



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



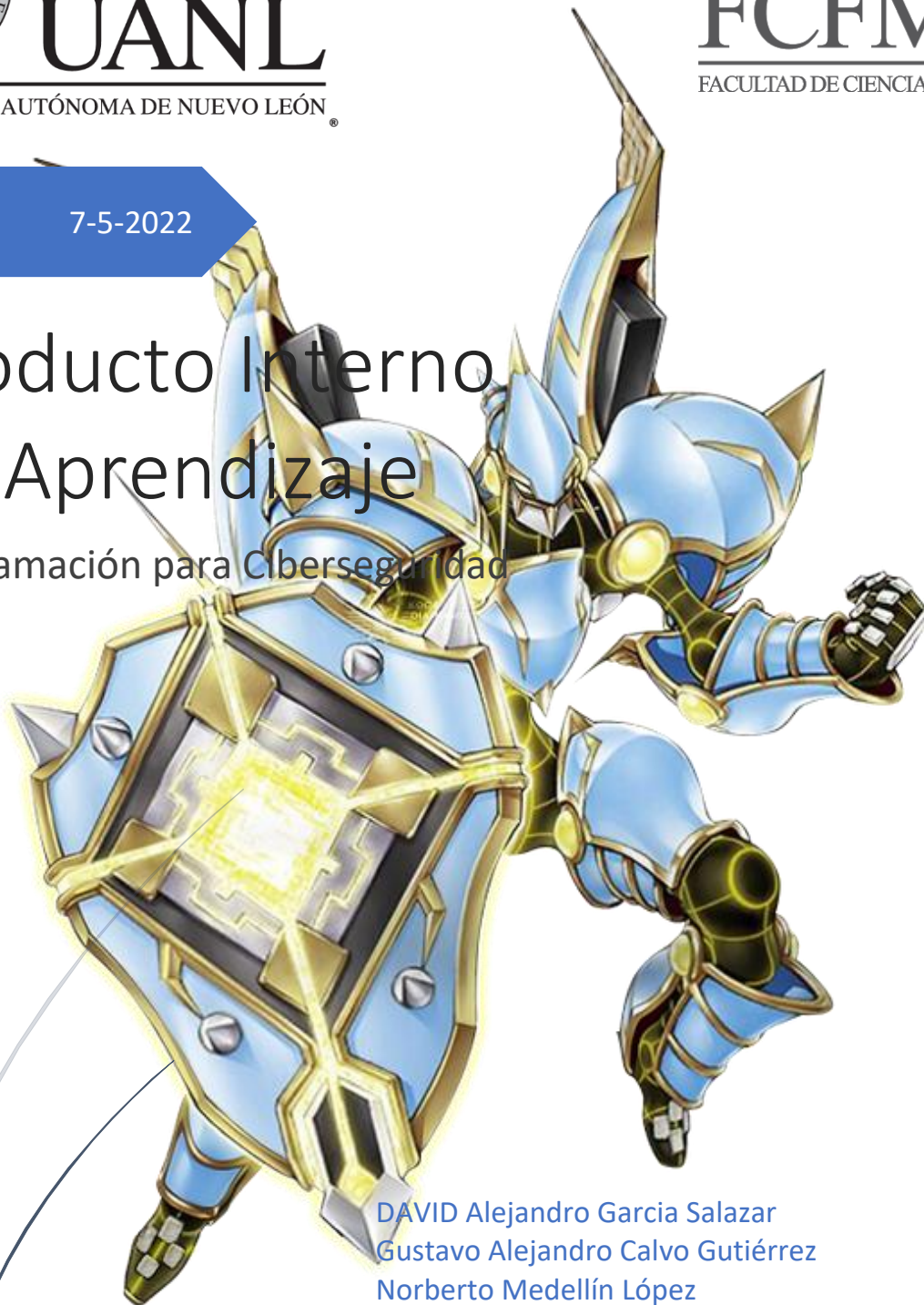
FCFM

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

7-5-2022

Producto Interno de Aprendizaje

Programación para Ciberseguridad



DAVID Alejandro Garcia Salazar
Gustavo Alejandro Calvo Gutiérrez
Norberto Medellín López
Maximiliano Velázquez Gonzales
Juan José Rivera Arroyo

Introducción

Este Proyecto es una recopilación de diversas herramientas de ciberseguridad, de las cuales podemos encontrar web scrapping, escaneo de puertos, extracción de metadatos, envío de correos y obtención de valores hash.

Este trabajo funciona mediante argumentos, aunque solo ciertos scripts requieren 'cierta' interacción del usuario.

Este proyecto fue elaborado y ejecutado con la versión de python 3.10.4

Guía de Instalación

Para la ejecución correcta de este trabajo se es necesario el tener instalado ciertos módulos y programas, en caso de que no se tenga instalado algo necesario para la ejecución, el script lo descargara e instalara automáticamente. O bien puede optar por instalar cada módulo manualmente como los siguientes ejemplos:

`pip install nmap Python-nmap`

`Pip install requests`

`Pip install beautifulsoup4`

`Pip install PyPDF2`

****NOTA****

Para el caso de nmap se deberá ir a la página oficial de nmap y se descargara e instalara el ncap 1.60 primero, después se instalará el nmap 1.7, ojo aquí que el instalador le pedirá sobrescribir el ncap instalado, por lo que le diremos que no, y finalmente se instalara en el cmd python-nmap

Ejecución de la Herramienta

Asegúrese de utilizar correctamente los argumentos, si tiene dudas puede hacer uso del argumento -h que le despliega los demás argumentos así como su utilidad.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19044.1645]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Admin>cd Downloads
C:\Users\Admin\Downloads>cd PDF_final111
C:\Users\Admin\Downloads\PDF_final111>cd PDF
C:\Users\Admin\Downloads\PDF_final111\PDF>python main_prueba.py.py -h
usage: main_prueba.py.py [-h] [-opc OPC] [-tipo_archivo TIPO_ARCHIVO] [-ruta_metadatos RUTA_METADATOS]
                        [-remite REMITENTE] [-ip IP] [-i I] [-f F] [-url URL]

options:
  -h, --help            show this help message and exit
  -opc OPC              -opc (1=Extraer y enviar informacion, 2=nmap, 3=web scrapping)
  -tipo_archivo TIPO_ARCHIVO
                        -tipo_archivo (1=Pdf)
  -ruta_metadatos RUTA_METADATOS
                        -ruta_metadatos "pon la ruta completa en donde quieres sacar los metadatos del pdf"
  -remite REMITENTE    -remite "pon el correo al que quieres enviar"
  -ip IP               -ip "ip a escanear"
  -i I                 -i "inicio de los puertos a escanear"
  -f F                 -f "Final de los puertos a escanear"
  -url URL             -url "Escribe la url para hacer el web scrapping"

C:\Users\Admin\Downloads\PDF_final111\PDF>
```

Herramienta de webscraping.

Python "nombre del script" -opc 1 -url "url"

Herramienta PortScanner.

Python "nombre del script" -opc 2 -ip "ip" -i "puerto inicial" -f "puerto final"

Herramienta metadatos

Python "nombre del script" -opc 1 -tipo_archivo 1 -ruta_metadatos "ruta completa de donde quieres sacar metadatos"

Módulos Usados

Beautiful Soup

Es una biblioteca de Python para extraer datos de archivos HTML y XML. Funciona con su analizador favorito para proporcionar formas idiomáticas de navegar, buscar y modificar el árbol de análisis. Comúnmente ahorra a los programadores horas o días de trabajo.

Nmap

Una de las primeras consideraciones al contemplar un escaneo de puertos es decidir qué técnicas utilizar. Nmap ofrece alrededor de una docena de estos métodos. La cobertura completa viene en el siguiente capítulo. Solo se puede utilizar un método de exploración a la vez, excepto que la exploración UDP (-sU) se puede combinar con cualquiera de los tipos de exploración TCP.

Subprocess

El módulo de subprocess le permite generar nuevos procesos, conectarse a sus tuberías de entrada / salida / error y obtener sus códigos de retorno. Este módulo pretende reemplazar varios módulos y funciones más antiguos.

Ssl

Este módulo proporciona acceso a las instalaciones de cifrado y autenticación del mismo nivel de Transport Layer Security (a menudo conocida como "Secure Sockets Layer") para sockets de red, tanto del lado del cliente como del lado del servidor. Este módulo utiliza la biblioteca OpenSSL. Está disponible en todos los sistemas Unix modernos, Windows, macOS y probablemente plataformas adicionales, siempre y cuando OpenSSL esté instalado en esa plataforma.

Smtplib

El módulo smtplib define un objeto de sesión de cliente SMTP que se puede utilizar para enviar correo a cualquier equipo de Internet con un demonio de escucha SMTP o ESMTP. Para obtener detalles sobre el funcionamiento de SMTP y ESMTP, consulte RFC 821 (Protocolo simple de transferencia de correo) y RFC 1869 (Extensiones de servicio SMTP).

PyPDF2

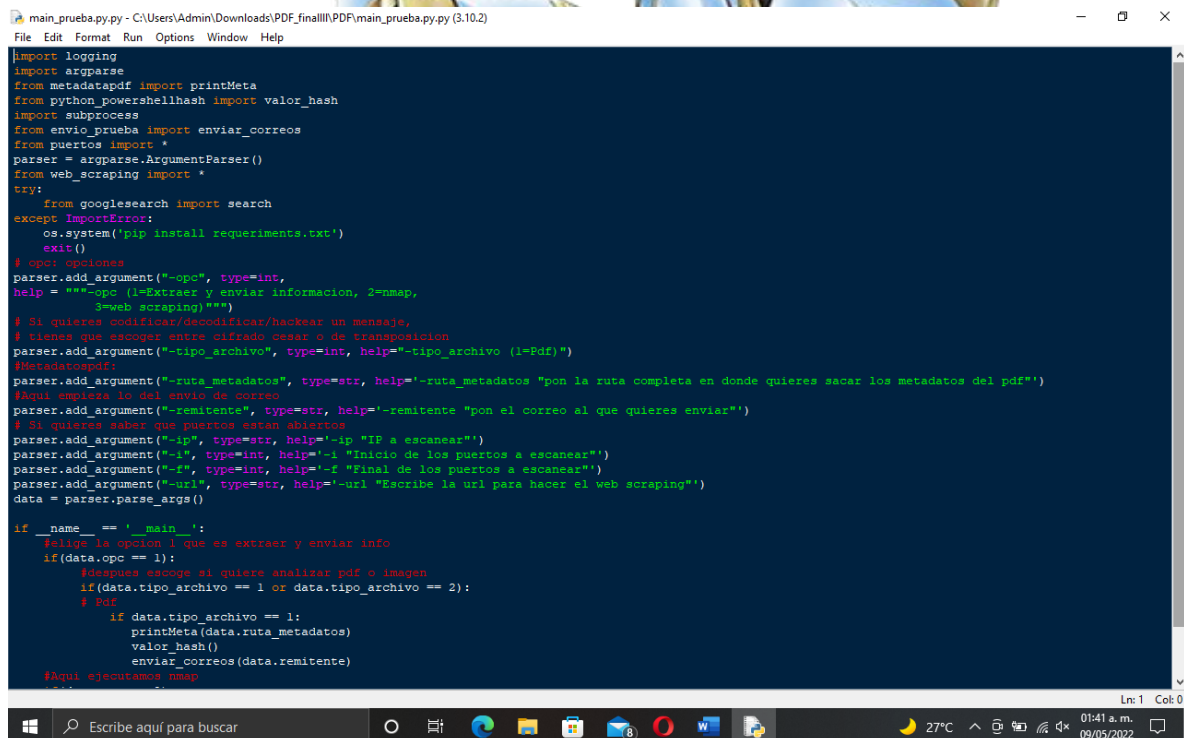
PyPDF2 es una biblioteca PDF de Python puro capaz de dividir, fusionar, recortar y transformar las páginas de los archivos PDF. También puede agregar datos personalizados, opciones de visualización y contraseñas a los archivos PDF. Puede recuperar texto y metadatos de archivos PDF, así como fusionar archivos completos.

Socket

Este módulo proporciona acceso a la interfaz BSD socket. Está disponible en todos los sistemas Unix modernos, Windows, MacOS y probablemente plataformas adicionales. Algunos comportamientos pueden depender de la plataforma, ya que las llamadas se realizan a las API de socket del sistema operativo.

Codigos y breve explicación

1. Main: este script lo que hace es establecer los argumentos así como trabajar de archivo enlazador con los demás script, además de que instala los módulos faltantes del archivo txt.



```
main_prueba.py.py - C:\Users\Admin\Downloads\PDF_final\PDF\main_prueba.py (3.10.2)
File Edit Format Run Options Window Help

import logging
import argparse
from metadatepdf import printMeta
from python_powershellhash import valor_hash
import subprocess
from envio_prueba import enviar_correos
from puertos import *
parser = argparse.ArgumentParser()
from web_scraping import *
try:
    from googlesearch import search
except ImportError:
    os.system('pip install requirements.txt')
    exit()
# opc: opciones
parser.add_argument("-opc", type=int,
help = """-opc 1=Extraer y enviar informacion, 2=nmap,
3=web scraping""")
# Si quieres codificar/decodificar/hackear un mensaje,
# tienes que escoger entre cifrado cesar o de transposicion
parser.add_argument("-tipo_archivo", type=int, help="-tipo_archivo (1=Pdf)")
#Metadatepdf:
parser.add_argument("-ruta_metadatos", type=str, help="-ruta_metadatos "pon la ruta completa en donde quieres sacar los metadatos del pdf")
#Aqui empieza lo del envio de correo
parser.add_argument("-remite", type=str, help="-remite "pon el correo al que quieres enviar")
# Si quieres saber que puertos estan abiertos
parser.add_argument("-ip", type=str, help="-ip "IP a escanear")
parser.add_argument("-i", type=int, help="-i "Inicio de los puertos a escanear")
parser.add_argument("-f", type=int, help="-f "Final de los puertos a escanear")
parser.add_argument("-url", type=str, help="-url "Escribe la url para hacer el web scraping")
data = parser.parse_args()

if __name__ == '__main__':
    #elige la opcion 1 que es extraer y enviar info
    if(data.opc == 1):
        #despues escoge si quiere analizar pdf o imagen
        if(data.tipo_archivo == 1 or data.tipo_archivo == 2):
            # Pdf
            if data.tipo_archivo == 1:
                printMeta(data.ruta_metadatos)
                valor_hash()
                enviar_correos(data.remite)
            #Aqui ejecutamos nmap
```

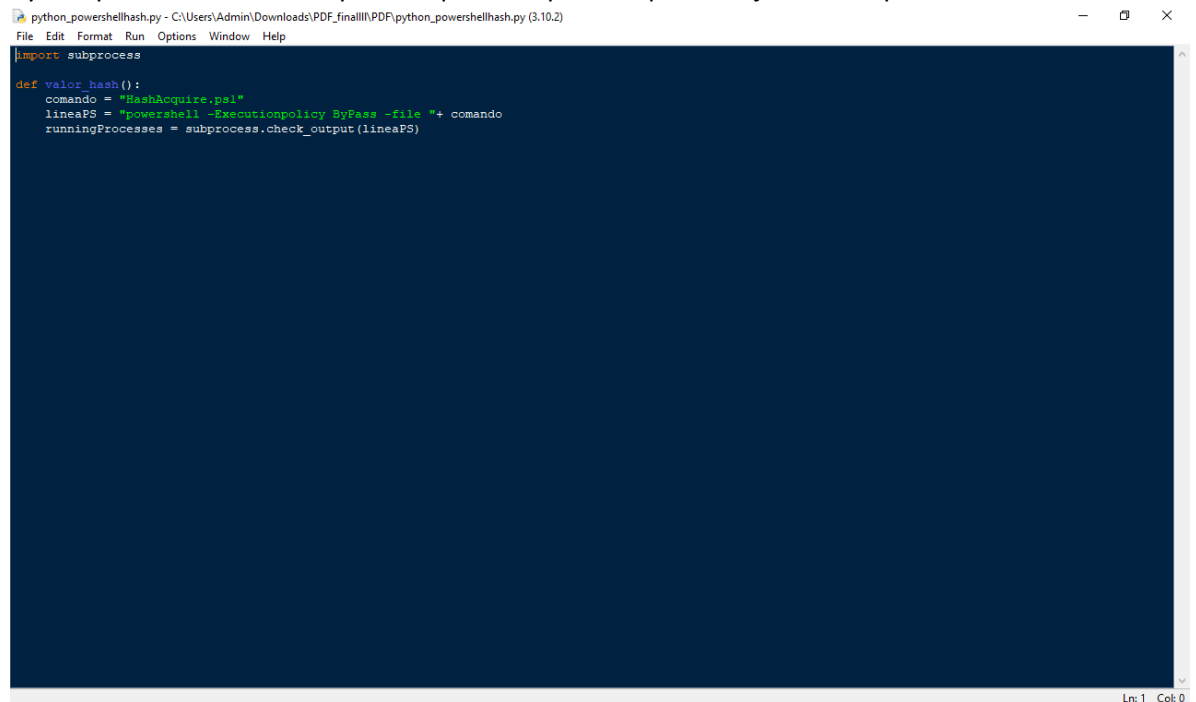
```
requirements: Bloc de notas
Archivo Edición Formato Ver Ayuda
beautifulsoup4==4.11.1
googlesearch_python==1.0.1
nmap==0.0.1
PyPDF2==1.27.12
python_nmap==0.7.1
requests==2.25.1
```

2. Metadatapdf: Este script lo que hace es que apartir de la ruta especificada obtendrá los metadatos de los archivos que estén en la ruta y los guardara en un archivo txt.

```
metadatapdf.py - C:\Users\Admin\Downloads\PDF_final\PDF\metadatapdf.py (3.10.2)
File Edit Format Run Options Window Help
from PyPDF2 import PdfFileReader, PdfFileWriter
import os
import pickle

def printMeta(ruta):
    os.chdir(ruta)
    for root, dirs, files in os.walk(".", topdown=False):
        for name in files:
            ext=name.lower().rsplit(".",1) [-1]
            if ext in ["pdf"]:
                print("\n%s Metadeta for file: %s " %(ruta+os.path.sep+name))
                pdfFile = PdfFileReader(open(ruta+os.path.sep+name, "rb"))
                docInfo = pdfFile.getDocumentInfo()
                print("\nTipo: ", type(docInfo))
                for metaItem in docInfo:
                    Info = "[+] " + metaItem + ":" + docInfo[metaItem]
                    filel = open("Metadatos.txt", "a")
                    filel.write(Info + "\n")
                    filel.close
```

3. Pythonpowershell: Este script es el que hace posible que trabajemos con powershell

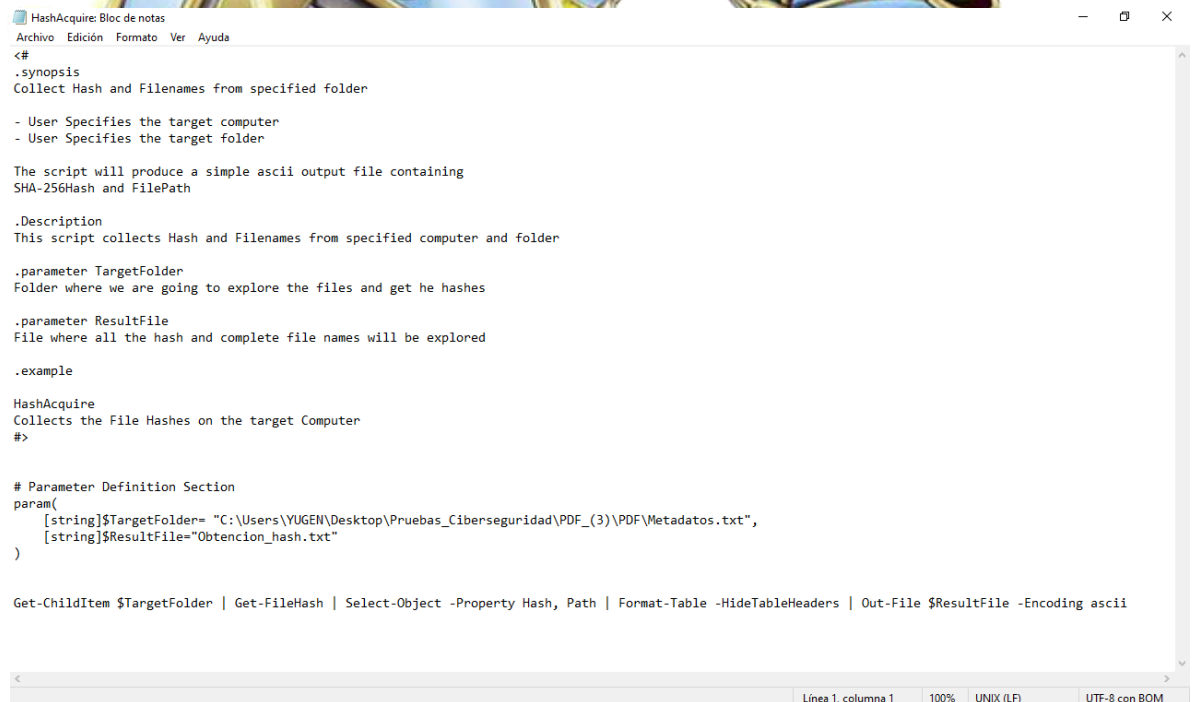


```
python_powershellhash.py - C:\Users\Admin\Downloads\PDF_finall\PDF\python_powershellhash.py (3.10.2)
File Edit Format Run Options Window Help

import subprocess

def valor_hash():
    comando = "HashAcquire.ps1"
    lineaPS = "powershell -Executionpolicy ByPass -file "+ comando
    runningProcesses = subprocess.check_output(lineaPS)
```

4. HashAquire: Este script es el encargado de obtener los valores hash. OJO poner en la variable \$TargetFolder la ruta completa de la carpeta que en donde se encuentra el Metadatos.txt



```
HashAcquire: Bloc de notas
Archivo Edición Formato Ver Ayuda

<#
.synopsis
Collect Hash and Filenames from specified folder

- User Specifies the target computer
- User Specifies the target folder

The script will produce a simple ascii output file containing
SHA-256Hash and FilePath

.Description
This script collects Hash and Filenames from specified computer and folder

.parameter TargetFolder
Folder where we are going to explore the files and get he hashes

.parameter ResultFile
File where all the hash and complete file names will be explored

.example
HashAcquire
Collects the File Hashes on the target Computer
#>

# Parameter Definition Section
param(
    [string]$TargetFolder= "C:\Users\YUGEN\Desktop\Pruebas_Ciberseguridad\PDF_(3)\PDF\Metadatos.txt",
    [string]$ResultFile="Obtencion_hash.txt"
)

Get-ChildItem $TargetFolder | Get-FileHash | Select-Object -Property Hash, Path | Format-Table -HideTableHeaders | Out-File $ResultFile -Encoding ascii
```

5. Envío_prueba: Este script es el que envía por correo la información extraída de los script anteriores

```
envio_prueba.py - C:\Users\Admin\Downloads\PDF_finall\PDF\envio_prueba.py (3.10.2)
File Edit Format Run Options Window Help

import smtplib, ssl
from email import encoders
from email.mime.base import MIMEBase
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import os

def enviar_correos(receiver_email):
    port = 587 # for starttls
    smtp_server = "smtp.office365.com"
    subject = "Obtención de metadatos y valores Hash de PDFs"
    body = "Pis"
    sender_email = "Prueba@deciberseguridad@outlook.com"
    password = "Prueba@contra9"

    message = MIMEMultipart()
    message["From"] = sender_email
    message["To"] = receiver_email
    message["Bcc"] = receiver_email
    message["Subject"] = subject
    message.attach(MIMEText(body, "plain"))

    filename = ["Obtencion_hash.txt", "Metadatos.txt"]

    for i in filename:

        with open(i, "rb") as attachment:
            # Add file as application/octet-stream
            # Email client can usually download this automatically as attachment
            part = MIMEBase("application", "octet-stream")
            part.set_payload(attachment.read())

            # Encode file in ASCII characters to send by email
            encoders.encode_base64(part)

            # Add header as key/value pair to attachment part
            part.add_header(
                "Content-Disposition",
                f"attachment; filename= {i}",
            )

        # Add attachment to message and convert message to string
        message.attach(part)
```

6. Nmap: Este script es el que se encarga del escaneo de puertos y el uso del socket

```
puertos.py - C:\Users\Admin\Downloads\PDF_finall\PDF\puertos.py (3.10.2)
File Edit Format Run Options Window Help

import nmap
import csv

#toma el rango de los puertos
#que serán escaneados

def nmap_funcion(target, begin, end):
    scanner = nmap.PortScanner()
    for i in range(begin, end+1):
        #Empieza el escaneo de puertos
        res = scanner.scan(target, str(i))
        for host in scanner.all_hosts():
            print('Host : %s (%s)' % (host, scanner[host].hostname()))
            print('State : %s' % scanner[host].state())
            for proto in scanner[host].all_protocols():
                print('Protocol : %s' % proto)

                lport = scanner[host][proto].keys()
                for port in lport:
                    print ('%s : %s\tstate : %s' % (port, scanner[host][proto][port]['state']))
                    if scanner[host][proto][port]['state'] == 'open':
                        port=open('puertos.csv', 'a')
                        port=scanner.csv()
                        for i in port:
                            port.write(i)
                        port.close()
```

7. Webscraping: Este script es el encargado de hacer el scrapping de la url especificada

```
web_scraping.py - C:\Users\Admin\Downloads\PDF_final\PDF\web_scraping.py (3.10.2)
File Edit Format Run Options Window Help
import urllib.request
def web(url):
    from bs4 import BeautifulSoup
    datos=urllib.request.urlopen(url).read().decode()
    soup = BeautifulSoup(datos,"html.parser")
    tags=soup('a')
    for tag in tags:
        print(tag.get('href'))
    tags=soup('link')
    for tag in tags:
        print(tag.get('rel'))
```

Tabla de Entregables

Insertar aquí todos los entregables y la información solicitada.
Se puede cambiar el tamaño de la tabla según se requiera.

Descripción de las tareas de Ciberseguridad que realiza el proyecto	1.- Obtención de metadatos de PDFS 2.- Obtención del valor hash. 3.-Envío de correos 4.-Escaneo de puertos 5.-Web Scraping
Liga de GitHub del Proyecto	https://github.com/Maximilianovizgzz/PA
Lista de HASH y archivos	<<Obtencion_hash_documentoss.txt>>
Video Permisos únicamente para el grupo de clase	El docente colocará aquí la grabación de 15 minutos de la demo del proyecto, los integrantes del equipo NO deben poner nada aquí
Video de trabajo en equipo Permisos únicamente para el grupo de clase	Ponernos de acuerdo. Llamada con PC EQ1 062 E2022-20220412 174854-Grabación de la reunión.mp4 Explicación Pía Llamada con PC EQ1 062 E2022-20220509 025939-Grabación de la reunión.mp4