Cyber Security

Project Docx

SUBMITTED BY

HASSAN ALI

AHSAN NASEER

SYED ALEEM

Software Engineering 22 (Blue)

SUBMITTED TO

Mr. Luqman Shehzad

DEPARTMENT OF IT & COMPUTER SCIENCE

PAK-AUSTRIA FACHHOCHSCHULE INSTITUTE OF APPLIED SCIENCES AND TECHNOLOGY

Repo Scanner-X Documentation

Overview

Repo Scanner-X is a comprehensive GitHub repository scanning tool that combines Trivy vulnerability scanning with AI-powered analysis to identify security risks and provide remediation recommendations. The tool offers a user-friendly Gradio interface for easy interaction.

Source Code Github Repo link:

https://github.com/hassanali167/SecureDeploy

Key Features

- **GitHub Repository Verification**: Validates repository accessibility and permissions
- **Comprehensive Vulnerability Scanning**: Uses Trivy to scan for:
  - Software vulnerabilities
  - Secrets in code
  - Misconfigurations
  - License compliance issues
- **AI-Powered Analysis**: Leverages Groq's LLaMA3-70B model to:
  - Identify critical vulnerabilities
  - Provide remediation steps
  - Highlight potential attack vectors
- **Report Generation**: Creates downloadable reports in text and markdown formats
- **Scan History Tracking**: Maintains statistics of scans per project

---

Installation

Prerequisites

1. Python 3.8+
2. Git

3. **Trivy** (install via package manager or official instructions)

Setup

1. Clone the repository (if applicable):

```
git clone https://github.com/your-username/repo-scanner-x.git
cd repo-scanner-x
```

2. Install Python dependencies:

```
pip install gradio requests python-dotenv
```

Configuration

The tool requires the following configuration:

1. **Groq API Key**: Set in the `GROQ_API_KEY` variable (default provided but may be rate-limited).
2. **GitHub OAuth Token**: Optional for private repositories (passed during runtime).

Usage

Interface Components

1. **Project Name**: Identifier for your scanning project
2. **GitHub Repo URL**: Full URL to the GitHub repository (HTTPS format)
3. **OAuth Token**: Optional GitHub token for private repositories
4. **Verify Button**: Checks repository accessibility
5. **Scan Button**: Initiates the scanning and analysis process
6. **Output Sections**:
    - Repository Status
    - Scan Status
    - Trivy Raw Report
    - AI Analysis
    - Downloadable Reports
    - Project Statistics

Workflow

1. Enter project details and repository URL
2. Optionally provide GitHub token for private repos
3. Verify repository accessibility
4. Run the scan
5. Review results in the interface or download reports

Technical Details

Scanning Process

1. **Repository Verification**:
    - Validates URL format
    - Checks GitHub API accessibility
    - Verifies permissions
2. **Repository Cloning**:
    - Creates temporary directory
    - Clones repository (with token if provided)
    - Extracts repository metadata
3. **Trivy Scanning**:
    - Scans for vulnerabilities, secrets, config issues, and licenses
    - Returns formatted table output
4. **AI Analysis**:
    - Extracts vulnerable files from Trivy output
    - Constructs detailed prompt with repository metadata
    - Sends to Groq API for analysis
    - Formats response for display

File Handling

- All scans create temporary directories that are cleaned up automatically.
- Reports are saved with UUID-based filenames in the working directory.

Customization

Environment Variables

You can modify these constants in the code:

- `GROQ_API_KEY`: Your Groq API key
- `GROQ_ENDPOINT`: API endpoint (default works for most cases)
- `GROQ_MODEL`: AI model to use (default: `"llama3-70b-8192"`)

UI Customization

The Gradio interface can be modified by editing the CSS in the `with gr.Blocks()` section:

- Colors
- Fonts
- Button styles
- Layout structure

Security Considerations

1. **Token Handling**:
    - GitHub tokens are only used for the scan session
    - Tokens are not stored persistently
    - Input field uses password masking
2. **Temporary Files**:
    - All cloned repositories are deleted after scanning
    - Report files remain in working directory
3. **API Security**:
    - Groq API uses HTTPS
    - API key is embedded but can be moved to environment variables

---

Limitations

1. **GitHub Rate Limits**: Without a token, you may hit API rate limits.
2. **Trivy Scope**: Limited to filesystem scanning (doesn't analyze runtime environments).
3. **AI Accuracy**: Recommendations should be verified by security professionals.
4. **Large Repositories**: May take significant time to scan.

Troubleshooting

Common Issues

1. **Repository Verification Fails**:
    - Check URL format.
    - Verify token permissions.
    - Check GitHub status.
2. **Scan Errors**:
    - Ensure Trivy is installed and in PATH.
    - Check network connectivity to GitHub.
    - Verify sufficient disk space for cloning.
3. **AI Analysis Fails**:
    - Check Groq API key.
    - Verify internet connectivity.
    - Check Groq service status.

Future Enhancements

1. **Additional Scanning Tools**: Integrate more security scanners.
2. **Scheduled Scans**: Add periodic scanning capability.
3. **Enhanced Reporting**: PDF/HTML report generation.
4. **Team Collaboration**: Share reports with team members.
5. **Dashboard**: Visual analytics of scan results.

For support or contributions, please contact the project maintainers.