cuándo se ejecutan los servicios.

Características Clave

Runlevels y Targets: En distribuciones basadas en Systemd, los targets definen grupos de servicios que deben ejecutarse en diferentes escenarios, como el inicio del sistema o el modo de recuperación.

Permisos y Seguridad: Linux utiliza permisos estrictos para garantizar que los servicios sensibles solo puedan ser gestionados por usuarios autorizados, como el administrador root o usuarios con privilegios elevados.

Monitoreo y Registro

Linux ofrece herramientas como journalctl para analizar registros de eventos relacionados con servicios. Esto es útil para identificar y solucionar problemas.

1.5. Comparación entre Windows y Linux

Característica	Windows	Linux
Herramienta Principal	Administrador de Servicios	Systemd (en la mayoría de distribuciones)
Configuración	GUI y Registro	Archivos de texto en /etc
Seguridad	Basada en ACLs y permisos NTFS	Basada en permisos POSIX y SELinux/AppArmor
Monitoreo	Visor de Eventos	journalctl, logs en /var/log

Consideraciones Clave para el Aprendizaje Inicial

Comprender la Arquitectura de los Servicios
Es fundamental entender cómo interactúan los servicios con el sistema operativo y otros componentes.

Familiarizarse con las Herramientas Disponibles
Tanto en Windows como en Linux, las herramientas de gestión son el punto de partida para aprender a controlar y monitorear servicios.

Seguridad en la Gestión de Servicios
Asegurarse de gestionar servicios con los permisos adecuados y minimizar riesgos de seguridad.

Práctica Regular
Configurar, iniciar y detener servicios en entornos de prueba ayuda a consolidar conocimientos teóricos.

1.6. Utilización del paquete de Software Libre XAMPP para la creación de un escenario para la gestión de servicios de Sistemas Operativos

XAMPP es un potente instalador todo incluido con algunas extensiones y herramientas preconfiguradas, además es un paquete de servidor web de código abierto. Contiene un servidor web Apache, una base de datos MySQL, PHP y Perl compatible con todas las plataformas, todos los componentes principales que se requieren para crear un servidor web local para probar sus aplicaciones web basadas en PHP.

Instalación de XAMPP para Linux.

Pasos para instalar XAMPP en Ubuntu

Paso 1: Descargar el paquete XAMPP

https://www.apachefriends.org/ y descargue la versión correspondiente a su sistema operativo.

Si está en windows lance el instalador .

si tiene linux primero hará lo siguiente .

Paso 2: Hacer ejecutable el archivo descargado

El paquete debe instalarse a través de la línea de comandos de Ubuntu, es decir, la terminal. Puedes pulsar directamente Ctrl+ALt+T para abrir el terminal. Ahora mueve la carpeta de descarga para acceder al archivo. El siguiente comando puede ayudarle a descargar el archivo.

$ cd Downloads ↵ (pulse intro)

Asegúrate de poner en mayúsculas «Descargas», de lo contrario te mostrará un mensaje de error «no such file or directory».
con ls -l puede ver todos los archivos en eta carpeta/ directorio

Antes de seguir utilizando el paquete, hay que hacerlo ejecutable. Y para ello ejecute el siguiente comando.

$ sudo chmod +x xampp-linux-x64-7.3.10-1-installer.run ↵ (pulse intro)

Los valores 7.3.10.1 pueden cambiar ya que indican la version

Si estás usando una versión diferente de XAMPP, puedes reemplazar el número de versión con el número de versión de XAMPP que has descargado
Escriba el comando de instalación para seguir adelante.

$ sudo ./xampp-linux-x64-7.3.10-1-installer.run ↵ (pulse intro)

Paso 3: Iniciar el asistente de instalación

Instalación de XAMPP
Ahora aparecerá la ventana de instalación. Basta con seguir las instrucciones de instalación. Y cuando vea la siguiente ventana pulse el botón 'Siguiente'

Paso 4. Inicie XAMPP desde el terminal

XAMPP no tiene ningún archivo en el escritorio, por lo que hay que lanzarlo desde el terminal. Sin embargo, lanzar XAMPP desde el terminal no es tan difícil. Sólo tiene que ejecutar el siguiente comando.

Escriba el comando 'Open' para ejecutar el instalador de XAMPP cambie el directorio a lampp.

$ cd /opt/lampp
$ sudo ./manager-linux-x64.run ↵ (pulse intro)

Asegurese de instalar solo Apache como Web server y Mysql como base de datos no necesita otros componentes.

1.7. Que son las API y Como es su Funcionamiento

Una API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de las aplicaciones.

Las API permiten que aplicaciones y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados.
Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero.

La documentación de las API indica como deben ser las solicitudes y como serán las respuestas de manera que se emula un contrato o protocolo de funcionamiento lo que se convierte en la base de una arquitectura de aplicaciones de microservicios a través de las API.

Las API son un medio simplificado para extender aplicaciones, existen API públicas que aportan un mucho valor, un ejemplo conocido es la API de Google Maps.

Por ejemplo, imagina una empresa distribuidora que permite a los clientes acceder a los datos con una API que les ayude a añadir información en los pedidos que realizan como los códigos internos de productos o disponibilidad de inventario

Con una API disponible de forma pública, los desarrolladores que trabajan para la distribuidora, los vendedores o los terceros podrían desarrollar una aplicación para ayudar a encontrar los productos que necesiten. Esto podría dar como resultado mayores ventas u otras oportunidades comerciales.

En resumen, las API te permiten habilitar el acceso a tus recursos, lo que implica al mismo tiempo establecer condiciones de seguridad para regular la forma en que se permite el acceso y a quien.

La seguridad de las API depende de una buena gestión, lo cual incluye el uso de una puerta de enlace o GATEWAY de API.

Políticas de lanzamiento de las API
Privado
Las API solo se pueden usar internamente, así que las empresas tienen un mayor control sobre ellas. Esto les da a las empresas un mayor control sobre sus API.

De partners
Las API se comparten con partners empresariales específicos, lo cual puede ofrecer flujos de ingresos adicionales, sin comprometer la calidad. Esto puede proporcionar flujos de ingreso adicionales, sin comprometer la calidad.

Público
Todos tienen acceso a las API. Esto permite que terceros desarrollen aplicaciones que interactúan con tu API, y puede ser un recurso para innovar.
Las API compartidas, ya sea con los partners elegidos o con todo el mundo, tienen efectos positivos. Mientras más te asocies con otros gracias a las API, mayor difusión obtendrá tu marca, independientemente de los esfuerzos publicitarios de la empresa. Si utilizas una API pública, por ejemplo, para dar acceso a la tecnología a todo el mundo, alienta a los desarrolladores a crear un ecosistema de aplicaciones en torno a tu API. Mientras más personas usen tu tecnología, más personas estarán dispuestas a hacer negocios contigo. Hacer pública la tecnología da resultados novedosos e inesperados que a veces alteran sectores completos, también requiere disponer de suficientes recursos para atender todos los pedidos que pudieran generarse y responder adecuadamente.

Explicación de la seguridad de la API
La seguridad de las API se refiere al conjunto de prácticas y productos que una organización utiliza para prevenir los ataques maliciosos y el uso indebido de las API. Dada la complejidad de los ecosistemas de API, el crecimiento de las plataformas IoT y el gran volumen de API que utilizan las organizaciones, controlar la seguridad de las API es cada vez más difícil y necesario.

Las API se sitúan entre los recursos informáticos de una organización y los desarrolladores de software de terceros, y entre los recursos informáticos y las personas, proporcionando datos e información en los endpoints del proceso. Es en estos endpoints donde los datos de la empresa y de los usuarios son vulnerables a diversos tipos de ataques y riesgos de seguridad, entre ellos:

Ataques basados en la autenticación: en los que los hackers intentan adivinar o robar las contraseñas de los usuarios o se aprovechan de procesos de autenticación débiles para acceder a los servidores API.

Ataques de intermediario: en los que un ciberdelincuente roba o modifica datos (por ejemplo, credenciales de inicio de sesión o información de pago) interceptando solicitudes y/o respuestas entre la API.

Inyecciones de código/ataques de inyección: donde el hacker transmite un script dañino (para insertar información falsa, eliminar o revelar datos, o interrumpir la funcionalidad de la aplicación) a través de una solicitud de API, explotando las vulnerabilidades de los intérpretes de API que leen y traducen los datos.

Ataque de denegación de servicio (DoS): estos ataques envían decenas de peticiones API para colapsar o ralentizar el servidor. Los ataques DoS a menudo pueden provenir de múltiples atacantes simultáneamente en lo que se denomina un ataque de denegación de servicio distribuido (DDoS).

Ataques de autorización de nivel de objeto rota (Broken Object Level Authorization, BOLA): se producen cuando los ciberdelincuentes manipulan identificadores de objetos en los endpoints de la API para obtener acceso no autorizado a los datos del usuario. Este problema surge cuando un endpoint de la API permite a un usuario acceder a registros a los que normalmente no debería. Los ataques BOLA son especialmente comunes, porque implementar comprobaciones de autorización adecuadas a nivel de objeto puede ser difícil y llevar mucho tiempo.

Estos y otros tipos de ciberataques son prácticamente inevitables en el dinámico panorama de TI actual. Además, con la proliferación de los ciberdelincuentes y su acceso a tecnologías de piratería más sofisticadas, la aplicación de protocolos de seguridad de las API será cada vez más crucial para la seguridad de los datos empresariales.

Buenas prácticas de seguridad de API

Las API permiten a las empresas agilizar la integración entre sistemas y el intercambio de datos, pero esta interconectividad conlleva una mayor exposición a los ciberataques. De hecho, la mayoría de los hackeos de aplicaciones móviles y web atacan las API para acceder a los datos de la empresa o de los usuarios. Las API pirateadas o comprometidas pueden provocar vulneraciones de datos catastróficas e interrupciones del servicio que pongan en peligro datos personales, financieros y médicos confidenciales.

Afortunadamente, los avances en la seguridad de las API permiten prevenir o mitigar el impacto de los ciberataques de actores malintencionados. Estas son 11 prácticas y programas de seguridad de API comunes que las organizaciones pueden aprovechar para proteger los recursos informáticos y los datos de los usuarios:

Pasarelas API. Instalar una pasarela API es una de las formas más sencillas de restringir el acceso a la API. Las pasarelas crean un único punto de entrada para todas las solicitudes de API y actúan como capa de seguridad aplicando políticas de seguridad, ayudando a estandarizar las interacciones de la API y ofreciendo funciones como la transformación solicitud/respuesta, el almacenamiento en caché y el registro.

Autenticación y autorización sólidas. El uso de protocolos de autenticación estándar del sector (como OAuth 2.0, claves API, JWT, OpenID Connect, etc.) garantiza que sólo los usuarios autenticados puedan acceder a las API empresariales. Además, la aplicación de controles de acceso basados en funciones impide que los usuarios accedan a recursos que no están autorizados a utilizar.
Protocolos de cifrado. La conexión SSL o los protocolos de cifrado TLS, como HTTP Secure (HTTPS), ayudan a los equipos a proteger la comunicación entre la API y las aplicaciones del cliente. HTTPS cifra todas las transmisiones de datos de la red, lo que impide el acceso no autorizado y la manipulación. Cifrar los datos en reposo, como las contraseñas almacenadas, puede proteger aún más los datos sensibles mientras están almacenados.

Firewalls de aplicaciones web (WAF). Los WAF proporcionan una capa adicional de protección para las API de las empresas, especialmente frente a los ataques habituales a las aplicaciones web, como los ataques de inyección, los ataques de secuencias de comandos en sitios cruzados (XSS) y la falsificación de solicitudes en sitios cruzados (CSRF). El software de seguridad WAF puede analizar las solicitudes API entrantes y bloquear el tráfico malicioso antes de que llegue al servidor.

Los cinco principales métodos de integración de la seguridad en un diseño de API basada en REST son:

Utilizar en todo momento el cifrado TLS.

Implementar un modelo de autenticación y autorización sólido y adaptable.

No incluir información confidencial en las URL.

Definir explícitamente las solicitudes y respuestas de API RESTful permitidas.

Aplicar funciones de detección continua de API.

Utilizar en todo momento el cifrado TLS.

Buenas prácticas para el uso de claves de aplicación/Mejores prácticas para la seguridad de las claves API

1. Nunca comparta su clave API
Manténgala confidencial: Al igual que no compartiría su contraseña personal, no comparta su clave API. Si alguien necesita acceso a la API, debe obtener su propia clave.

No comparta su clave en foros públicos: No incluya su clave API en discusiones públicas, correos electrónicos o tickets de soporte, incluso entre usted y Anthropic.

Tenga precaución con las herramientas de terceros: Considere que cuando carga su clave API en herramientas o plataformas de terceros (como un IDE basado en web, proveedor de nube o plataforma CI/CD), está dando al desarrollador de esa herramienta acceso a su cuenta de Anthropic. Si no confía en su reputación, no confíe en ellos con su clave API.

Cuando use un proveedor de terceros, siempre agregue su clave API como un secreto encriptado. Nunca la incluya directamente en su código o archivos de configuración.

2. Monitoree de cerca el uso y los registros
Recomendamos revisar regularmente los registros y los patrones de uso de sus claves API dentro de Console.

Para organizaciones de Scale API: Implemente límites de uso y gasto en la configuración de su cuenta.

Estos límites actúan como una salvaguarda contra el uso inesperado debido a claves filtradas o scripts erróneos.

Para organizaciones de Build API: Habilite y configure los ajustes de recarga automática en su cuenta.

Esta función le permite establecer un umbral en el que su cuenta cargará automáticamente la tarjeta registrada para reponer los créditos de uso.

Considere cuidadosamente los límites de recarga automática. Si bien aseguran un servicio continuo, también actúan como una salvaguarda contra un uso alto inesperado que podría resultar de claves filtradas o errores en su código.

3. Manejo seguro de claves API con variables de entorno y secretos
Una mejor práctica para manejar de forma segura las claves API es usar variables de entorno para inyectar y compartir variables de entorno de manera segura. Cuando implementa su aplicación en un entorno en la nube, puede usar su solución de gestión de secretos para pasar de forma segura la clave API a su aplicación a través de una variable de entorno sin compartir inadvertidamente su clave API.

Si está almacenando secretos localmente usando dotenv, debe agregar sus archivos .env al archivo de ignorar del control de fuentes (por ejemplo, gitignore para git) para evitar distribuir inadvertidamente información sensible públicamente. En entornos en la nube, prefiera el almacenamiento de secretos encriptados en lugar de archivos dotenv.

Ejemplo en Python:

1. Cree un archivo .env en el directorio de su proyecto.

2. Agregue su clave API al archivo .env:

ANTHROPIC_API_KEY=su-clave-api-aquí

3. Instale el paquete python-dotenv:

pip install python-dotenv

4. Cargue la clave API en su script de Python:

from dotenv import load_dotenvimport osload_dotenv()my_api_key = os.getenv("ANTHROPIC_API_KEY")

5. Si está implementando su aplicación en un entorno de alojamiento en la nube, consulte la documentación de su proveedor de nube sobre cómo agregar su clave API de Anthropic y compartirla con su aplicación (AWS, GCP, Azure, Vercel, Heroku). Algunos proveedores ofrecen múltiples formas de inyectar de manera segura variables de entorno en su aplicación.

4. Rote las claves API regularmente
Rote regularmente sus claves API en un horario consistente (por ejemplo, cada 90 días) creando nuevas y desactivando las antiguas. Esta rutina ayuda a minimizar los riesgos potenciales si una clave se ve comprometida alguna vez.

5. Use claves separadas para diferentes propósitos
Si es posible, use diferentes claves API para entornos de desarrollo, prueba y producción. De esta manera, puede correlacionar su uso con diferentes casos de uso internos. Si su clave API se ve comprometida, esto le permite deshabilitar rápidamente solo ese caso de uso y limitar cualquier daño potencial.

6. Escanee los repositorios en busca de secretos
Revise regularmente sus repositorios de control de fuentes en busca de secretos cometidos accidentalmente.