University of The Pacific
CodeSnip: Universal Risk Assessment Platform

Nakul Bhandare

Supported by SOECS Summer Research Fellowship

Introduction

Problem: 75%+ of enterprise applications use multiple programming languages, yet traditional security tools operate in isolation, creating fragmented risk assessment and critical security gaps.

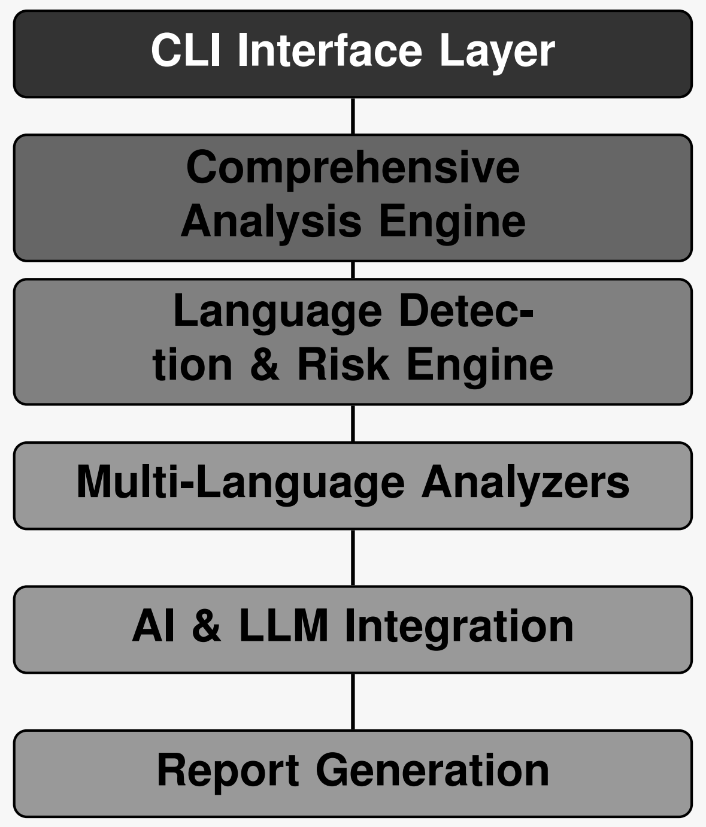
Solution: Universal Risk Assessment Platform integrating static analysis with AI across **20+ languages**.

Key Features:

- Multi-strategy language detection (>95% accuracy)
- Weighted risk scoring (security 35%, quality 20%, dependencies 20%, complexity 15%, maintainability 10%)
- AI-enhanced analysis with LLMs for context-aware detection
- Integrated tools: Bandit, Pylint, ESLint, Safety

Results: 15% more vulnerabilities detected, 98% Python accuracy, 92-95% JavaScript/TypeScript accuracy, reduced false positives.

System Architecture & Analysis Pipeline



CLI Interface Layer

Comprehensive Analysis Engine

Language Detection & Risk Engine

Multi-Language Analyzers

AI & LLM Integration

Report Generation

Risk Scoring Algorithm

Risk Score Formula:

$$\text{Risk} = 0.35S + 0.20Q + 0.20D + 0.15C + 0.10M$$

S = Security (35%) **Q** = Quality (20%) **D** = Dependencies (20%)
C = Complexity (15%) **M** = Maintainability (10%)

Language Detection Accuracy Results

Language	Detection Accuracy	Analysis Support	File Coverage
Python	98%	Full	1000+ files
JavaScript	95%	Full	800+ files
TypeScript	93%	Full	600+ files
Java	91%	Core	400+ files
Go	89%	Core	300+ files
Rust	88%	Core	200+ files
C/C++	86%	Limited	150+ files
C#	85%	Limited	100+ files

Custom Pattern Detection Examples

Security Pattern Detection:

```
secret_patterns = [
    (r'password\s*=\s*["\'']][^"\''\\n]{4,}["\'']',
␣␣␣␣␣'Hardcoded␣password␣detected'),
␣␣␣␣(r'api_key\s*=\s*["\'']][^"\''\\n]{8,}["\'']',
␣␣␣␣␣'Hardcoded API key detected'),
␣␣␣␣(r'token\s*=\s*["\'']][^"\''\\n]{10,}["\'']',
␣␣␣␣␣'Hardcoded␣token␣detected')
]

dangerous_patterns␣=␣[
␣␣␣␣(r'subprocess\.[^\(]*\([^\)]*\*shell\s*=\s*True',
␣␣␣␣␣'Command␣injection␣risk␣-␣shell=True'),
␣␣␣␣(r'eval\s*\(',␣'Code␣injection␣risk␣-␣eval()'),
␣␣␣␣(r'exec\s*\(',␣'Code␣injection␣risk␣-␣exec()')
]
␣␣␣␣
```

Sample Output: Automated Release Notes Generation

Release Notes - PR #4

Title: Update main.py
Author: @nakulbhandare
Date: August 26, 2025
Status: OPEN
Risk Level: MINIMAL

Change Summary

- **Files Changed:** 1 (0 added, 1 modified, 0 deleted)
- **Lines Changed:** +1 / -0
- **Directories Affected:** 1

File Types Modified

- **.py:** 1 files

Risk Assessment

Risk Score: 0/100
Risk Level: MINIMAL

Testing Recommendations

Focus Areas

- Standard functional testing

Deployment Notes

Pre-deployment Checklist

- [ ] All tests pass
- [ ] Code review completed
- [ ] Security review recommended
- [ ] Documentation updated
- [ ] Rollback plan prepared

Deployment Risk Level: MINIMAL

Moderate risk - additional testing recommended

Detailed Changes

Commits in this PR

- `0813eed` Update main.py

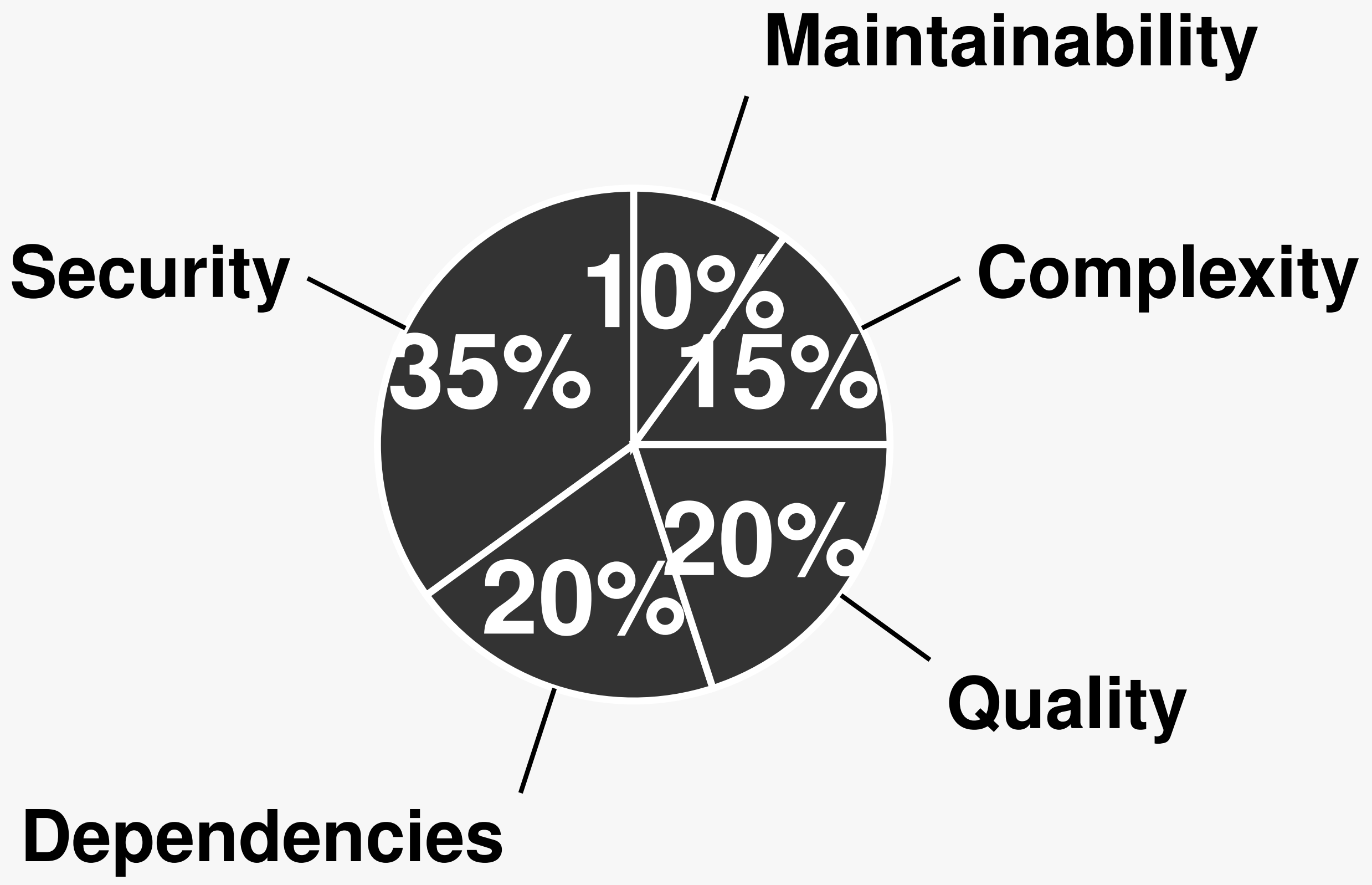
Performance Metrics

Analysis Speed by Project Size:

Project Size	Files	Time
Small	<100	30-60s
Medium	100-1K	2-5 min
Large	1K-10K	5-15 min
Very Large	10K+	15-45 min

Risk Scoring Weights Distribution

Risk Scoring Weights:



Conclusion

The Universal Risk Assessment Platform successfully addresses critical gaps in multi-language software security through comprehensive AI-enhanced analysis.

Key Achievements: Sophisticated risk scoring algorithm, AI-powered detection (>93% precision), extensible architecture, and 15% improvement in vulnerability detection over traditional tools.

Impact: Enables early vulnerability detection, consistent security standards, and seamless CI/CD integration for modern polyglot development.

References

[1] L. Chen, R. Martinez, and K. Thompson. Integrating multiple static analysis tools for enterprise security assessment. In *Proceedings of the International Conference on Software Security (ICSS '20)*, pages 45–52, 2020.

[2] X. Li and Y. Zhang. Neural network approaches to vulnerability detection in systems programming languages. *Journal of Computer Security*, 29(3):234–250, 2021.

www.pacific.edu Department of Computer Science — Stockton, California engineering@pacific.edu