Documentación del proyecto final

ICS202L-01 LABORATORIO
ALGORITMOS MALICIOSOS

MARÍA HAMILTON - 1117462

FECHA: 25/01/2025

Descripción

Este script es una simulación educativa diseñada para demostrar cómo funciona el encriptado de archivos y la interacción con el usuario en un entorno controlado. Al ejecutarse, el programa busca archivos con las extensiones .pdf, .png, .jpg, .jpeg, y .txt en el escritorio del usuario y los cifra utilizando un algoritmo de cifrado simétrico.

Posteriormente, muestra un mensaje de advertencia en una ventana emergente (GUI) indicando que los archivos han sido encriptados. El usuario tiene la opción de "pagar" o "no pagar". Si el usuario decide no pagar, el programa provoca intencionadamente una "pantalla azul de la muerte" (BSOD) como medida demostrativa y elimina los archivos encriptados. Si el usuario paga, los archivos son desencriptados.

Advertencia

Este script es únicamente para fines educativos y de investigación.
No debe ser utilizado en situaciones reales o con intenciones maliciosas. El desarrollo y uso de programas que dañen dispositivos, cifren archivos sin autorización, o provoquen interrupciones en sistemas informáticos son ilegales en muchas jurisdicciones y violan principios éticos fundamentales.

Librerías y Dependencias

Biblioteca `cryptography.fernet`

Es parte de la biblioteca cryptography, diseñada para realizar cifrado simétrico de datos mediante el uso del algoritmo AES (Advanced Encryption Standard) en modo CBC con HMAC para garantizar la integridad.

Biblioteca `os`

Biblioteca estándar de Python para interactuar con el sistema operativo. Trabaja con rutas de archivos y directorios (e.g., os.path.join, os.getcwd), gestionar procesos y ejecutar comandos del sistema operativo, manipular el entorno del sistema, como variables de entorno o permisos.

Biblioteca `tkinter`

Biblioteca estándar de Python para construir interfaces gráficas de usuario (GUIs). Proporciona herramientas para crear ventanas, botones, etiquetas, cuadros de texto y otros elementos visuales.

Biblioteca `ctypes`

Biblioteca de Python que permite interactuar con bibliotecas y funciones en lenguaje C (incluyendo APIs del sistema operativo). Permite realizar llamadas directas a funciones de bajo nivel.

Descripción de Funciones

`generate_key():` Genera una clave de cifrado segura para usar con el algoritmo de cifrado simétrico.

`encrypt(data):` Encripta los datos pasados como parámetro utilizando la clave generada.

`decrypt(data):` Desencripta los datos encriptados utilizando la clave generada.

`getcwd():` Obtiene el directorio de trabajo actual del script.

`remove(path):` Elimina el archivo ubicado en la ruta especificada (path).

`walk(directory):` Recorre un directorio y sus subdirectorios de manera recursiva, devolviendo una lista de tuplas con las rutas de los directorios, subdirectorios y archivos.

`Tk():` Crean una ventana principal de la aplicación

`Label(root, text):` Crea un widget de etiqueta para mostrar texto en la ventana.

`messagebox.askquestion(title, message):` Muestra un cuadro de tipo pregunta.

`messagebox.showinfo(title, message):` Muestra un cuadro de mensaje informativo con el título y el mensaje proporcionado.

`root.after(ms, func):` Llama a la función func después de un retraso especificado en milisegundos (ms).

`root.mainloop():` Inicia el bucle de eventos de Tkinter para mostrar la interfaz gráfica y esperar interacciones del usuario.

`windll.ntdll.RtlAdjustPrivilege(c_uint(19), c_uint(1), c_uint(0), byref(c_int())):` Ajusta los privilegios del proceso actual. Se utiliza aquí para otorgar privilegios que permitan provocar un BSOD (pantalla azul de la muerte).

`windll.ntdll.NtRaiseHardError(c_ulong(0xC000007B), c_ulong(0), nullptr, nullptr, c_uint(6), byref(c_uint())):` Provoca un error grave en el sistema, generalmente un BSOD.

`encrypt():` Encripta los archivos con las extensiones especificadas y guarda una clave de cifrado.

`decrypt(key_file_path):` Desencripta los archivos usando la clave almacenada en key_file_path.

`remove_files_with_extensions(directory, extensions):` Elimina archivos con las extensiones especificadas en un directorio y sus subdirectorios.

`BSOD():` Realiza una serie de acciones para provocar un BSOD, primero eliminando los archivos y luego ajustando los privilegios y generando el error.

`show_popup():` Muestra un cuadro de mensaje solicitando al usuario si desea pagar o no. Dependiendo de la respuesta, procede con la eliminación de archivos o la restauración de ellos.

Flujo general del programa

1. **Inicialización**:
   - Importa las bibliotecas necesarias.
   - Define las extensiones de archivos a procesar.
2. **Cifrado**:
   - Cifra todos los archivos con las extensiones específicas en el directorio actual y subdirectorios.
   - Guarda la clave de cifrado en un archivo.
3. **Interfaz gráfica**:
   - Muestra una ventana con un mensaje de advertencia, exigiendo el pago.
4. **Decisión del usuario**:
   - Si el usuario decide pagar, los archivos son descifrados y el programa termina.
   - Si no paga, se eliminan los archivos y se fuerza un BSOD.
5. **Acciones finales**:
   - Dependiendo de la elección del usuario, el sistema puede restaurar los archivos o sufrir un error crítico.

Herramientas para Análisis del Malware

Pylint

Herramienta de análisis estático para Python que detecta errores en el código, identifica problemas de estilo y ofrece recomendaciones para mejorar la calidad y seguridad.

```
py -m pip install pylint
```

```
py -m pylint doggie.py
```

Uncompyle6

Herramienta para descompilar archivos .pyc y recuperar el código fuente original de Python. Para su uso, debe de tener un archivo `.pyc`.

```
py -m pip install uncompyle6
```

```
py -m uncompyle -o . doggie.pyc
```

Pdb

Usado para análisis dinámico es un depurador que permite pausar, inspeccionar scripts línea por línea. Para su uso, debe de importar la biblioteca `import pdb` y escribir esta línea al final del código `pdb.set_trace()`