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) 5
10. GitLab CI/CD Integration 5
12. About the Author 6

1. Introduction

VaultScan Community Edition (v1.2) is a **privacy-first, offline-first CLI tool** that detects secrets and credentials accidentally committed to code.

It empowers **DevOps, Cloud, and Security teams** to 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

💻 Run a Scan

```
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 ignore file: (VaultScan will skip any files/folders matching these patterns.)

```
node_modules/
tests/
*.jpg
*.png
*.md
```

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, an optional **PowerShell 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)

VaultScan Community Edition (v1.2) can detect a wide range of hardcoded secrets using regex-based rules:

- 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	No AI/static analysis in Community Edition
📊 No alerting/dashboard	CLI-only; Pro version includes dashboard

10. GitLab CI/CD Integration

VaultScan can be integrated into **GitLab pipelines** to automatically detect leaked secrets during builds.

📄 **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.

This mirrors your GitHub Actions integration, giving you a **consistent CI/CD story** across platforms.

12. About the Author

VaultScan is developed and maintained by **Pavan Gajjala**, a DevOps Engineer with expertise in **cloud security, CI/CD, and privacy-first tooling**. The project was created to solve real-world problems faced during enterprise cloud migrations.

📢 Disclaimer

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

Commercial editions — **VaultScan Pro** and **VaultScan Enterprise** — are under active development for future release.

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

Thank you for using VaultScan Community Edition.

For feedback, issues, or contributions, visit:

🔗 **GitHub: vaultscan-community**