Práctica 02: Procesamiento por Lotes 2

Maestro:

Javier Rosales Martinez

Materia:

Seminario de Solución de Problemas de Sistemas Operativos

Sección:

D06

Alumno:

Alejandro Covarrubias Sánchez

Código:

221350192

Antecedentes

Algunos antivirus utilizan el procesamiento por lotes para analizar múltiples archivos en un directorio de manera secuencial y organizada. En lugar de examinar cada archivo individualmente en tiempo real, el procesamiento por lotes permite al antivirus acumular un conjunto de archivos y analizarlos todos juntos en una sola operación, lo que puede ser más eficiente y rápido cuando se trata de volúmenes grandes de datos.

Estos antivirus primero escanean los directorios seleccionados, identificando todos los archivos presentes. Luego, estos archivos se agrupan en lotes y se comparan con una base de datos de firmas de virus conocidas, buscando patrones de código malicioso. Además, pueden aplicar análisis heurísticos para detectar comportamientos sospechosos en archivos no registrados en su base de datos. Una vez que el procesamiento del lote se completa, el antivirus genera un informe sobre los archivos detectados, tomando acciones como eliminar, poner en cuarentena o reparar los archivos sospechosos o infectados.

Metodología

El programa está escrito en Python y utiliza las librerías `os`, `schedule` y `time` para automatizar la búsqueda y eliminación de archivos sospechosos en un directorio. Se tienen 2 funciones principales, `get_evil_files()` y `delete_files()`. La primera utiliza `os.walk()` para recorrer de manera recursiva los directorios y subdirectorios, buscando archivos con una extensión específica. Los archivos encontrados se almacenan en una lista, y luego son eliminados por la segunda función, utilizando `os.remove()`, y manejando errores en caso de que algún archivo no pueda eliminarse.

La tarea de búsqueda y eliminación se programa para ejecutarse periódicamente dentro de la función `scheduled_task()`, usando la librería `schedule`, lo que permite que el proceso se repita automáticamente cada 30 segundos sin intervención manual.

El bucle principal mantiene el programa en ejecución, verificando las tareas programadas y haciendo pausas de 1 segundo con `time.sleep()` para optimizar el uso del CPU. El diseño modular del programa facilita su mantenimiento, y el manejo de errores asegura que el programa siga funcionando incluso si hay problemas con algunos archivos.

```
def get_evil_files(directory, evil_extension):
    """
    Recorre el directorio y sus subcarpetas para encontrar archivos con la extensión maligna.
    Devuelve una lista de rutas de archivos malignos.
    """
    evil_files = []
    for root_dir, sub_dir, files in os.walk(directory):
        for file in files:
            if file.endswith(evil_extension):
                file_dir = os.path.join(root_dir, file)
                evil_files.append(file_dir)
    return evil_files
```

```
def scheduled_task():
    """
    Función que se ejecuta periódicamente para buscar y eliminar archivos malignos.
    """
    directory = "D:/test"
    evil_extension = ".pdf"
    evil_files = get_evil_files(directory, evil_extension)
    if evil_files:
        delete_files(evil_files)
    else:
        print("No se encontraron archivos malignos en este lote.")
```

Conclusión

El programa actual permite buscar y eliminar archivos con una extensión específica en un directorio y sus subdirectorios, de manera periódica y automática. Ya es capaz de recorrer los directorios usando `os.walk()`, identificar archivos sospechosos y eliminarlos con `os.remove()`. La tarea está programada para ejecutarse en intervalos regulares (cada 30 segundos) mediante la librería `schedule`, lo que asegura un monitoreo continuo sin intervención manual. Además, el manejo de errores garantiza que el programa no se detenga si falla al eliminar un archivo.

También hay una optimización básica en el uso de recursos con el uso de `time.sleep()`, que evita un consumo excesivo de CPU. Sin embargo, el programa podría añadiendo características como el análisis de múltiples tipos de archivos maliciosos (no solo una extensión), la posibilidad de mover los archivos sospechosos a una carpeta de cuarentena en lugar de eliminarlos directamente, o la integración de un mejor sistema de notificaciones para alertar al usuario cuando se encuentren y eliminen archivos sospechosos. Además, se podrían implementar mejoras en la seguridad, como la verificación de permisos antes de eliminar archivos o un registro más detallado de las acciones realizadas.

Referencias

Amazon Web Services. (n.d.). *¿Qué es el procesamiento por lotes? - Explicación de los sistemas de procesamiento por lotes*. AWS. https://aws.amazon.com/es/what-is/batch-processing/