Security Policy Document

Project: All-in-One CLI Pentesting Tool
Last Updated: August 2025
Maintainer: Roman Shubin

Section 1: Key Security Rules / Guidelines

1. Multi-Factor Authentication (MFA) for Access
All contributors must enable MFA on GitHub and any cloud or deployment platforms. This prevents unauthorized access even if credentials are compromised.

2. Code Review & Static Analysis Required
Every pull request must:

- Be reviewed by at least one trusted contributor
- Pass automated linting and static code analysis (e.g., Bandit for Python)
  This reduces the chance of introducing insecure code or logic flaws.

3. Secure Dependency Management

- All dependencies must be pinned with version numbers (e.g., `requests==2.31.0`)
- Packages must be reviewed for CVEs via tools like `safety` or `pip-audit`
- No unverified external scripts or tools may be used

Section 2: Incident Response Plan

This plan follows the **NIST framework** and applies to the CLI tool's development and usage lifecycle.

1. Detection

- Monitor CLI logs for unusual activity (e.g., auto-running exploits, rogue network calls)

- Get reports from users or contributors (via GitHub Issues or Discord)
- Watch GitHub activity for suspicious commits or credential leaks

2. Classification

- **Low:** Misconfigured settings or input bugs
- **Medium:** Code vulnerability or unauthorized access detected
- **High:** Malware injected into CLI tool, compromised release, or external data leak

3. Containment

- Remove malicious versions from public channels (GitHub, PyPI)
- Lock down the repo (disable merges, commits, CI/CD)
- Revoke affected API keys or tokens

4. Eradication

- Patch the source code
- Remove affected files or libraries
- Audit access logs and roll back to last safe version

5. Recovery

- Release a clean version with changelog
- Notify users through GitHub, Discord, or mailing list
- Re-test tool functionality in a controlled environment

6. Lessons Learned

- Conduct a post-mortem

- Strengthen policies or automation to prevent recurrence

Section 3: CIA Triad Alignment

CIA Principle	How This Policy Supports It
Confidentiality	- Environment variables and API keys are stored in `.env` files and never hardcoded - MFA prevents unauthorized access to source code
Integrity	- All code changes require peer review and static analysis - Git version control ensures traceability and rollback
Availability	- Redundant backups of the codebase and tool releases - Incident plan allows fast recovery after security breaches