Universidad Tecnológica Fidel Velázquez
Ingeniería en Redes Inteligentes y Ciberseguridad - Hacking Ético

Resolver el siguiente ejercicio contestando únicamente en las hojas. Enviar un sólo archivo en formato PDF a través de la plataforma Google Classroom. Valor de la actividad: 10 puntos.

Nombre del estudiante	
Fecha de la actividad	
Calificación	

Práctica 1. Cifrado César

Uno de los primeros ejemplos de encriptado fue usado por Julio César. César necesitaba proveer instrucciones escritas a sus generales, pero él no quería que sus enemigos conocieran sus planes si el mensaje caía en manos equivocadas. Como resultado, él desarrolló lo que después se conocería como el Cifrado César.
La idea de éste cifrado es simple (y como resultado, no provee de protección contra algunas técnicas modernas de crackeado). Cada letra en el mensaje original es movida tres posiciones. Como resultado, A se convierte en D, B se convierte en E, C se convierte en F, D se convierte en G, etc. Las últimas tres letras en el alfabeto son devueltas al inicio: la X se convierte en A, Y se convierte en B y Z se convierte en C. Los caracteres que no son letras no son modificadas en el cifrado.
Escriba un script que implemente el Cifrado César. Permita que el usuario inserte el mensaje y el número de posiciones a moverse, y después despliegue el mensaje movido. Tenga en consideración que su programa codifique ambos casos de mayúsculas y minúsculas. Su script debe también soportar valores de movimiento negativo con el fin de usar ambos (positivos y negativos) para codificar los mensajes y descodificarlos.

1. (30 puntos) Ejecute el siguiente código:

Listing 1: Cifrado César

```
SIMBOLOS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890 !?.'

def cifrado_cesar(texto, clave, modo):
    """
    Cifra o descifra un mensaje usando el cifrado Cesar.

    Argumentos:
        texto (str): La cadena de entrada a procesar.
        clave (int): La clave entera para el desplazamiento.
        modo (str): 'cifrar' o 'descifrar'.

    Retorna:
        str: La cadena procesada (cifrada o descifrada).
    """
    if modo not in ['cifrar', 'descifrar']:
        print("Error: El modo debe ser 'cifrar' o 'descifrar'.")
        return ""

    traducido = ""
    for caracter in texto:
        if caracter in SIMBOLOS:
            indice_simbolo = SIMBOLOS.find(caracter)

            if modo == 'cifrar':
                indice_traducido = indice_simbolo + clave
                # Manejar el desbordamiento para el cifrado
                if indice_traducido >= len(SIMBOLOS):
                    indice_traducido -= len(SIMBOLOS)
            elif modo == 'descifrar':
                indice_traducido = indice_simbolo - clave
                # Manejar el desbordamiento para el descifrado
                if indice_traducido < 0:
                    indice_traducido += len(SIMBOLOS)

            traducido += SIMBOLOS[indice_traducido]
        else:
            # Anadir simbolo sin cifrar/descifrar
            traducido += caracter

    return traducido

# --- Ejemplo de Uso ---
```

```
# 1. Cifrar un mensaje
mensaje_original = "Este es un mensaje secreto."
clave_cifrado = 13
mensaje_cifrado = cifrado_cesar(mensaje_original, clave_cifrado, 'cifrar')
print(f"Mensaje Original:  {mensaje_original}")
print(f"Mensaje Cifrado: {mensaje_cifrado}")
print("-" * 20)


# 2. Descifrar el mensaje
mensaje_descifrado = cifrado_cesar(mensaje_cifrado, clave_cifrado, 'descifrar')
print(f"Mensaje Cifrado: {mensaje_cifrado}")
print(f"Mensaje Descifrado: {mensaje_descifrado}")
print("-" * 20)

# 3. Ejemplo del script de fuerza bruta
ejemplo_fuerza_bruta = 'R67rJr6J81Jzr16nwrJ6rp5r72M'
clave_correcta = 13
mensaje_revelado = cifrado_cesar(ejemplo_fuerza_bruta, clave_correcta, 'descifrar')
print(f"Mensaje Cifrado: {ejemplo_fuerza_bruta}")
print(f"Mensaje usando fuerza bruta:  {mensaje_revelado}")
```

2. (30 puntos) Ejecute el siguiente código:

Listing 2: Fuerza bruta para Cifrado César

```
# El mensaje que queremos intentar descifrar
mensaje = 'R67rJr6J81Jzr16nwrJ6rp5r72M'

# Una cadena que contiene todos los caracteres (simbolos) posibles que se pueden cifrar/descifrar.
SIMBOLOS = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890 !?.'

# Iterar a traves de cada clave posible. El numero de claves posibles es la longitud de la cadena SIMBOLOS.
for clave in range(len(SIMBOLOS)):
        # Esta cadena contendra el texto descifrado para el intento de clave actual.
        descifrado = ""

        # Iterar a traves de cada caracter en el mensaje cifrado.
        for caracter in mensaje:
                # Comprobar si el caracter esta en nuestro conjunto de simbolos.
                if caracter in SIMBOLOS:
                        # Encontrar la posicion numerica del caracter en el conjunto de simbolos.
                        indiceSimbolo = SIMBOLOS.find(caracter)
                        # Descifrar el caracter retrocediendo su indice segun el valor de la clave.
                        indiceDescifrado = indiceSimbolo - clave

                        # Manejar el caso de desbordamiento. Si el indice es negativo,
                        # sumamos la longitud del conjunto de simbolos para volver al final.
                        if indiceDescifrado < 0:
                                indiceDescifrado += len(SIMBOLOS)

                        # Anadir el caracter descifrado a nuestra cadena traducida.
                        descifrado += SIMBOLOS[indiceDescifrado]

                else:
                        # Si el caracter no esta en nuestro conjunto SIMBOLOS (como '!'),
                        # anadirlo directamente a la cadena traducida sin cambiarlo.
                        descifrado += caracter

        # Imprimir el numero de clave y el intento de descifrado para esa clave.
        # El usuario puede leer la salida para encontrar la linea que tiene sentido.
        print('Clave #%s: %s' % (clave, descifrado))
```

3. (40 puntos) Viendo que este script puede encontrar el mensaje oculto probando todas las claves posibles en menos de un segundo, ¿qué diferencias clave hacen que este tipo de ataque de "fuerza bruta"sea impráctico contra una contraseña moderna de un servicio real como Netflix o tu correo, y qué medidas de seguridad utilizan esos sistemas para defenderse?

.....

.....

.....

.....

.....

.....

.....

.....

.....

Evaluación del desempeño

Pregunta:	1	2	3	Total
Puntos:	30	30	40	100
Calificación:				