Universidad Diego Portales

ESCUELA DE INFORMÁTICA & TELECOMUNICACIONES

Tarea5: Correos Electrónicos

Autor:
Marcos Fantóval Castro
marcos.fantoval@mail.udp.cl
19.962.344-3

Profesor: Nicolás Boettcher Ayudante: Francisco Lara

29 de Junio de 2021

Índice

1. Primer código

El primer código permite extraer la meta data de un correo electrónico o mejor dicho permite extraer la meta data solicitada dentro de todos los correos que sean recibidos desde un mismo emisor. El código es repetitivo osea se realiza la acción de extraer los datos de un correo y luego se replica ese código para los otros 4 correos por esto se mostrara el ejemplo del primer correo.

El código inicia con las librerías a utilizar:

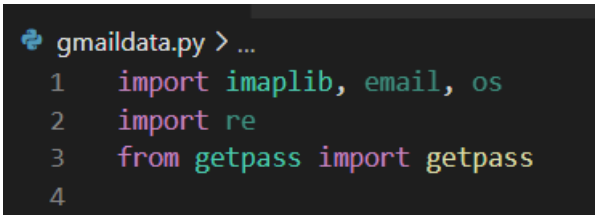

Figura 1: Librerías utilizadas

Luego se crea el host y se genera la conexión con el con el correo electrónico en este caso es mi propio gmail. Como medida de seguridad cada ves que se usa el código hay que ingresar el correo y la contraseña por terminal.

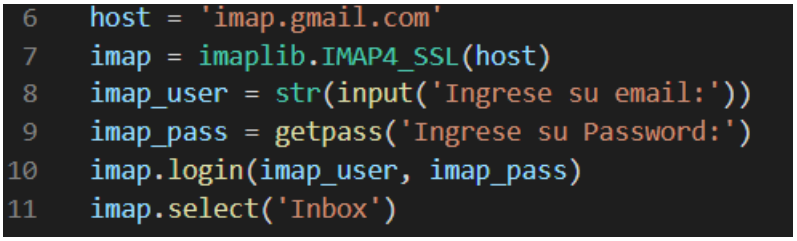

Figura 2: Conexión con el correo

Luego se guardan todos los correos recibidos en el inbox correspondiente a ximena.geoffroy@udp.cl en la variable data y también se crea el txt que contendra la meta data junto con el inicio del for que extraerá la data.

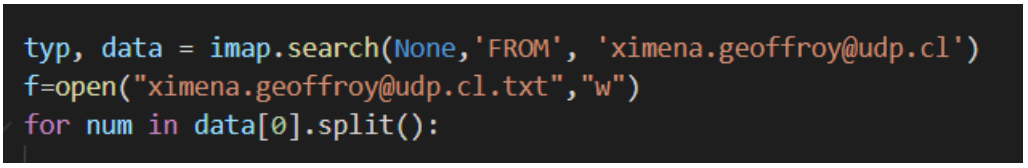

Figura 3: Extracción de los correos y creación del txt

Posteriormente se guardan los Message-Id se les extrae lo que no nos sirve se imprime por consola y se escribe en el txt.

```
typ, data = imap.fetch(num, '(BODY[HEADER.FIELDS (MESSAGE-ID)])')
datito= data[0][1].decode()
datito=datito.replace("Message-ID:", "")
datito=datito.replace(">", "")
datito=datito.replace("<", "")
datito=datito.replace("Message-Id:", "")
datito=datito.strip()
print(datito)
f.write(datito+"\n")
```

Figura 4: Message-Id

En seguida extrae los From los imprime por consola y los escribe en el txt. Hace lo mismo para el Date.

```
typ, data = imap.fetch(num, '(BODY[HEADER.FIELDS (From)])')
form=data[0][1].decode()
print(form)
f.write(form+"")

typ, data = imap.fetch(num, '(BODY[HEADER.FIELDS (Date)])')
fe=data[0][1].decode()
print(fe)
f.write(fe+"")
```

Figura 5: From y Date

Finalmente guarda los Received en un arreglo para poder extraer el primero y el penúltimo de que solo tenga 2 se muestran por consola los dos, después de mostrar los Received correspondientes los escribe en el txt, a la par se colocoan condicion en caso de que el correo solo tenga dos received.

```
typ, data = imap.fetch(num, '(BODY[HEADER.FIELDS (Received)])')
te=data[0][1].decode()
arr = None
arr = te.split("Received:")
largo = len(arr)
segundo =arr[largo-2]
primero = arr[largo-1]
print(primero)
f.write(primero+"")
penultimo = arr[2]

if primero == penultimo:
    print("El correo tiene 2 received.")
    print(segundo)
    f.write(primero+"")
else:
    print(penultimo)
    f.write(penultimo+"")
```

Figura 6: Received

El resultado por consola es el siguiente:

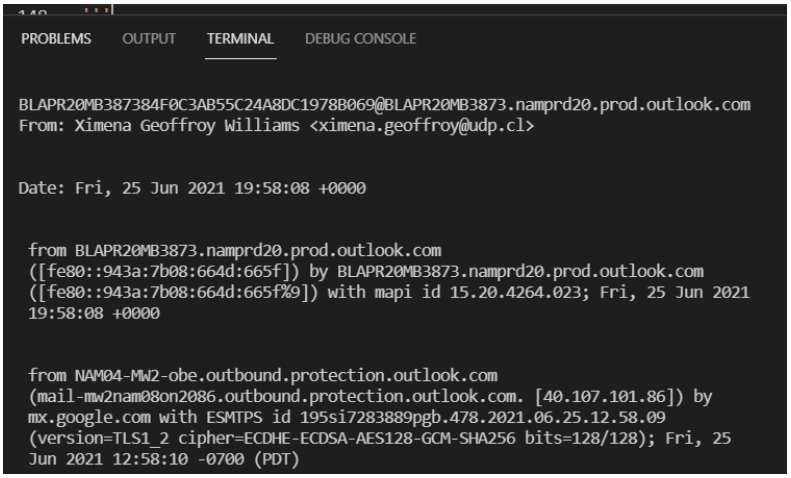

Figura 7: Received

2. Segundo código

Las regex (en inglés, regular expressions) son las unidades de descripción de los lenguajes regulares, que se incluyen en los denominados lenguajes formales.Es por esto que las expresiones regulares, claramente definidas, se utilizan principalmente en el ámbito del desarrollo de software.

La función principal de este código es que en base a la excreción regular de los correos electrónicos legítimos se pueda detectar cuando se esta tratando de suplantar la identidad en un correo supuestamente legitimo.

A continuación se muestra un correo legitimo elegido:

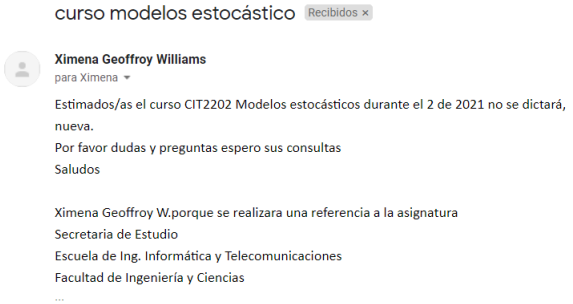

Figura 8: Correo legitimo

Luego se envió un correo suplantando el de Ximena con el fin de comprobar si se podía detectar el correo falso entre la lista de correos reales.

A continuación se muestra un correo falso:

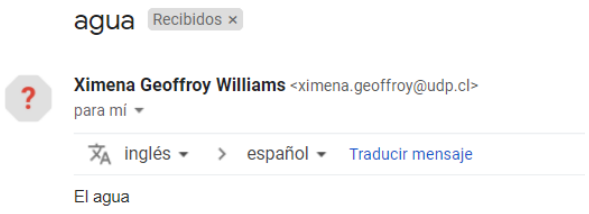

Figura 9: Correo falso

Para el paso de la comprobación del regex es que se utiliza la pagina web regex101 la cual ayuda a identificar un regex con el cual comparar la expresión de la cadena que se va a inutilizar para poder detectar el phishing y se guarda en un txt.

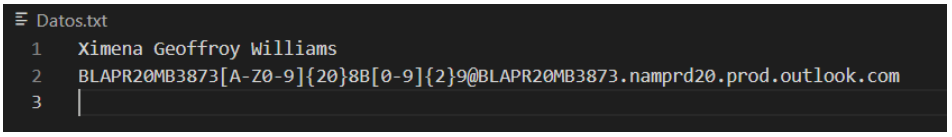

Figura 10: Correo falso

El código parte con las librería a utilizar:

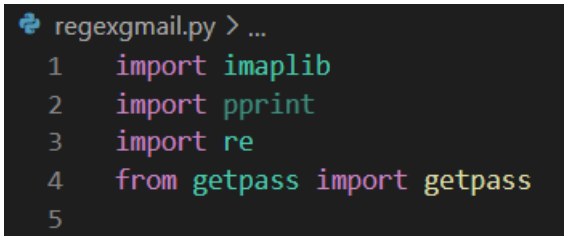

Figura 11: Librerías

Se extrae el remitente y el regex a comparar del archivo Datos.txt

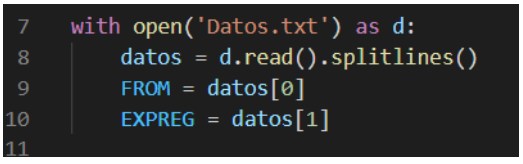

Figura 12: Datos requeridos

Al igual que el primer código se crea el host, se genera la conexión y se piden las credenciales por consola:

```
12  imap_host = 'imap.gmail.com'
13  imap_user = str(input('Ingrese su email:'))
14  imap_pass = getpass('Ingrese su Password:')
15  imap = imaplib.IMAP4_SSL(imap_host)
16  imap.login(imap_user, imap_pass)
17  imap.select('Inbox')
18  tmp, data = imap.search(None,'(FROM "'+FROM+'")')
19
```

Figura 13: Conexión con gmail

Para finalizar con el segundo código se crea un for que va a recorrer la bandeja de inbox buscando los correos. Después se guarda la excreción regula a comparar luego se guarda el regex a comparar que se encuentra en Datos.txt y se guarda en obtenido de los correos que se están analizando finalmente se colocan las condiciones que determinara si es el regex correcto o no.

```
20  for itera in data[0].split():
21          tmp, data = imap.fetch(itera, '(BODY[HEADER.FIELDS (FROM MESSAGE-ID DATE)])')
22          print('Numero de mensaje en la bandeja de entrada:{0}'.format(itera.decode("utf-8")))
23          aux = str(data[0][1]).replace('\r\n', '\n')
24          mail = re.compile(r"""""+EXPREG+""""", re.X)
25          mails =  data[0][1].decode('utf-8')
26          if re.search(mail, mails):
27              x = mail.findall(mails)
28              print(str(x)+' Se confirma que es un emisor real\n')
29          else:
30              print('ALERTA, POSIBLE PHISHING!.\n')
31  imap.close()
```

Figura 14: Respuesta de consola

Al ejecutar el codigo da por consola el siguiente resultado:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
['BLAPR20MB38730C93BEAE3EB7C9910A8B8B389@BLAPR20MB3873.namprd20.prod.outlook.com'] Se confirma que es un emisor real

Numero de mensaje en la bandeja de entrada:1195
Los datos del correo son los siguientes:
b'From: Ximena Geoffroy Williams <ximena.geoffroy@udp.cl>\r\nDate: Tue, 15 Jun 2021 14:54:52 +0000\r\nMessage-ID: <BLAPR20MB38730128CA023F838
60706528B309@BLAPR20MB3873.namprd20.prod.outlook.com>\r\n\r\n'
['BLAPR20MB38730128CA023F838607065 28B309@BLAPR20MB3873.namprd20.prod.outlook.com'] Se confirma que es un emisor real

Numero de mensaje en la bandeja de entrada:1242
Los datos del correo son los siguientes:
b'From: Ximena Geoffroy Williams <ximena.geoffroy@udp.cl>\r\nDate: Tue, 22 Jun 2021 22:59:42 +0000\r\nMessage-ID: <BLAPR20MB38732143B1C37578D
32CBCB18B099@BLAPR20MB3873.namprd20.prod.outlook.com>\r\n\r\n'
['BLAPR20MB38732143B1C37578D32CBCB18B099@BLAPR20MB3873.namprd20.prod.outlook.com'] Se confirma que es un emisor real

Numero de mensaje en la bandeja de entrada:1261
Los datos del correo son los siguientes:
b'From: Ximena Geoffroy Williams <ximena.geoffroy@udp.cl>\r\nDate: Fri, 25 Jun 2021 19:58:08 +0000\r\nMessage-ID: <BLAPR20MB387384F0C3AB55C24
A8DC1978B069@BLAPR20MB3873.namprd20.prod.outlook.com>\r\n\r\n'
['BLAPR20MB387384F0C3AB55C24A8DC1978B069@BLAPR20MB3873.namprd20.prod.outlook.com'] Se confirma que es un emisor real

Numero de mensaje en la bandeja de entrada:1286
Los datos del correo son los siguientes:
b'From: "Ximena Geoffroy Williams" <ximena.geoffroy@udp.cl>\r\nMessage-Id: <20210630031931.E301124113@emkei.cz>\r\nDate: Wed, 30 Jun 2021 05:
19:31 +0200 (CEST)\r\n\r\n'
ALERTA, POSIBLE PHISHING!.


Marcos F@LAPTOP-6U6AS94O MINGW64 ~/Desktop/cripto/Tarea5
$
```

Figura 15: Respuesta de consola

3. Link del GitHub

https://github.com/MarcosFantoval2/CriptoT5