Topic 1: Automated Vulnerability Assessment Tool Development

GitHub Project 1

OWASP Nettacker

- Citation: OWASP. (2024). OWASP Nettacker – Automated Penetration Testing Framework. GitHub. https://github.com/OWASP/Nettacker

- Project Type: Security Tool (Automated Reconnaissance & Vulnerability Assessment)

- Synopsis: Nettacker is a Python-based framework designed to automate information gathering, port/service scanning, vulnerability checks, credential brute-forcing, subdomain enumeration, and more. It supports multiple protocols (HTTP/HTTPS, FTP, SSH, SMB, etc.), produces reports in HTML/JSON/CSV, and includes a modular architecture for extending tasks. It currently has ~ 900 forks (per GitHub page) and is actively maintained. Limitations: broad scope but may not cover specialized application-level vulnerabilities deeply; more network/asset-scanning focus than source-code scanning.

- Link: https://github.com/OWASP/Nettacker

- Relevance Rating: 4/5 — Very relevant for automated vulnerability assessment tool development; though oriented more toward network/asset scanning than deep code-level vulnerability assessment.

GitHub Project 2

projectdiscovery nuclei

- Citation: ProjectDiscovery. (2024). Nuclei – Fast, customizable vulnerability scanner with YAML-based DSL. GitHub. https://github.com/projectdiscovery/nuclei

- Project Type: Security Tool (Vulnerability Scanner)

- Synopsis: Nuclei is a powerful, high-performance open-source scanner that uses a YAML-based templating language to enable custom vulnerability detection across applications, APIs, networks, DNS, cloud configurations, etc. It's used widely in security operations for custom scanning, has large community support (stars/forks suggest significant adoption), and is designed for automation in CI/CD pipelines. Limitations: While extremely capable, it focuses on templated scanning rather

than building from scratch an assessment engine; learning curve for creating custom templates.

- Link: https://github.com/projectdiscovery/nuclei

- Relevance Rating: 5/5 — Highly relevant: demonstrates automation of vulnerability assessment, template-based detection logic, integration with CI/CD — useful as reference or component in your tool development.

GitHub Project 3

infobyte Faraday

- Citation: Infobyte. (2024). Faraday – Open Source Vulnerability Management Platform. GitHub. https://github.com/infobyte/faraday

- Project Type: Security Tool / Vulnerability Management Platform

- Synopsis: Faraday is a platform that aggregates vulnerability scan results from multiple tools, normalizes them, and offers dashboards, visualizations, multi-user collaboration and remediation tracking. It allows users to integrate many scanners and maintain a historical database of findings. This is valuable because when developing an automated vulnerability assessment tool you'll likely need to consider not just detection, but also how findings are managed, triaged and tracked. Limitations: It is more focused on aggregation and management rather than new detection logic; may require considerable setup (database, Docker) and integration with other tools.

- Link: https://github.com/infobyte/faraday

- Relevance Rating: 4/5 — Good complement to detection-focused tools, especially for building out the workflow around tool output. Slightly less focused purely on "assessment tool development" than detection engines.

Topic 2:

Secure Code Review: Common Vulnerabilities in Python Applications

GitHub Project 1

PyCQA Bandit

- Citation: PyCQA. (2024). Bandit – Security static analyzer for Python code. GitHub. https://github.com/PyCQA/bandit

- Project Type: Security Tool (Static Code Analyzer for Python)

- Synopsis: Bandit analyses Python code by building an AST (abstract syntax tree) and running "plugins" to detect common security issues (e.g., unsafe uses of subprocess, insecure imports, etc.). It currently has ~ 7.4k stars, good community support, and supports integration with CI pipelines and containerized usage. Limitations: Focuses on certain classes of issues, may not cover complex logic vulnerabilities or framework-specific issues fully; possibly higher false-positive rate for some checks.

- Link: https://github.com/PyCQA/bandit

- Relevance Rating: 5/5 — Directly applicable to secure code review in Python; an excellent baseline tool to study and compare against.

GitHub Project 2

247arjun ai-secure-code-review

- Citation: 247arjun. (2025). ai-secure-code-review – Integrating static analysis with Generative AI for secure code review. GitHub. https://github.com/247arjun/ai-secure-code-review

- Project Type: Research Implementation / Educational Framework

- Synopsis: This repository explores combining static analysis (via tools like Semgrep) with generative AI (Azure OpenAI GPT models) to automate and enhance code reviews — including identifying vulnerabilities in code, proposing fixes, and scaling review processes. It has ~36 stars and is relatively new (6 commits as of last check, so minimal history). Limitations: Early-stage, fewer contributors, low adoption so far, limited documentation and real-world deployment evidence.

- Link: https://github.com/247arjun/ai-secure-code-review

- Relevance Rating: 4/5 — Very relevant and interesting for your topic, especially in the "automating secure code review" area; but less mature, so you'll need to evaluate its effectiveness and constraints.

GitHub Project 3

python-security pyt

- Citation: python-security. (2020). pyt – Static Analysis Tool for Detecting Security Vulnerabilities in Python Web Applications. GitHub. https://github.com/python-security/pyt

- Project Type: Security Tool / Static Analysis for Python (Research/Proof-of-Concept)

- Synopsis: 'pyt' is a static analysis tool designed for Python web applications. It uses control flow graphs, data flow analysis and fixed-point computations to detect issues such as command injection, SSRF, SQL injection, XSS and directory traversal. This is directly aligned with your topic of common vulnerabilities in Python code and code review. Limitations: The project appears older, may not be actively maintained, fewer stars/forks, and likely limited in coverage compared to more modern tools.

- Link: https://github.com/python-security/pyt

- Relevance Rating: 4/5 — Strong match for Python code review and vulnerability detection; some risk that the tool is outdated or less mature than alternatives.