Ejercicio de reglas Yara

Para este ejercicio se debe crear un archivo Python que descargue y compile de diferentes repositorios varias reglas Yara para poder después realizar comprobaciones y lograr identificar diferentes malware.

Para ello, los pasos a seguir son:

1º Buscar los repositorios con reglas Yara en GitHub

2º Programamos el fichero en el lenguaje elegido, en nuestro caso en Python

3º Una vez creado, ejecutamos el comando “python3 YaraRules.py”

4º Una vez finalice la compilación usamos la función de yara para que trate de identificar el malware que deseamos usando las reglas compiladas usando el comando “yara -C compiled_rules <fichero malware>”

Funcionamiento del archivo Python

El archivo Python se divide en 3 partes

1ª -> Importamos las funciones necesarias y creamos las nuestras propias funciones para descargar, descomprimir, copiar y compilar.

```python
##FUNCIONES
#Importamos funciones
import requests, zipfile, os, shutil, glob, yara

#Comprobamos si existe la carpeta y sino la creamos
def create(folder):
    if not os.path.exists(folder):
        os.mkdir(folder)

#Copiamos los ficheros a la carpeta de destino
def copyfiles(src,dst):
    for root, dirs, files in os.walk(src):
        for filename in files:
            if(".yara" in filename or ".yar" in filename):
                shutil.copy(os.path.join(root, filename), os.path.join(dst,filename))

#Descargamos el repositorio en formato ZIP
def download(dst, path):
    r = requests.get(path)
    open(dst, "wb").write(r.content)

#Descomprimimos el ZIP
def unzip(filename, dst):
    with zipfile.ZipFile(filename, "r") as zip_ref:
        zip_ref.extractall(dst)

#Localizamos y compilamos las reglas Yara
def compile(filepaths, save_folder):
    compiled_rules = dict()
    for folder in filepaths:
        for filename in glob.glob(folder + "/*.yar*"):
            namespace = os.path.basename(os.path.splitext(filename)[0])
            compiled_rules[namespace] = filename
    rules = yara.compile(filepaths = compiled_rules)
    if os.path.exists(save_folder):
        os.remove(save_folder)
    rules.save(save_folder)
```

2º -> Creamos las variables que vamos a usar, en este caso, se tratan de rutas donde se encuentran los ficheros que vamos a descargar, descomprimir y copiar.

```
##VARIABLES
#Definimos directorio raiz
root = os.path.dirname(os.path.abspath(__file__))

#Definimos carpeta para reglas compiladas
compiled_rules = os.path.join(root,"rules", "rules-compiled")

#Ficheros ZIP
cape_filename = os.path.join(root, "CAPEv2.zip")
reversingLabs_filename = os.path.join(root, "reversinglabs-yara-rules-develop.zip")
MalGamy_filename = os.path.join(root, "YARA_Rules-main.zip")
IrishIRL_filename = os.path.join(root, "yara-rules-main.zip")
PhishingKit_filename = os.path.join(root, "PhishingKit-Yara-Rules-master.zip")
mwbar_filename = os.path.join(root, "yara-rules-master.zip")
malpedia_filename = os.path.join(root, "signator-rules-main.zip")
droberson_filename = os.path.join(root, "yararules-master.zip")

#Carpeta descomprimida
capev2_folder = os.path.join(root, "CAPEv2-master")
yara_cape_folder = os.path.join(root, "CAPEv2-master", "data", "yara", "CAPE")
reversingLabs_folder = os.path.join(root, "reversinglabs-yara-rules-develop")
MalGamy_folder = os.path.join(root, "YARA_Rules-main")
IrishIRL_folder = os.path.join(root, "yara-rules-main")
PhishingKit_folder = os.path.join(root, "PhishingKit-Yara-Rules-master")
mwbar_folder = os.path.join(root, "yara-rules-master")
malpedia_folder = os.path.join(root, "signator-rules-main")
droberson_folder = os.path.join(root, "yararules-master")

#Carpetas locales
local_cape_folder = os.path.join(root, "rules", "Cape")
local_reversingLabs_folder = os.path.join(root, "rules", "reversingLabs")
local_MalGamy_folder = os.path.join(root, "rules", "MalGamy")
local_IrishIRL_folder = os.path.join(root, "rules", "IrishIRL")
local_PhishingKit_folder = os.path.join(root, "rules",  "PhishingKit")
local_mwbar_folder = os.path.join(root, "rules",  "mwbar")
local_malpedia_folder = os.path.join(root, "rules",  "malpedia")
local_droberson_folder = os.path.join(root, "rules",  "droberson")

#Carpetas para compilar
directories = [local_cape_folder, local_reversingLabs_folder, local_MalGamy_folder, local_IrishIRL_folder, local_PhishingKit_folder,
 local_mwbar_folder, local_malpedia_folder, local_droberson_folder]
```

3º > En esta parte es donde se ejecuta nuestro código. En ella usamos para cada uno de los repos que queremos usar con reglas yara las funciones que hemos creado y finalmente compilamos todas las reglas en un único ejecutable que usaremos para comparar el malware que deseemos.

```
##CODIGO
#Creamos carpeta "rules"
create(folder="rules")

#CAPEv2
download(dst=cape_filename, path="https://codeload.github.com/kevoreilly/CAPEv2/zip/refs/heads/master")
unzip(filename=cape_filename, dst=root)
create(folder=local_cape_folder)
shutil.copytree(src=yara_cape_folder, dst=local_cape_folder, dirs_exist_ok=True)
shutil.rmtree(capev2_folder)
os.remove(cape_filename)

#ReversingLabs
download(dst=reversingLabs_filename, path="https://codeload.github.com/reversinglabs/reversinglabs-yara-rules/zip/refs/heads/develop")
unzip(filename=reversingLabs_filename, dst=root)
create(folder=local_reversingLabs_folder)
copyfiles(reversingLabs_folder, local_reversingLabs_folder)
shutil.rmtree(reversingLabs_folder)
os.remove(reversingLabs_filename)
```

```
#Compilamos reglas
compile(directories, compiled_rules)
```