



ESCUELA SUPERIOR DE CÓMPUTO



Asignatura: Computer Security

Actividad: Tarea 2 – Metadatos

Alumno: De los Santos Montiel Emmanuel

Profesor: Aldama Coahuila Mario Alberto

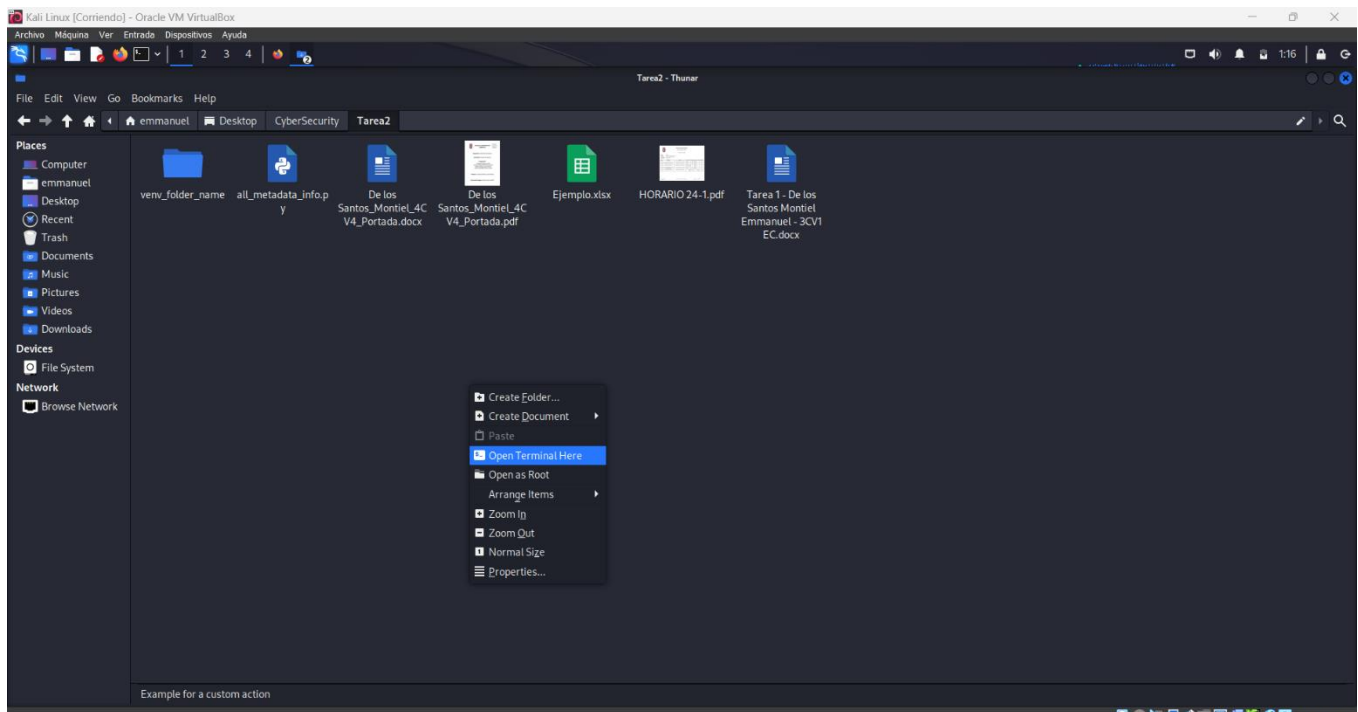
Fecha de Entrega: 22 de marzo de 2024

Con lo aprendido durante las sesiones de la asignatura con el profesor, se nos pidió elaborar un “script”, en el lenguaje de programación de Python, en el cual nosotros pudiéramos leer y sacar los metadatos de 3 tipos de archivos diferentes, en este caso se trabajó con:

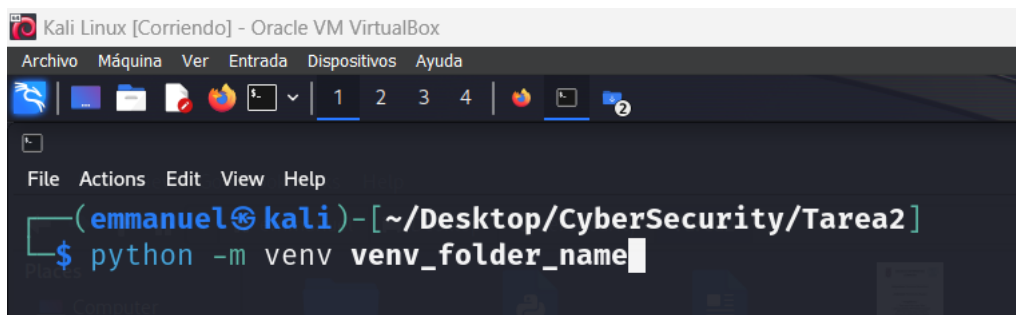
- Archivos docx
- Archivos xlsx
- Archivos PDF

A continuación, se explica a detalle lo que se realizó para que se pudiera extraer dicha información correctamente.

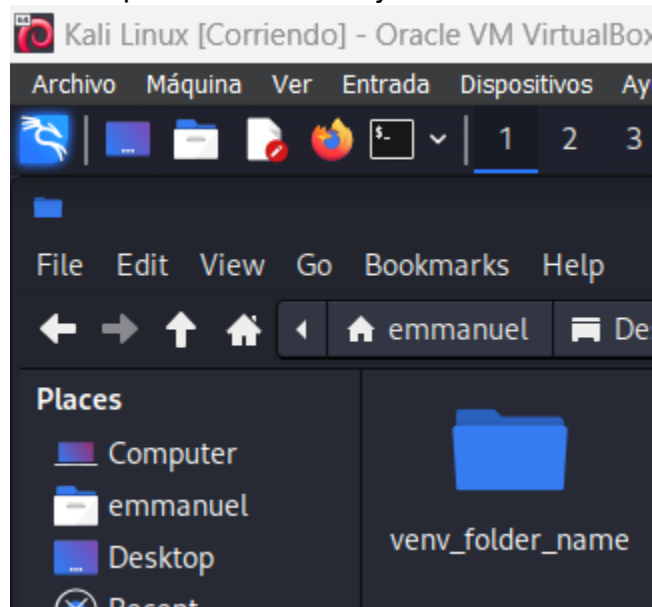
1. Teniendo en cuenta que, al trabajar con la máquina virtual de Kali, ya contamos con la instalación de Python por default, vamos a abrir una terminal en la carpeta en donde vayamos a tener nuestro ambiente virtual.



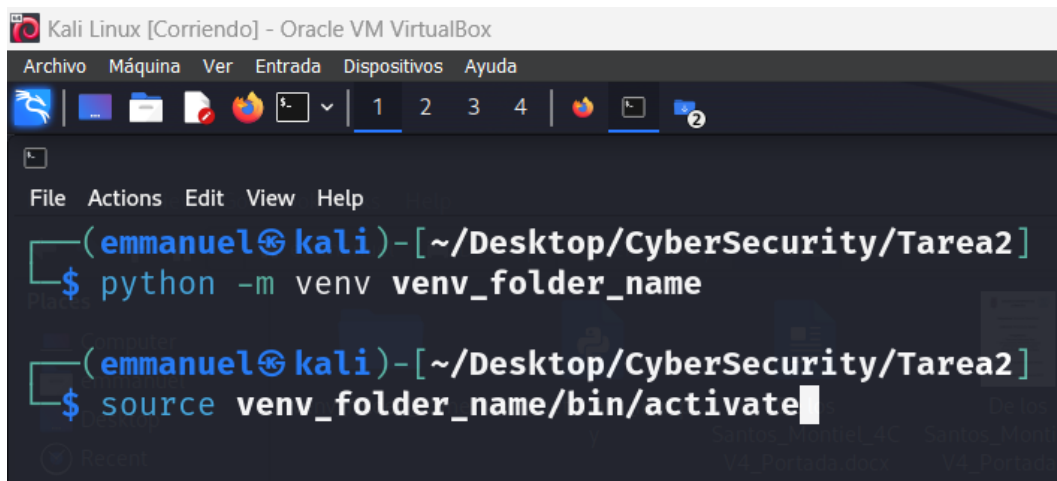
2. Después de que ya tenemos abierta la terminal, ponemos el siguiente comando y lo ejecutamos: **python -m venv venv_folder_name**



3. Con este comando habremos creado una carpeta y con ello podremos activar el ambiente virtual en que vamos a trabajar.



4. Para activar el ambiente virtual, escribimos el siguiente comando: **source venv_folder_name/bin/activate**



5. Ahora con el comando “pip”, vamos a instalar algunas bibliotecas que vamos a utilizar para poder leer y extraer los metadatos de los documentos que vamos a analizar. En este caso ponemos el siguiente comando: **pip install python-docx openpyxl PyPDF2**

```
Kali Linux [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
(venv_folder_name)emmanuel@kali: ~/Desktop/CyberSecurity/Tarea2

(venv_folder_name)-(emmanuel@kali)-[~/Desktop/CyberSecurity/Tarea2]
$ pip install python-docx openpyxl PyPDF2
Collecting python-docx
  Using cached python_docx-1.1.0-py3-none-any.whl.metadata (2.0 kB)
Collecting openpyxl
  Using cached openpyxl-3.1.2-py2.py3-none-any.whl.metadata (2.5 kB)
Collecting PyPDF2
  Using cached pypdf2-3.0.1-py3-none-any.whl.metadata (6.8 kB)
Requirement already satisfied: lxml<=3.1.0 in ./venv_folder_name/lib/python3.11/site-packages (from python-docx) (5.1.0)
Requirement already satisfied: typing-extensions in ./venv_folder_name/lib/python3.11/site-packages (from python-docx) (4.10.0)
Requirement already satisfied: et-xmlfile in ./venv_folder_name/lib/python3.11/site-packages (from openpyxl) (1.1.0)
Using cached python_docx-1.1.0-py3-none-any.whl (239 kB)
Using cached openpyxl-3.1.2-py2.py3-none-any.whl (249 kB)
Using cached pypdf2-3.0.1-py3-none-any.whl (232 kB)
Installing collected packages: python-docx, PyPDF2, openpyxl
Successfully installed PyPDF2-3.0.1 openpyxl-3.1.2 python-docx-1.1.0

(venv_folder_name)-(emmanuel@kali)-[~/Desktop/CyberSecurity/Tarea2]
$
```

6. Verificamos que se hayan instalado las bibliotecas con el comando: **pip list**

```
(venv_folder_name)-(emmanuel@kali)-[~/Desktop/CyberSecurity/Tarea2]
$ pip list
Package            Version
-----
et-xmlfile          1.1.0
lxml                 5.1.0
openpyxl            3.1.2
pip                 23.3
PyPDF2              3.0.1
python-docx         1.1.0
setuptools          68.1.2
typing_extensions   4.10.0
```

7. Ahora, ya que tenemos las bibliotecas instaladas, creamos nuestro script

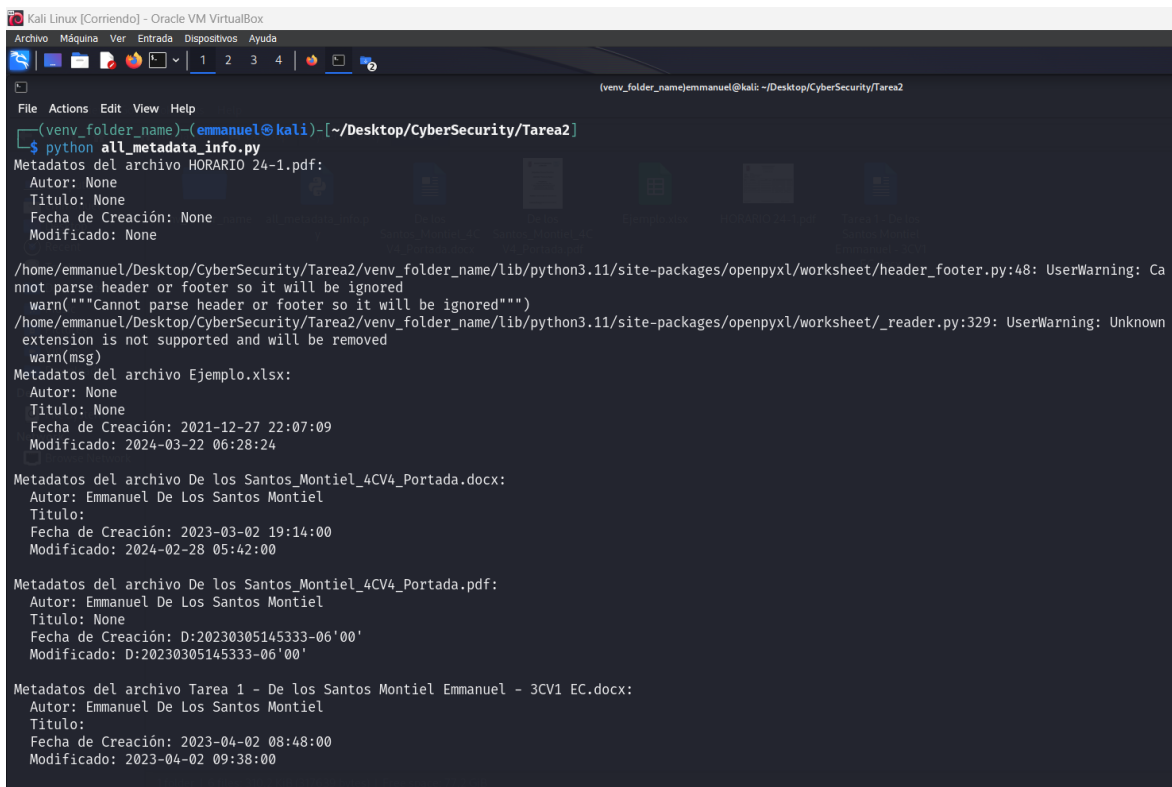
```
1 import os
2 import re
3 from docx import Document
4 from openpyxl import load_workbook
5 from PyPDF2 import PdfReader
6
7 def extract_metadata_from_pdf(filename):
8     pdf_file = PdfReader(filename)
9     metadata = {
10         "Autor": pdf_file.metadata.author,
11         "Titulo": pdf_file.metadata.title,
12         "Fecha de Creación": pdf_file.metadata.get('/CreationDate', None),
13         "Modificado": pdf_file.metadata.get('/ModDate', None)
14     }
15     return metadata
16
17 def extract_metadata_from_docx(filename):
18     document = Document(filename)
19     core_properties = document.core_properties
20     metadata = {
21         "Autor": core_properties.author,
22         "Titulo": core_properties.title,
23         "Fecha de Creación": core_properties.created,
24         "Modificado": core_properties.modified
25     }
26     return metadata
27
28 def extract_metadata_from_xlsx(filename):
29     wb = load_workbook(filename)
30     author = None
31     #verificamos la información para revisar si existe algún autor
32     if wb.sheetnames:
33         sheet = wb[wb.sheetnames[0]]
```

```

34         if sheet.dimensions:
35             author = sheet.cell(row=1, column=1).value
36     metadata = {
37         "Autor": author,
38         "Titulo": wb.properties.title,
39         "Fecha de Creación": wb.properties.created,
40         "Modificado": wb.properties.modified
41     }
42     return metadata
43
44 def process_files(directory_path):
45     for file_name in os.listdir(directory_path):
46         file_path = os.path.join(directory_path, file_name)
47         if os.path.isfile(file_path):
48             file_extension = re.findall(r"\.(pdf|docx|xlsx)$", file_name)
49             if file_extension:
50                 file_extension = file_extension[0]
51                 if file_extension == "pdf":
52                     metadata = extract_metadata_from_pdf(file_path)
53                 elif file_extension == "docx":
54                     metadata = extract_metadata_from_docx(file_path)
55                 elif file_extension == "xlsx":
56                     metadata = extract_metadata_from_xlsx(file_path)
57                 print(f"Metadatos del archivo {file_name}:")
58                 for k, v in metadata.items():
59                     print(f"    {k}: {v}")
60                 print()
61
62 # Ruta al directorio
63 directory_path = "/home/emmanuel/Desktop/CyberSecurity/Tarea2"
64
65 # Procesar los archivos en el directorio
66 process_files(directory_path)

```

8. Lo mandamos a llamar con el siguiente comando: **python “nombre_archivo.py”**, en este caso sería: **python all_metadata_info.py**



```
Kali Linux [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
(venv_folder_name)emmanuel@kali: ~/Desktop/CyberSecurity/Tarea2
File Actions Edit View Help
(venv_folder_name)~(emmanuel@kali)-[~/Desktop/CyberSecurity/Tarea2]
$ python all_metadata_info.py
Metadatos del archivo HORARIO 24-1.pdf:
Autor: None
Titulo: None
Fecha de Creación: None
Modificado: None

/home/emmanuel/Desktop/CyberSecurity/Tarea2/venv_folder_name/lib/python3.11/site-packages/openpyxl/worksheet/header_footer.py:48: UserWarning: Cannot parse header or footer so it will be ignored
warn("Cannot parse header or footer so it will be ignored")
/home/emmanuel/Desktop/CyberSecurity/Tarea2/venv_folder_name/lib/python3.11/site-packages/openpyxl/worksheet/_reader.py:329: UserWarning: Unknown extension is not supported and will be removed
warn(msg)
Metadatos del archivo Ejemplo.xlsx:
Autor: None
Titulo: None
Fecha de Creación: 2021-12-27 22:07:09
Modificado: 2024-03-22 06:28:24

Metadatos del archivo De los Santos Montiel_4CV4_Portada.docx:
Autor: Emmanuel De Los Santos Montiel
Titulo:
Fecha de Creación: 2023-03-02 19:14:00
Modificado: 2024-02-28 05:42:00

Metadatos del archivo De los Santos Montiel_4CV4_Portada.pdf:
Autor: Emmanuel De Los Santos Montiel
Titulo: None
Fecha de Creación: D:20230305145333-06'00'
Modificado: D:20230305145333-06'00'

Metadatos del archivo Tarea 1 - De los Santos Montiel Emmanuel - 3CV1 EC.docx:
Autor: Emmanuel De Los Santos Montiel
Titulo:
Fecha de Creación: 2023-04-02 08:48:00
Modificado: 2023-04-02 09:38:00
```

Aquí podemos ver el resultado de los metadatos que pudimos extraer, y aunque nos aparecen unos “warnings”, no nos genera más problemas, ya que el script sigue leyendo los demás documentos que están en el directorio que especificamos y va extrayendo todos los datos.