Laboratorio 2

Deivis J. Castro

TALLER – Diferenciar entre Confidencialidad, Integridad y Disponibilidad

Paso 1: Definir los Términos

Confidencialidad: Se refiere a la protección de la información contra accesos no autorizados. Garantiza que solo las personas con permisos adecuados puedan ver o modificar los datos. Se logra mediante técnicas como el cifrado, autenticación y control de acceso

Integridad: Asegura que los datos sean precisos, completos y no hayan sido alterados de manera no autorizada. Se utilizan mecanismos como el hashing y las firmas digitales para verificar que la información no ha sido modificada

Disponibilidad: Se enfoca en garantizar que los sistemas y datos estén accesibles cuando se necesiten. Se logra mediante redundancia, copias de seguridad y protección contra ataques que puedan interrumpir el servicio, como los ataques DDoS

Paso 2: Proporcionar Ejemplos

2.1 Ejemplo de confidencialidad: Una empresa de salud protege los historiales médicos de los pacientes mediante cifrado y autenticación de usuarios. Solo el personal autorizado puede acceder a los registros, evitando filtraciones de datos sensibles.

Ejemplo de integridad: Un banco utiliza firmas digitales y funciones de hash para garantizar que las transacciones no sean alteradas. Si un cliente transfiere dinero, el sistema verifica que los datos no hayan sido modificados antes de completar la operación.

Ejemplo de disponibilidad: Un servicio de comercio electrónico implementa servidores redundantes y copias de seguridad para garantizar que su plataforma esté operativa 24/7. Esto evita interrupciones en compras y pagos, asegurando una experiencia fluida para los usuarios.

Análisis Comparativo:

Actividad:

− ¿Cómo un fallo en la confidencialidad puede comprometer la integridad de los datos?

R: Cuando la confidencialidad se ve comprometida, los datos pueden ser accesibles para actores malintencionados. Si un atacante accede a información sensible sin autorización, puede modificarla, alterando su integridad. Por ejemplo, en un sistema financiero, si una brecha permite a un intruso modificar registros de transacciones, la información deja de ser confiable.

− ¿Cómo la falta de disponibilidad puede afectar la integridad y confianza en los datos?

R: La falta de disponibilidad impide que los usuarios accedan a la información cuando la necesitan. Esto puede provocar pérdida de datos o inconsistencias si los sistemas no están

operativos para validar, actualizar o proteger la información. Por ejemplo, en una base de datos de pacientes, si el sistema no está disponible y los registros deben ingresarse manualmente, hay un riesgo de errores y pérdida de información, afectando su integridad y fiabilidad.

Preguntas de Reflexión:

• Pregunta 1: ¿Qué concepto consideras más crítico en una empresa de salud? ¿Y en una empresa de comercio electrónico?

R: En una empresa de salud, la integridad es esencial para garantizar que los datos médicos sean precisos y no alterados, pues decisiones médicas dependen de ellos. Además, la confidencialidad es fundamental para proteger la privacidad de los pacientes.

En una empresa de comercio electrónico, la disponibilidad es crucial, ya que los sistemas deben estar operativos en todo momento para procesar ventas, pagos y logística. La confidencialidad también es clave para proteger datos financieros de los clientes.

• Pregunta 2: ¿Cómo podrías priorizar la implementación de estos conceptos en una organización con recursos limitados?

R:

-Autenticación multifactorial (MFA) para fortalecer la seguridad sin grandes costos.

-Cifrado de datos en bases de datos y correos electrónicos.

-Capacitación en seguridad informática para empleados, evitando brechas por error humano.

-Respaldo y redundancia de datos para garantizar disponibilidad y recuperación ante fallas.

-Uso de herramientas gratuitas o de código abierto para monitoreo y protección de sistemas.

LABORATORIO 2 SEGUNDA PARTE

1.Definir los tipos de malware:

-Virus: Programa malicioso que se adjunta a archivos legítimos y se propaga cuando el archivo infectado es ejecutado. Puede dañar sistemas, eliminar archivos o robar información.

-Gusano: Similar a un virus, pero se propaga automáticamente sin necesidad de intervención del usuario. Puede consumir recursos del sistema y causar fallos en redes.

-Troyano: Se disfraza de software legítimo para engañar a los usuarios y permitir el acceso no autorizado a sistemas. Puede instalar puertas traseras para ataques futuros.

-Ransomware: Bloquea el acceso a archivos o sistemas y exige un pago (rescate) para restaurarlos. Es una de las amenazas más peligrosas en ciberseguridad.

-Spyware: Diseñado para espiar la actividad del usuario sin su conocimiento, recopilando información como contraseñas, datos bancarios y hábitos de navegación.

2. Proporcionar Ejemplos de Cómo Afectan a los Sistemas:

-Virus:

*Se adjunta a archivos legítimos y, al ejecutarse, puede corromper datos, eliminar archivos esenciales del sistema o modificar configuraciones críticas.

*Puede ralentizar el rendimiento del equipo al consumir recursos innecesarios.

*Algunos virus desactivan programas de seguridad, dejando el sistema vulnerable a otros ataques.

-Gusano:

*Se propaga automáticamente por redes sin intervención del usuario, saturando el ancho de banda y causando fallos en la conectividad.

*Puede replicarse en múltiples dispositivos dentro de una red, afectando servidores y estaciones de trabajo.

*Algunos gusanos instalan puertas traseras para permitir accesos no autorizados.

-Troyano:

*Se disfraza de software legítimo y, una vez instalado, permite el acceso remoto a atacantes.

*Puede robar credenciales de usuario, como contraseñas y datos bancarios.

*Algunos troyanos instalan otros tipos de malware, como ransomware o spyware.

-Ransomware:

*Cifra archivos del sistema y exige un pago para desbloquearlos, dejando al usuario sin acceso a su información.

*Puede afectar redes empresariales, paralizando operaciones y causando pérdidas económicas.

*Algunos ransomware eliminan archivos si el rescate no se paga en un tiempo determinado.

-Spyware:

*Se ejecuta en segundo plano, registrando pulsaciones de teclado para capturar contraseñas y datos sensibles.

*Puede monitorear la actividad del usuario, recopilando información sobre hábitos de navegación y comunicaciones.

*Algunos spyware desvían datos a servidores externos sin que el usuario lo note.

CODIGO PRACTICO GOOGLE COLAB:

Ejemplo De Cifrado:

```
from cryptography.fernet import Fernet

# 1. Generar una clave

clave = Fernet.generate_key()

fernet = Fernet(clave)

# 2. Texto que quieres cifrar

mensaje_original = "Prueba de cifrado simetrico2"

# 3. Cifrar el mensaje

mensaje_cifrado = fernet.encrypt(mensaje_original.encode())

print("Mensaje cifrado:", mensaje_cifrado)

# 4. Descifrar el mensaje

mensaje_descifrado = fernet.decrypt(mensaje_cifrado).decode()

print("Mensaje descifrado:", mensaje_descifrado)
```

Resultados:

Mensaje cifrado: b'gAAAAABoL3LAoqc8sRDIeYxvUWdS0W6o-TV-Zmr7rviFkM5heNOQUqL-ZfpRlz5ARt7mtHSIAkiG6w2C12ZECkwW0pin0oUEEorGAhovel-NcZ0XnObkvgY='

Mensaje descifrado: Prueba de cifrado simetrico2

CIFRADO ASIMETRICO

```
#EJEMPLO ASIMETRICO

from cryptography.hazmat.primitives.asymmetric import rsa, padding

from cryptography.hazmat.primitives import hashes, serialization


# 1. Generar par de claves (privada y pública)

clave_privada = rsa.generate_private_key(

    public_exponent=65537,

    key_size=2048,

)


clave_publica = clave_privada.public_key()



# 2. Texto original
```

```
mensaje_original = b"Este es un mensaje secreto2."


# 3. Cifrar usando la clave pública
mensaje_cifrado = clave_publica.encrypt(
    mensaje_original,
    padding.OAEP(
        mgf=padding.MGF1(algorithm=hashes.SHA256()),
        algorithm=hashes.SHA256(),
        label=None
    )
)


print("Mensaje cifrado:", mensaje_cifrado)
# 4. Descifrar usando la clave privada
mensaje_descifrado = clave_privada.decrypt(
    mensaje_cifrado,
    padding.OAEP(
        mgf=padding.MGF1(algorithm=hashes.SHA256()),
        algorithm=hashes.SHA256(),
        label=None
    )
)
print("Mensaje descifrado:", mensaje_descifrado.decode())
```

Resultados:

Mensaje cifrado:
b'=\x06c\x99L\xa0&\x1csY\x1d+\xc4\x8b\'\x1cU\xa1/T(\xbc\xfd\xf3\xf4\xb7\x08\xc8]\x1cwk\xbd\x0e\xf2\x0e"\xaa\xa4\x0c!\xb2\xdaB\x17\x9d\x02\xa3\x91\x13\xbb\x80wZ\x8b\xbb%\xb6\x1b\n>\x8c(W\xf7f\n\x91t\xf3\xb3Xt\xa1\xdc\xc4<\x8e\x83n\xdb\xff\xf0\xb8\xe6\xd7H.\xdf\xac\x1d\xba\r\x9e>x\xc4\x125"\x89\xadM\x9f\xde\x89\xe1\x19\x8e\x062\xabA8\x9b\x00li%j\xce\x93\xd2\xd1\xf80#\xc3\x13T\x81\xb4\x10?\xf3Bgu\xc2\xb2\n\xbc\xa1\xd3\xf9\xe60\x92*\xecN\x02\xad\x1b\xae8O\xf5\xc8\x81\x8c\xec\x88\x07NG\x86\xe5\xa9?\x12-
\xca*:\'\xeb$\xde\xd5`\xf6\xba\x87\xa51\x9c\x90w\x08)1\r\xcd\xf3s_\xdb\xa5\x0c\x01}\x92\xc8\x958\xa4KVn\n\x84\xdd\xaa\x07\t\xfd0\x1b\x80\xee\xdc\\R\x8d\xc3\x8a\xa3HA\xa1\x84ao\x02\xe7 \xb0x\xa1F\x02\xab\xf4\xe1\x1a\x899y\xa5V.\xb3(\x0b\xaa'

Mensaje descifrado: Este es un mensaje secreto2.

Ejemplo Hashes:

from google.colab import drive

drive.mount('/content/drive')

Resultados:

Mounted at /content/drive

---

import hashlib

ruta_archivo =
'/content/drive/MyDrive/TalentoTech/Ciberseguridad/Laboratorios/Sesion2/ejemplo.txt'

Función para calcular el hash SHA-256 de un archivo

def calcular_hash_archivo(ruta_archivo):

```
    sha256 = hashlib.sha256()
    with open(ruta_archivo, 'rb') as f:
        while chunk := f.read(4096):
            sha256.update(chunk)
    return sha256.hexdigest()


# Hash original generado antes de enviar el archivo
hash_original = calcular_hash_archivo('/content/drive/MyDrive/TalentoTech/Ciberseguridad/Laboratorios/Sesion2/ejemplo.txt')
print(f"Hash original: {hash_original}")
```

Resultados:

Hash original: 68454d54877e77c3f9090d092f4534e471a94a1fec8b2f88fb28b50a834878a0

```
# Simula la recepción del archivo y verificación del hash
hash_recibido = calcular_hash_archivo('/content/drive/MyDrive/TalentoTech/Ciberseguridad/Laboratorios/Sesion2/ejemplo2.txt')
print(f"Hash recibido: {hash_recibido}")


if hash_original == hash_recibido:
    print("Integridad verificada: los datos no han sido alterados.")
else:
    print("Integridad comprometida: los datos han sido modificados.")
```

Resultados:

Hash recibido:
4380cb21feb762d92b76b594d64067e2cafa8e2a81c66086b54a3a578b204f25

Integridad comprometida: los datos han sido modificados.