Google Summer of Code

Add GitHub Action including fancy reporting and triage integration

CVE Binary Tool

Python Software Foundation

About Me

Name	Pramurta Sinha
Github	b31ngd3v
Email	contact@b31ngd3v.eu.org
Age	18
Timezone	IST (GMT +5:30)

Education and Background

College	Chandigarh Group of Colleges
University	I. K. Gujral Punjab Technical University
Location	Punjab, India
Program	Bachelor of Technology
Major	Computer Science Engineering
Semester	2nd Sem. (1st Year)

I'm a first year engineering student. I've considerable experience in Python and other scripting languages. I have been contributing to cve-bin-tool since April, 2022 and have learned a lot about the structure and workings of the tool. I would love to work on this project and learn with the community.

Contributions in cve-bin-tool:

Pull Requests:

Contribution	Issue	PR	Status
fix: improve excel macro filter This was my 1st contribution to the project where I updated the macro filters for excel.	#1644	#1647	Closed, Merged
feat: pull updates from mirror with --use-mirror flag This feature will make the update process of cve-bin-tool faster and smoother.	#2577	#2811	Closed, Merged
feat: import and export database as json Helps to export the cve.db as a bunch of JSON files chopped up by years, also load those json files and import CVEs to cve.db	#2577	#2774	Closed, Merged
fix: GAD source version parsing Previously it was parsing the version incorrectly if the version string contained '[', ']', '(' or ')'. That was fixed in this issue.	#2793	#2809	Under review
And 66 other merged pull requests	–	PRs	Closed, Merged

Issues:

Issue	Issue Link	Status
[HTML Report] "Filter by Remarks" not working	#1830	Closed, Resolved
And filed 9 other issues	Issues	Closed, Resolved

Project Information

Organisation

Python Software Foundation

Sub-Organisation

CVE Binary Tool

Abstract

- Create a github action for cve-bin-tool which will produce CVE reports in the GitHub security tab and will be able to split the issues on the basis of triage. It will be smart enough to scan dependency lists of various languages and suggest version upgrades in the form of pull requests. Also it will produce reports in the form of html and pdf by default in the security tab.
- Add the feature which will help the tool running as a GitHub Action to scan SBOM files in the repository and will help to generate an SBOM and keep it up to date through regular scans. The github action will also provide a little badge which will indicate the state of the repository.

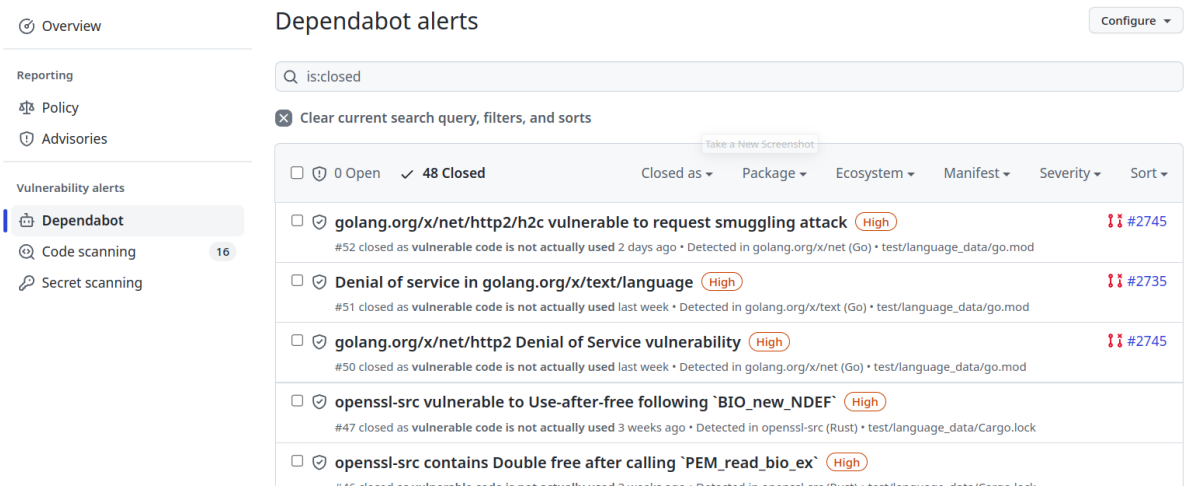
Detailed Description

The CVE Binary Tool helps the users to determine if their system includes any known vulnerabilities or not.
This project idea focuses on making a GitHub Action that will help the developers scan their repositories on a regular basis and provide a detailed report of the scan in multiple formats sorted by triage in the security tab.

Phase 1: Developing a GitHub Action that can scan repositories and display security issues on Security Tab

In Phase 1, I'll be working on the primary feature of the GitHub Action and will try to give this idea a shape. I'll start with a basic GitHub Action that scans the repository and reports the user in the security tab if it finds something. It would work something like dependabot. The action will scan the repository on regular intervals specified by the user. It will also be able to scan language specific requirement files (like requirements.txt for python, package.json for javascript) and when it detects

any vulnerability it will also try to open a Pull Request (if possible) which might fix the vulnerability. The security tab issues will look something like this:



I'll take help of the Code Scanning feature (in GitHub) and build the workflow on top of that. GitHub Code Scanning will provide the API that will be used to alert the developer about the vulnerability in their repositories. Also the GitHub Action will be highly configurable through yml file. It'll automatically open a Pull Request if the version of the package specified in the requirement file is outdated and update the version of the package in the requirement file. It'll also allow the user to ignore small version upgrades through the yml config file.

To summarise, I would carry out the following steps to implement the 1st half of the project:

- Develop basic structure of CVE Binary Tool GitHub Action with alerts on the security tab.
- Develop the feature which will help to scan requirement files of various languages.
- Develop the feature which will auto upgrade the version of the packages by creating automatic Pull Requests.

Phase 2: Adding SBOM support to the GitHub Action

In Phase 2, I'll be mostly working on the more advanced features like SBOM integration, scanning SBOM files with GitHub Actions and generating SBOMs after each scan. The tool will also make sure that the SBOM stays in sync with the original repository. The GitHub Action will also provide a badge indicating the health of the repository which will be updated in each scan. In this phase, I'll also write all the necessary tests for the GitHub Action which will be a key element to make sure that everything works after every update.

To summarise, I would carry out the following steps to implement the 2nd half of the project:

- Add SBOM scanning feature in addition to repository scanning.
- Generate SBOM after each scan and keep the SBOM in sync with the repository.
- Add Badge/Score Tag that will indicate the current state of the repository from the last scan.

Stretch Goals

After finishing tasks mentioned above in phase 1 and phase 2, I would be working on improving the GitHub Action and also side by side I would work on the mirroring feature that is currently being implemented which helps the user to get updates from a mirror instead of NVD and other data sources, so as a result the update process gets a lot faster and also avoids being rate limited. I'm hoping that the mirroring feature would be pretty mature by then and with the help and suggestions of the mentors I would be able to make it better.

Weekly Timeline

Pre GSoC (Upto May 04)

- Make more contributions through issues and features to further my understanding of the codebase.
- Brush up on all the necessary topics and libraries in the proposed project.

Community Bonding (May 04 - May 28)

- Discussing and refining the project idea with the help of the community and the mentors.
- Communicate with other selected applicants about their projects and how we can help each other.
- Start gaining more knowledge about GitHub Actions and Code Scanning.

Phase 1 of 2 (CVE Scanning Action)

Week 1 (May 29 - June 05)

- Start giving the idea a real shape, create a basic Action which will be able to scan repositories.

Week 2 (June 06 - June 12)

- With the help of Code Scanning, customise the security tab.
- Provide reports in multiple formats (eg. HTML, PDF) in the security tab.

Week 3 (June 13 - June 19)

- Split issues on the basis of triage and prioritise important issues.
- Auto update packages with automatic Pull Requests.

Week 4 (June 20 - June 26)

- If a vulnerability is found in a product, try to suggest a solution.
- If a version of a binary has a vulnerability, suggest a version which is not vulnerable.

Week 5 (June 27 - July 03)

- Configuration option which will allow to ignore minor version upgrades.
- Start looking for possible optimizations.

Week 6 (July 04 - July 10)

- Work on fixing all the new and reported issues.
- Finish up phase 1 of the project with documentation.

Phase 2 of 2 (SBOM Integration)

Week 7 (July 11 - July 17)

- Phase 1 submission and evaluation.
- Discuss and start implementing ideas for SBOM Integration.

Week 8 (July 18 - July 24)

- Add SBOM scanning feature in addition to repository scanning.

Week 9 (July 25 - July 31)

- Generate SBOM after each scan.
- Keep the SBOM in sync with the repository.
- Add a Badge that will indicate the current state of the repository.

Week 10 (August 01 - August 07)

- Make a video tutorial which will help the user to setup the GitHub Action.
- Make a blog post as a walkthrough of the whole process.

Week 11 (August 08 - August 14)

- Work on fixing all the new and reported issues.
- Add new test cases to suit the changes made.

Week 12 (August 15 - September 21)

- Keep fixing new issues and adding new test cases.
- Wrap things up for phase 2 of the project with documentation.

Final Week (September 21 - September 28)

- Prepare a final summary and organise the work into a presentable form.
- Submission of the work for final mentor evaluation.

Other Commitments

The GSoC timeline aligns with my availability, therefore providing me the opportunity to work throughout the summer with focus and absolute conviction.
I do not have any exams during the GSoC contribution period.

Are you applying for other projects in GSoC?

No, I am only applying for this project.

Further Contributions

CVE Binary tool is a security-based python project that corresponds with my interests in the field, therefore I will be pleased to be a part of the community even after the GSoC to continue contributing and learning.