Tareas de Ciberseguridad Automatizadas

version

Noe Rosales || Karla Dominguez || Ivan Reyes || Karen Aidee || Victor Cruz

noviembre 06, 2021

Contenido

Bienvenido a la documentacion de Tareas de Ciberseguridad Automatizadas!

Contenido

1. Codificacion y Decodificacion de Mensajes

Nuestro Codigo trabaja con 2 tipos de cifrado diferente, Transposicion y Cesar. El metodo de Transposicion toma el mensaje y la llave(string) y usa ambas para hacer una matriz y cambiar las columnas por filas.

2. Hackeo de Mensajes

Al igual que la codificacion/decodificacion de mensajes, esta opcion cuenta con 2 maneras diferentes para hackear el mensaje, Transposicion y Cesar.

3. Investigar Organizacion (API KEY)

Para esta opcion se necesita una llave API para funcionar, se tomara la llave y el dominio de la organizacion que desea investigar, el codigo imprimira la informacion recopilada en pantalla y generera un archivo txt con la informacion.

4. Enviar Correos

Para enviar un correo se necesita contar con lo siguiente:

- Tu correo electronico
- Tu contraseña
- El correo al que se deasea mandar el mensaje
- El asunto
- El mensaje
- El nombre de la imagen y el directorio en donde esta ubicada(opcional)

Nota: solo se aceptan cuentas gmail/hotmail/outlook para el usuario.

5. Escaneo de Puertos

En esta opcion se necesitan la IP, el puerto inicial y puerto final y el programa imprimira los puertos abiertos en pantalla.

6. Obtener tu Cache de DNS

Esta opcion es un script de PowerShell, no es necesario ningun parametro para su ejecucion. Una vez seleccionada el script generara 2 archivos .txt en donde la informacion es la misma, lo unico que cambia es que estan organizadas de diferente manera.

Modulos

Message

1. spanishDictionary()
2. englishDictionary()
3. setLanguage()
4. getLanguage()
5. getMessage(message)
6. getMessage()
7. messageWords(sentence)
8. verifyLanguage(sentence)

- spanishDictionary()

  Es una lista de palabras en español que utilizamos para el cifrado de mensajes.

- englishDictionary()

  Es una lista de palabras en ingles que utilizamos para el cifrado de mensajes.

- setLanguage(language)

  Recibimos un dato de tipo entero y los usamos para saber el lenguaje en que sera escrito, puede ser en ingles o en español. Y lo guardamos como boleano en self.language.

- getLanguage()

  Regresamos el boleano.

- setMessage(Message)

  Recibimos el mensaje escrito por el usuario y varificamos que sea un mensaje valido.

- getMessage()

  Regresamos el mensaje.

- messageWords(sentence)

  Recibimos el mensaje y guardamos cada palabra de la oracion en la lista, y regresamos la lista de palabras.

- **verifyLanguage(sentence)**

  Recibimos la oracion y validamos que este escrita en uno de los lenguajes disponibles ingles/español.

Cifrado Cesar

1. verifyKey()
2. codifyMessage()
3. decodeMessage()
4. hackMessage

- verifyMode()

  El modulo recibe datos de tipo entero y los verificamos, esto es necesario ya que utiliza los siguentes modos(opciones).

  1. Codificar
  2. Decodificar
  3. Hackear

- verifyKey()

  El modulo recibe la llave de tipo entero y la verificamos por que solo el modo 1 y 2 necesitan una llave.

- codifyMessage()

  Este es el modo 1 y aqui ciframos el mensaje con el metodo Cesar.

- decodeMessage()

  Este es el modo 2 y aqui decodificamos el mensaje con el metodo Cesar.

- hackMessage()

  Este es el modo 3 y aqui tratamos de hackear el mensaje con numeros del 1 al 25, cada numero represenata una letra del alfabeto.

Transposition

1. setKey(key)
2. getKey()
3. encodeMessage()
4. unencodeMessage()
5. hackMessage()

- setKey(key)

  Recibimos la llave de tipo string, tomamos la longitud de la llave y lo guardamos en esa variable.

- getKey()

  Regresamos la longitud de la llave.

- encondedMessage()

  Tomamos la transposicion para codificar el mensaje, tomamos la llave y el mensaje.

- unencodedMessage()

  Tomamos la transposicion para decodificar el mensaje, tomamos la llave y el mensaje.

- hackMessage()

  Tomamos la transposicion para hackear el mensaje, tomamos la llave y el mensaje.

Email

1. setOpc(opc)
2. getOpc()
3. setEmailAccount(email_account)
4. getEmailAccount()
5. setPassword
6. getPassword()
7. setTo(to)
8. getTo()
9. setPicture(picture, directory)
10. getPicture()
11. sendEmail(subject, msg)

- setOpc(opc)

  Usamos esta funcion para verificar que la opcion sea valida (debe ser 1 o 2).

- getOpc()

  Si la opcion es valida, la regresamos.

- setEmailAccount(email_account)

  El usuario agrega su correo, pero como solo se aceptan 2 dominios “gmail” y “outlook/hotmail”, asi que tenemos que asegurarnos que el usuario haya agregado un correo valido. Si lo es lo almacenamos en una variable.

- getEmailAccount()

  Regresamos el correo.

- setPassword(password)

  El dominio correspondiente especifica que se usen ciertos caracteres en la contraseña, por lo que validamos dicha contraseña.

- getPassword()

  Si la contraseña es valida, la regresamos.

- setTo()

  Verificamos el correo al que se le enviara el mensaje.

- getTo()

  Regresamos el correo.

- setPicture(image, directory)

  En este modulo, tomamos el nombre de la imagen y el directorio en donde este almacenada, y guardamos esta informacion en una lista.

- getPicture()

  Regresamos la lista.

- sendEmail(subject, msg)

  El usuario agrega el asunto y el mensaje aqui, y mandamos el correo.

Hunter

1. setAPIKey(apikey)
2. getAPIKey()
3. setDomain()
4. getDomain()
5. search()
6. showInfo(results)
7. saveInfo(results)

- setAPIKey(apikey)

  Verificamos si la llave API es valida.

- getAPIKey()

  Regresamos la llave API.

- setDomain()

  El dominio tiene ciertos caracteres algo especificos, por lo que hay que verificarlo.

- getDomain()

  Regresamos el dominio.

- search()

  Buscamos la informacion del dominio.

- showInfo(results)

  La informacion que encontramos, la imprimimos en pantalla.

- saveInfo(results)

  Tomamos la informacion que encontramos y la guardamos.

ScanPort

1. validarIP
2. validaPuertos
3. scanPort

- validarIP

  Obtenemos la IP y verificamos que sea valida.

- validaPuertos

  Tomamos el rango de puertos que haya establecido el usuario y validamos que estos esten dentro del rango permitido.

- scanPort

  Usamos esta funcion para el escaneo de puertos, mostramos los resultados en pantalla y a su vez generamos un archivo .csv con los resultados obtenidos.

DNS(Script en PowerShell)

- Este script solo obtiene el DNS del cache del usuario.