Zer0Leaks — Data Leakage Prevention (DLP) Solution

---

1. How It Works

The system acts as a **background security guard** for your computer. It continuously watches three main **"exit points"** where data often leaves an organization:

Monitored Areas

1. **File System**
   Watches specified folders (including Desktop, Documents, Downloads).
   If a file is created or modified, its content is immediately read.
2. **Clipboard**
   Monitors anything you **Copy (Ctrl+C)**.
3. **External Drives (USB)**
   Automatically detects when a USB drive is inserted and begins monitoring it for sensitive files.

Once text is captured, it is sent to the **Detector**, which scans for sensitive information.
If detected, the system:

- Logs a warning using **color-coded output**
- Saves the event into a log file

---

2. Features

Interactive CLI Menu

A user-friendly dashboard to:

- Add/remove directories
- Toggle USB scanning
- Open logs while the tool runs

Visual Feedback

- **Startup Banner**: Displays "Zer0Leaks" ASCII art with a clear status summary
- **Color-Coded Logs**:
  - **Red** → Monitoring Warnings (Local Files)
  - **Purple** → USB / External Drive Warnings
  - **Yellow** → Clipboard Warnings
  - **Blue** → Informational messages

Real-Time Monitoring

- **Clipboard Monitoring**: Detects sensitive data instantly
- **File System Monitoring**: Watches specific directories (plus default user folders)
- **USB Auto-Detection**: Automatically identifies removable drives
- **Hybrid Detection**: Combines strict rules (Regex) + smart guessing (AI / NLP)
- **Startup Scan**: Scans existing files in watched directories immediately on launch

---

3. What is NLP (Natural Language Processing)?

NLP is a field of Artificial Intelligence that helps computers understand human language.

Comparison

- **Standard Programming (Regex)**
  Looks for exact shapes.
  Example: *"Find a word with an @ symbol"* — rigid and limited.
- **NLP (Spacy)**
  Understands *context*.
  Example:
  - "Apple is a fruit"
  - "Apple Inc. stock price"

  These mean different things — NLP knows that.

In this project, NLP is used to detect **Named Entities**, such as:

- Names of people
- Organizations
- Countries
- Monetary values

These are infinite in variety and impossible to manually list.

---

4. Components & Libraries

Files

File	Purpose
main.py	The **Manager** — Handles arguments, displays the Interactive Menu, and orchestrates the monitors.
src/monitor.py	The **Eyes** — Listens for file changes (watchdog) and clipboard updates (pyperclip). Manages watched paths dynamically.
src/detector.py	The **Brain** — Decides if text is "sensitive". Holds Regex patterns and loads the Spacy NLP model.
src/logger.py	The **Scribe** — Custom logging system with colored output and file logging (dlp_log.log).
src/banner.py	The **Face** — Handles ASCII art display and screen clearing.
src/usb_detector.py	The **Gatekeeper** — Uses Windows API to detect removable drives.

Libraries

Library	Role
watchdog	Efficiently waits for file system events (Create/Modify).
pyperclip	Allows Python to read/write to the system clipboard.
spacy	Industrial-grade NLP library for Named Entity Recognition.
colorama	Cross-platform colored terminal text.
ctypes	(Built-in) Used to interface with Windows Kernel for drive detection.

---

5. Constraints & Rules (Detection Logic)

File Constraints

- **Monitored Extensions** (text-based only):
  .txt, .csv, .log, .md, .json, .xml

- **Ignored Directories**:
  .git, .vscode, __pycache__, .venv, src
  (Prevents loops and errors.)
- **Performance Note**:
  Large files are supported, but extremely large logs may cause short pauses.

Detection Rules (Triggers)

The system flags data as **"Sensitive"** if it matches:

A. Strict Patterns (Regex)

- **Email** → anything@anything.anything
- **SSN** → xxx-xx-xxxx
- **Credit Card** → 13–16 digit sequences
- **Keywords** → confidential, private, secret, restricted (case-insensitive)

B. Smart Context (NLP)

- **PERSON** → Names of people (e.g., *John Doe*)
- **ORG** → Organizations (e.g., *Google*, *FBI*)
- **GPE** → Countries, Cities (e.g., *Paris*, *China*)
- **MONEY** → Monetary values (e.g., *$500*, *1 million dollars*)

---

6. How to Run & Use

Installation

1. Install Python 3.x
2. Install dependencies:
3. pip install -r requirements.txt
4. python -m spacy download en_core_web_sm

Running

Open a terminal and run:

python main.py

Command Line Arguments

- `--external` → Enable USB monitoring at startup
- `--path "C:/Path/To/Folder"` → Monitor a specific directory
- `--no-user-dirs` → Disable Desktop, Documents, and Downloads

Interactive Menu Controls

While running, you can:

- **[1] Add Directory** → Start watching a folder
- **[2] Remove Directory** → Stop watching a folder
- **[4] Toggle USB Scanner** → Enable / Disable USB monitoring
- **[6] Start/Resume Monitoring** → Enter active monitoring mode
- **Ctrl + C** → Pause monitoring and return to menu