UC SANTA BARBARA

Optimizing Python Vulnerability Detection with LLM-enhanced Rule Generation

Weiheng Bai[1], Evelyn Gutierrez[2], Austin Chan[3], Lukas Dresel[4]
1. University of Minnesota 2. California State University, San Bernardino 3. University of California, Berkeley 4 University of California, Santa Barbara

Introduction

Background

- The rapid detection and patching of code vulnerabilities are crucial in minimizing the window of exploitation by malicious attackers.
- *Static Application Security Testing (SAST)* tools are designed to aid developers in identifying such vulnerabilities.
- SAST tools rely on ***pattern matching*** to detect vulnerabilities.

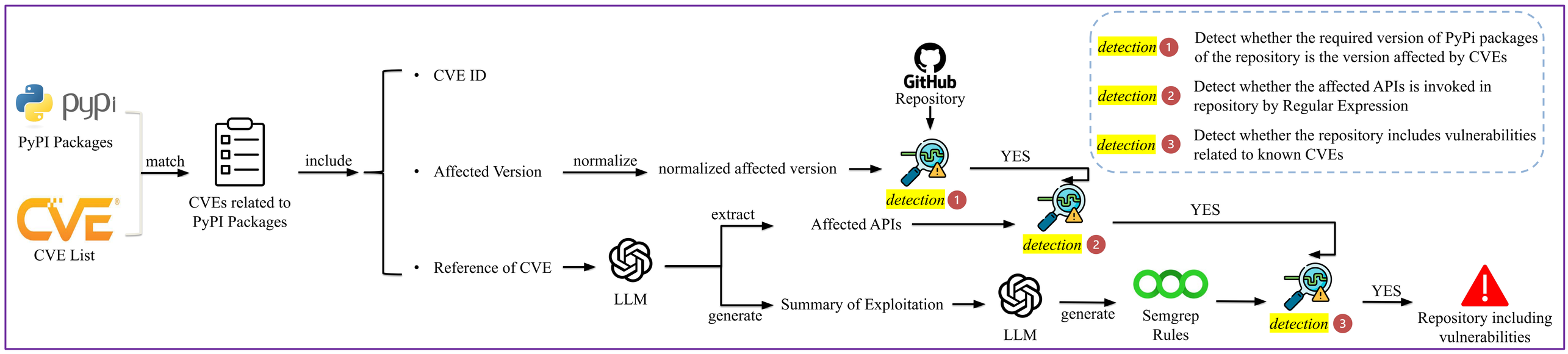
Limitation of SAST tools

- The generation of vulnerability detection rule patterns currently relies ***heavily on extensive manual effort***.
- These rule patterns are often incomplete, contributing to the **low detection rates** observed in SAST tools.
- The rule patterns **are unable to identify newly reported vulnerabilities** in a timely manner.

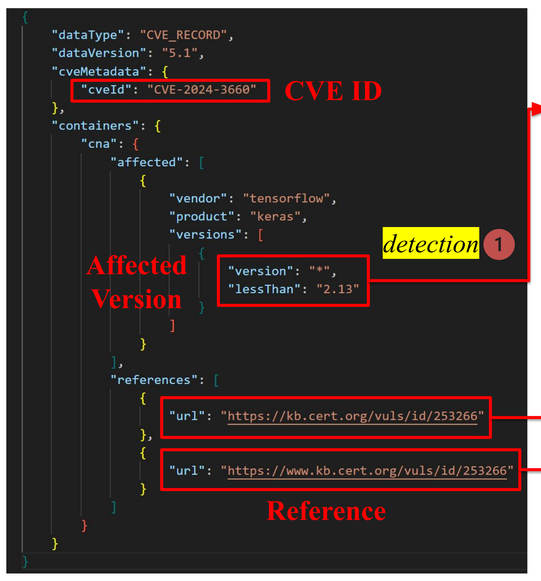
Research Questions

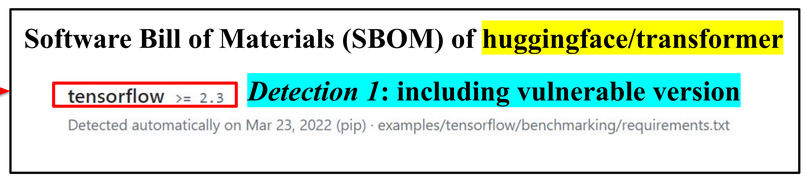
- ***RQ1:*** How to automatically generate the pattern rules?
- ***RQ2:*** How to improve the detection rates?
- ***RQ3:*** How to catch the newly reported vulnerabilities in real time?

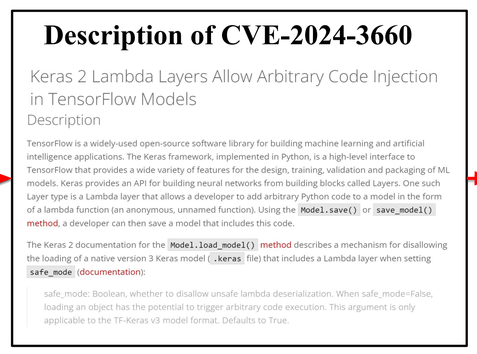
Architecture

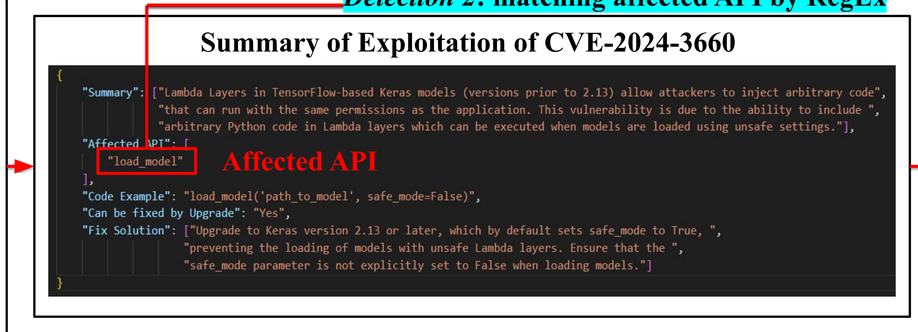


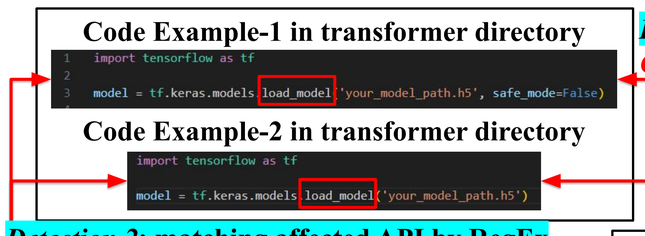
Detection Example (CVE-2024-3660)

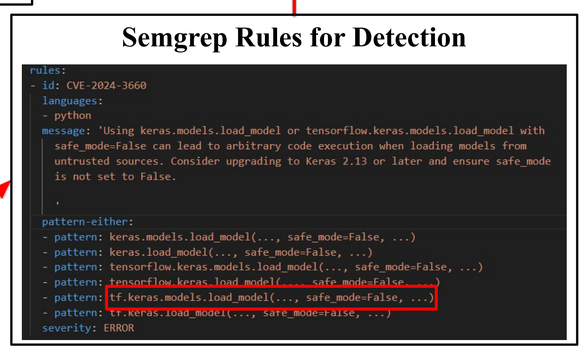












Preliminary Result

Dataset Selection

- We generated Semgrep Rules for **392 CVEs** related to Top 100 PyPI packages since 2019.
- We process our 3 detections on **50, 681** Github repositories.

Finding

- Result of Detection 1: ***11691 repositories*** include vulnerable version of packages in SBOM.
- Result of Detection 2: ***4532 repositories*** invoke the vulnerable APIs
- Result of Detection 3: ***3210 repositories*** are truly affected by CVE vulnerabilities

Accuracy of components

- Accuracy of affected version normalization: 97.27%
- Accuracy of CVEs matching with PyPI packages: 64.49%
- Accuracy of API extraction: 78.91%
- Accuracy of Semgrep rules generation: 80.87%

Limitations

- Challenges in Achieving High Accuracy in Matching CVEs with PyPI Packages
  - The current approach for matching CVEs with PyPI packages relies on *product* and *vendor* names. Some of CVEs lack these identifiers, which hinders effective matching.
- Issues with Accuracy in API Extraction
  - The accuracy of API extraction is compromised by the current methodology, involving selective analysis by LLMs of a subset of references to minimize computational costs. However, this approach leads to inaccuracies, as some references include information on multiple CVEs. Consequently, the LLM may inadvertently extract APIs related to CVEs beyond the specific one being targeted.
- Inaccuracies in Semgrep Rules Generation
  - The generation of Semgrep rules also faces accuracy issues, primarily due to the presence of grammatical errors within the YAML syntax. These errors are likely attributable to the limited quantity and quality of Semgrep rules in the training datasets used by the LLM, resulting in suboptimal rule generation.

Future Work

- To improve the accuracy of matching CVEs with PyPI packages, we will integrate additional features, such as the URLs of references, into the matching algorithm.
- To address the challenges in API extraction, a novel method is proposed: first, generate summaries for each reference, then compile these summaries to enable LLMs to extract the correct affected APIs and summaries of exploitation more accurately.
- Improving the accuracy of Semgrep rules generation can be achieved by constructing a benchmark dataset that includes both exploitation summaries and Semgrep rules. This dataset can then be used to fine-tune an LLM specifically for the purpose of generating accurate Semgrep rules.
- Extending the evaluation process to include a broader range of PyPI packages and GitHub repositories.

Reference

AI Institute for Agent-based Cyber Threat Intelligence and Operation

Supported by the National Science Foundation under grant no. IIS-2229876 and is supported in part by funds provided by the NSF, by the Department of Homeland Security, and by IBM.

https://action.ucsb.edu