Python Resume Generator

Technical Overview

The Resume Generator CLI project automates PDF résumé production using a Jinja2–LaTeX pipeline. Although the tool itself is non-security related, the development approach mirrors patterns used in security automation tooling such as reporting frameworks, templated evidence generators, and data-driven documentation systems.

Objective

The thought of leaking personal data on multiple online 'résumé makers' really frightened me, so I myself built a reproducible document-generation workflow that eliminates manual editing errors and ensures consistent formatting across multiple outputs similar to how security teams generate standardized assessment reports.

Technical Summary

- Implemented a structured input model using nested Python dictionaries to enforce data consistency and reduce template injection risks.
- Integrated Jinja2 with custom delimiters to avoid interpreter collisions and ensure safe document rendering.
- Added recursive sanitization to mitigate LaTeX meta-character ingestion, preventing template breakage in the rendering phase.
- Automated compilation via pdflatex to reduce manual build steps and ensure deterministic output.

Security-Relevant Considerations

- Implemented character-level input sanitization to prevent user-supplied fields from causing compiler errors similar to defensive encoding used in secure templating.
- Enforced strict separation between data and templates to avoid unsafe direct insertion.
- Leveraged subprocess execution with controlled arguments to prevent command injection scenarios.

Outcome

An automation-friendly document generation workflow consistent with security reporting standards and capable of supporting batch generation, CI pipelines, and future UI integration.