



VaultScan Community Edition – User Guide (v1.2)

Confidential Notice

This guide is for **VaultScan – Community Edition v1.2** users only.
Unauthorized sharing, reproduction, or disclosure is prohibited.

Contents

VaultScan Community Edition – User Guide (v1.2)	1
1. Introduction.....	2
2. Installation Instructions	2
3. Basic Usage	2
4. Advanced Usage	2
5. Docker Support.....	3
6. PowerShell Launcher Script (scan.ps1).....	3
7. GitHub Actions Integration	3
8. Supported Secret Patterns (Community Edition)	4
9. Known Limitations (v1.2)	4
10. GitLab CI/CD Integration.....	5
11. Future Roadmap (Private Advanced Version).....	6
12. About the Author	7

1. Introduction

VaultScan Community Edition (v1.2) is a **privacy-first, offline-first CLI tool** built to detect secrets and credentials accidentally committed to code. It helps DevOps, Cloud, and Security teams identify risks early – without sending any data outside your machine.

2. Installation Instructions

Clone the Repository

```
git clone https://github.com/vaultscanhq/vaultscan-community.git
```

```
cd vaultscan-community
```

Install Python Dependencies

```
pip install -r requirements.txt
```

3. Basic Usage

CLI Command

```
python -m vaultscan.main --path ./path/to/your/code
```

Example:

```
python -m vaultscan.main --path ./tests/dummy_repo
```

4. Advanced Usage

Enable Verbose Mode (Recommended)

```
python -m vaultscan.main --path ./your/code --verbose
```

Using .vaultscanignore

Create a .vaultscanignore file in your project root to skip unwanted folders/files.

Sample:

```
node_modules/
```

```
tests/
```

```
*.jpg
```

```
*.png
```

```
*.md
```

VaultScan will ignore all files/folders matching patterns listed.

5. Docker Support

VaultScan can be containerized for easy use without installing Python.

Build Docker Image

```
docker build -t vaultscan-community .
```

Run VaultScan via Docker

```
docker run -it -v ${PWD}:/app vaultscan-community --path /app/tests/dummy_repo --verbose
```

6. PowerShell Launcher Script (scan.ps1)


For Windows users (PowerShell terminal), an optional launcher script is available:

How to Use

```
.\scan.ps1
```

You will be prompted to enter a scanning path interactively.

Example:

 Enter path to scan (leave empty for current directory): D:\simple-java-maven-app-master

7. GitHub Actions Integration

VaultScan can be integrated into GitHub workflows to detect leaked secrets during Pull Requests or Pushes.

Sample `.github/workflows/scan.yml`

name: VaultScan Secrets Detection

on:

push:

branches: [main]

pull_request:

branches: [main]

jobs:

vaultscan:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

```
- uses: actions/setup-python@v4
with:
  python-version: '3.11'
- run: |
  pip install -r requirements.txt
  python -m vaultscan.main --path . --verbose
```

8. Supported Secret Patterns (Community Edition)

- AWS Access Keys
 - AWS Secret Keys
 - GitHub Personal Access Tokens
 - Google Cloud API Keys
 - Azure Keys
 - Stripe Secret Keys
 - Twilio Auth Tokens
 - Private SSH Keys
 - JWT Tokens
 - Database Connection Strings
 - Basic Auth in URLs
 - Slack Tokens
 - Generic API Keys
-

9. Known Limitations (v1.2)

Limitation	Notes
No Git history scanning	Only current working directory is scanned
Regex-based detection only	No AI/static analysis in Community Edition
No alerting or dashboard UI	CLI-only, Pro version includes dashboard

10. GitLab CI/CD Integration

VaultScan can now be used in GitLab pipelines.

To scan your codebase using GitLab CI/CD, add the following to your `.gitlab-ci.yml`:

```
``yaml
stages:
  - scan
vaultscan:
  stage: scan
  image: python:3.11
  before_script:
    - pip install rich
    - git clone https://github.com/pavangajjala/vaultscan-community.git
  script:
    - cd vaultscan-community
    - python -m vaultscan.main --path ../ --verbose
...
```

✔ Secrets detected will be printed directly in your GitLab job logs.

11. Future Roadmap (Private Advanced Version)

- **GitHub/GitLab/Bitbucket API integrations** – Scan entire orgs via secure token-based access.
- **AWS/GCP/Kubernetes secret scanning** – Detect secrets in IaC, configs, and containerized workloads.
- **Automated Slack, Jira, Email alerts** – Notify teams instantly when secrets are detected.
- **Cloud dashboard with scheduling** – View scan results, history, trends, and manage scans centrally.
- **SaaS version with teams, RBAC & policy control** – Role-based access and audit-ready configurations.
- **Obfuscated & base64 secret detection** – Catch secrets split, encoded, or hidden in code.
- **AST/static analysis** – Detect secrets built via code logic (e.g., string joins).
- **Custom ruleset engine** – Define and manage your own secret detection logic.
- **REST API for dashboards & integrations** – Programmatically access scan data for automation.
- **Compliance audit logs** – Track who scanned what, when, and what was found.
- **Severity scoring in CI/CD** – Classify secrets by impact for better enforcement.
- **Visual dashboard** – Web UI with filters, risk views, and export options.
- **Multi-language support (Java, Python, JS, etc.)** – Deep analysis for real-world codebases.
- **IDE plugin support (VS Code, JetBrains)** – Inline detection while coding.

12. About the Author

Developed and maintained by **Pavan Gajjala**,
focused on building privacy-first, security-first solutions for DevOps and Cloud ecosystems.

Disclaimer

VaultScan Community Edition is an open-source prototype intended for learning, personal branding, and early-stage security testing.

VaultScan Pro and VaultScan Enterprise versions are under active development for commercial release.

End of VaultScan Community Edition – User Guide (v1.2)