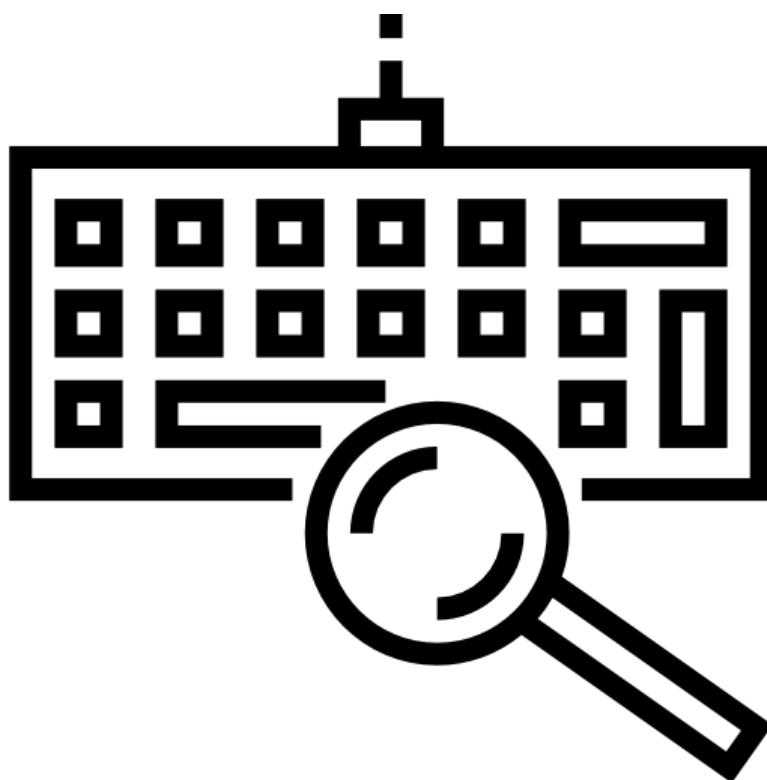


INFORME ANÁLISIS Y DETECCIÓN DE KEYLOGGERS EN SISTEMAS WINDOWS



Autor: **Joan David Torres Garcia**

1. Introducción

En el ámbito de la ciberseguridad, los keyloggers representan una de las amenazas más peligrosas para la privacidad y la seguridad de los datos personales. Estos programas maliciosos están diseñados para registrar las pulsaciones del teclado sin el conocimiento del usuario, lo que puede llevar al robo de credenciales, información bancaria y datos sensibles.

Este informe tiene como objetivo presentar un análisis detallado sobre los keyloggers y describir las herramientas que desarrollé para su detección en sistemas Windows utilizando Python.

2. Análisis de Keyloggers

2.1 ¿Qué es un Keylogger?

Un keylogger es un software o hardware diseñado para registrar todas las pulsaciones del teclado de un usuario sin su conocimiento. Existen dos tipos principales:

- **Keyloggers de Software:** Programas maliciosos que se ejecutan en segundo plano y capturan cada tecla presionada.
- **Keyloggers de Hardware:** Dispositivos físicos conectados entre el teclado y el PC que almacenan la información ingresada.

2.2 Métodos de Instalación y Persistencia

Los atacantes pueden instalar keyloggers a través de:

- Archivos adjuntos maliciosos en correos electrónicos (phishing).
- Descargas de software infectado.
- Explotación de vulnerabilidades en el sistema.
- Acceso físico al dispositivo para instalar hardware espía.

Una vez instalado, un keylogger puede ocultarse en el sistema utilizando técnicas como la **inyección de procesos**, la **modificación del Registro de Windows** o la **persistencia en el arranque del sistema**.

3. Métodos de Detección de Keyloggers

Para detectar keyloggers en Windows, desarrollé un conjunto de scripts en Python que permiten realizar diferentes análisis en el sistema:

1. **Análisis de Procesos en Ejecución:** Busca programas sospechosos que puedan estar registrando el teclado.
2. **Detección de Hooks en el Sistema:** Revisa si algún programa está utilizando hooks de teclado.

A continuación, presento cada uno de estos métodos junto con su código.

4. Implementación de Herramientas de Detección

4.1 Detección de Procesos Sospechosos

Este script revisa los programas que están corriendo en nuestro PC y busca si alguno tiene un nombre sospechoso, como "keylogger", "spy" o "capture", que podrían indicar que alguien está grabando lo que escribes o ves en pantalla.

1. **Busca todos los procesos en ejecución** (los programas que están abiertos).
2. **Compara sus nombres con una lista de palabras sospechosas.**
3. **Si encuentra coincidencias**, guarda el ID del proceso (PID) y su nombre.
4. **Muestra los procesos sospechosos en pantalla.**
5. **Si no encuentra nada raro**, te avisa que todo está bien.

```
import psutil

def detectar_procesos_sospechosos():
    palabras_clave = ["keylogger", "hook", "logger", "capture", "spy", "record", "sniffer"]
    procesos_detectados = []

    for proceso in psutil.process_iter(attrs=['pid', 'name']):
        try:
            nombre_proceso = proceso.info['name'].lower()
            if any(palabra in nombre_proceso for palabra in palabras_clave):
                procesos_detectados.append((proceso.info['pid'], nombre_proceso))
        except (psutil.NoSuchProcess, psutil.AccessDenied, psutil.ZombieProcess):
            continue

    return procesos_detectados

if __name__ == "__main__":
    procesos = detectar_procesos_sospechosos()
    if procesos:
        print("⚠️ Procesos sospechosos detectados:")
        for pid, nombre in procesos:
```

```
        print(f"PID: {pid} | Nombre: {nombre}")
    else:
        print("✅ No se encontraron procesos sospechosos.")
```

4.2 Detección de Hooks de Teclado en Windows

Este script analiza nuestro PC en busca de posibles *keyloggers* (programas que registran lo que escribes en el teclado). Para hacerlo, usa dos métodos principales:

1. Detección de hooks de teclado

Los *keyloggers* suelen interceptar las teclas usando un "hook" especial en Windows.

El script intenta establecer un *hook* propio. Si falla, significa que otro programa ya está usando ese método, lo que podría indicar un *keylogger*.

2. Escaneo de procesos sospechosos

Busca programas en ejecución con nombres comunes de *keyloggers* (como `keylogger.exe` o `spyware.exe`).

Si encuentra alguno, te avisa para que puedas investigarlo.

```
import ctypes
import ctypes.wintypes
import psutil
```

```
WH_KEYBOARD_LL = 13
user32 = ctypes.windll.user32
kernel32 = ctypes.windll.kernel32
```

```
def detectar_hooks():
    print("🔍 Analizando hooks de teclado en el sistema...")

    hook = user32.SetWindowsHookExW(WH_KEYBOARD_LL, 0, 0, 0)

    if hook == 0:
        print("⚠️ Posible keylogger detectado en el sistema.")
    else:
        print("✅ No se encontraron hooks sospechosos.")
        user32.UnhookWindowsHookEx(hook)
```

```
def detectar_procesos_sospechosos():
    print("🔍 Analizando procesos en ejecución...")
    procesos_sospechosos = ["keylogger.exe", "hook.exe", "spyware.exe"]

    for proceso in psutil.process_iter(['pid', 'name']):
```

```

        if proceso.info['name'].lower() in procesos_sospechosos:
            print(f"⚠️ Proceso sospechoso detectado: {proceso.info['name']} (PID: {proceso.info['pid']})")
        print("✅ Análisis de procesos finalizado.")

if __name__ == "__main__":
    detectar_hooks()
    detectar_procesos_sospechosos()

```

5. Cómo Probar los Scripts

5.1 Requisitos Previos

Antes de ejecutar los scripts, aseguré que mi sistema tuviera Python instalado y configuré las bibliotecas necesarias con el siguiente comando:

```
pip install psutil pygetwindow pyautogui keyboard
```

```

C:\Windows\System32>pip install psutil pygetwindow pyautogui keyboard
Collecting psutil
  Downloading psutil-7.0.0-cp37-abi3-win_amd64.whl.metadata (23 kB)
Collecting pygetwindow
  Downloading PyGetWindow-0.0.9.tar.gz (9.7 kB)

```

5.2 Ejecución de los Scripts

Cada script se puede ejecutar en la terminal de Windows con el comando:

```
python nombre_del_script.py
```

Por ejemplo, para detectar procesos sospechosos:

```

C:\Users\Usuario\Desktop\Deteccion>python detectar_procesos_sospechosos.py
✅ No se encontraron procesos sospechosos.

```

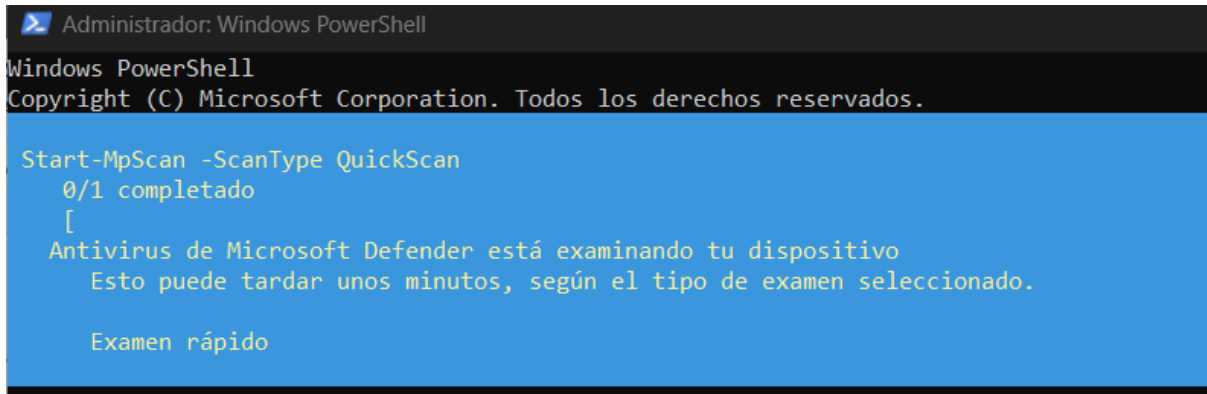
5.3 Qué Hacer Si Se Detecta un Keylogger

Si alguno de los scripts muestra actividad sospechosa:

1. Verifiqué los procesos detectados en el **Administrador de Tareas (Ctrl + Shift + Esc)**.
2. Si encontraba un proceso sospechoso, lo terminaba manualmente o con el Administrador de Tareas.

Ejecute un análisis con **Windows Defender**:

Start-MpScan -ScanType QuickScan



```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Start-MpScan -ScanType QuickScan
0/1 completado
[
Antivirus de Microsoft Defender está examinando tu dispositivo
Esto puede tardar unos minutos, según el tipo de examen seleccionado.

Examen rápido
```

3. Eliminé cualquier archivo sospechoso encontrado.

6. Medidas de Prevención

Para evitar la instalación de keyloggers, tendríamos que seguir las siguientes medidas de seguridad en nuestro sistema:

- Mantener nuestro sistema **siempre actualizado**.
- No descargar software de **fuentes desconocidas**.
- Activar **autenticación en dos pasos** en cuentas importantes.
- Usar **Windows Defender y un firewall** para bloquear software malicioso.

7. Conclusión

Los keyloggers son una seria amenaza para la seguridad informática, pero con las herramientas adecuadas es posible detectarlos y eliminarlos.

Este informe presenta un conjunto de **scripts en Python** que permiten identificar posibles keyloggers en un sistema Windows mediante el análisis de procesos, monitoreo de teclado y detección de hooks del sistema.

Gracias a estos métodos, puedo monitorear la seguridad de mi equipo y tomar medidas preventivas en caso de detectar actividad sospechosa.

8. Referencias

- Microsoft Documentation: API de Windows.
 - OWASP: Guía de detección de malware.
 - NIST: Prácticas recomendadas en ciberseguridad.
-