CVE Binary Tool

Introducing support for EPSS

About Me:

Name: Sukhveer Singh
Github: Rexbeast2
Email:
University: Indian Institute of Information Technology, Gwalior
Program: Bachelor of Technology in Computer Science and Engineering
Year: 3rd year
Location: India
Timezone: Indian Standard Time (GMT +5:30)

In order to undertake this project, I possess the essential competencies in Python, databases, and utilizing APIs. I have been actively contributing to the development of CVE-BIN-TOOL since the beginning of this year and have gained a comprehensive understanding of its structure and functionality.
Since the GSoC timeline aligns with my university's summer break and my classes are asynchronous, I will have uninterrupted time to focus solely on the project.

Collaborating with the community and contributing to this open-source project would be a valuable learning experience, and I am enthusiastic about the opportunity. I look forward to working with the community to improve the accuracy and reliability of CVE-BIN-TOOL.

Code Contribution:

Pull Requests

After carefully reviewing the codebase multiple times, I am confident in my understanding of the tool's functionality. Therefore, I feel comfortable working on the project with minimal difficulties or challenges.

Project Information:

Sub-org Name: CVE-BIN-TOOL

Project Abstract:

The CVE-BIN-TOOL currently relies solely on CVSS as a metric to score vulnerability severity. However, high CVSS scores do not necessarily equate to exploitability. To improve the accuracy of the tool's output, this project aims to integrate the Exploit Prediction Scoring System (EPSS) as an additional metric.

The project will involve updating the database schema to accommodate EPSS data, adding EPSS data to the database through download in CSV format and parsing it for each CVE, updating database queries to retrieve EPSS data, and updating output reports to include EPSS data. The project will also include adding extra CLI commands to filter EPSS scores and updating documentation and tests to reflect the new changes.

Overall, integrating EPSS will enhance the precision and reliability of the CVE-BIN-TOOL, allowing users to understand the severity of vulnerabilities better and prioritize their remediation efforts.

This is an overview of the tasks for this project:

- Update database schema
- Add EPSS to the database
- Update output reports to include EPSS data
- Add extra CLI commands to filter EPSS scores
- Update documentation and tests

Detailed Description:

Update database schema :

The current database schema needs to be updated to accommodate EPSS. The updated schema will include CVEs with their ID, severity, description, score, CVSS_version, CVSS_vector, last_modified, and EPSS score.
Regarding supporting multiple metrics, I was thinking about having a uniform data distribution (a simple SQL table), maybe something called "metrics," where each column would represent a metric and new columns would be able to be added based on as per need, something like :

CVE_Number	metric_1	metric_2	metric_3	etc	etc..
cve_ID	score_m1	score_m2	score_m3	..	..

In terms of how to implement this, all the while trying to keep it modular, robust, and easy to use, I've taken inspiration from the currently implemented "framework" of checkers, data sources, and parsers. Since all metrics simply have different sources but are other than entirely similar to one another (i.e., simple columns in a table). Creating a structure like this would make it highly convenient to add new metrics, such as just adding a new file which will then introduce a new column metric in the said "metric" table.

Adding EPSS data to the database:

After updating the database schema, we will need to add EPSS data to the new database. I propose to download the data in CSV format, parse it for every CVE, and combine it with CVE data downloaded from other data sources.(datasource) The data will then be stored in the new database according to the updated schema. For every new CVE, we will use the EPSS API to retrieve the EPSS score, percentile, and date and store the data in the database. Regarding the EPSS data record, I propose adding a new column that includes the most recent timespan in which the data was updated. This column would automatically update every time the EPSS data for that vulnerability is modified. Additionally, we can create and store a copy of the EPSS data for 30 days to track changes in the data over that time period. After the 30-day duration has passed, we will discard that data and replace it with a new copy of the updated EPSS data.

Update output reports to include EPSS data :

The current output reports need to be updated to include the EPSS score for every CVE found in the product. We will update the output reports to include the new field and ensure that it is displayed correctly. I propose creating a new separate summary with a metrics table would be a great option that would contain all the metrics of the CVE's present so it will be easy to compare and select. These output reports need to be improved for every type of format we provide CSV, JSON, console, HTML, and pdf.

Add extra CLI commands to filter on EPSS scores :

As the EPSS score is a probability in the range of [0,1], I propose adding a command-line option that allows the user to enter the minimum probability they want to have in the CVEs found. The CLI command will filter out the CVEs based on the EPSS score provided by the user to the CVE-BIN-TOOL though rather than getting the user input in decimal, I propose to convert it to the [0,1] range to [0,100]. I suggest setting the EPSS default threshold at 30 percent (.3), which would encompass approximately 7,500 CVEs. However, users should have the option to configure this default threshold to their preference. This can be easily accomplished by adding a flag to the EPSS filter to allow for setting the default threshold. Updating config files(TOML and YAML files) to include EPSS fields, including config files taken as input and config files generated.

Update documentation and tests :

After adding EPSS to the database, updating the database schema, and adding the command-line filter, we will need to add test cases to test the synchronous working of the application. We will also update the documentation to explain the new features and uses of the tool to new users. The documentation will include being for updating the database schema, adding EPSS data to the database, new updated output reports, and step-by-step instructions for the user to use the new command-line option for EPSS.

Weekly Timeline:

Pre GSoC:

- To enhance my comprehension of the codebase, I will contribute to the project by addressing issues and challenges.
- Specifically, I will focus on resolving various database-related problems to gain a deeper understanding of the codebase.
- Additionally, I will engage in constructive dialogues with mentors to explore opportunities for streamlining and optimizing the project.

Community Bonding Period (4 May - 28 May):

- Enhance my community involvement by engaging with its members.
- Compile a list of issues that require attention and prioritize them accordingly.
- Maintain a consistent level of engagement to address and fix bugs or issues.
- Resolve any ambiguities or misconceptions related to my project, which will facilitate coding during the implementation phase.

Week 1 (29 May - 4 June):

- Updating current database Schema to include EPSS.

Week 2 (5 June - 11 June):

- Making database queries for fetching EPSS data.
- Fetching, parsing, and storing the EPSS data in the database.

Week 3 (12 June - 18 June):

- Updating database queries and matching/fixing missing data.
- Fixing bugs and flaws in code.

Week 4 (19 June - 25 June):

- Improving and refining code.
- Improving EPSS Integration with new database scheme.

Week 5 (26 June - 2 July):

- Updating the Output Report to Accommodate the EPSS section.
- Updating Output Report for every format(HTML, CSV, JSON, pdf, and console).

Week 6 (3 July - 9 July):

- Updating test cases to accommodate the EPSS section.
- Working on feedback/review.

Week 7 (10 July - 16 July):

- Updating the CLI command to include EPSS filter.
- Working on Output format test cases.

Week 8 - 9 (17 July - 30 July):

- Adding test cases for CLI command for EPSS.
- Working on feedback/review.

Week 10 (31 July - 6 August):

- Refactor and optimize code and add tests.
- Fixing bugs.
- Updating config files.

Week 11 (7 August - 13 August):

- Adding the documentation and remaining test cases.

Week 12 (14 August - 20 August):

- Keeping this week as a buffer.
- Fix bugs if any present
- Updating blog about the work done so far.

Final Evaluation (21 August - 28 August)

- Finalise code and documents for final submission.
- Work on feedback/review, if any.
- Wrap up everything

Stretched Goal

If the project is completed ahead of schedule,

- I plan to address a few issues related to the database,
    - Implementing a command for reloading the database.
        - The reload command is designed to quickly update the database by pulling data from the locally stored cache on the disk rather than querying the original data source. This can be accomplished by calling one of several related functions in a systematic manner. This command is similar initialize command feature for which I have submitted a PR.

    - Adding a time limit to the NVD API 2.0 code
        - One of the NVD API calls doesn't have an overall timeout, which has caused some issues because the default timeout of 5 minutes is insufficient for certain calls. To address this, set a maximum timeout of 15 minutes (900 seconds) for this call. However, to provide flexibility, adding an environment variable called NVD_TIMEOUT that can be easily adjusted based on specific needs.
        - It's important to note that 15 minutes is roughly double the time it usually takes for the NVD to download all of the data using the 2.0 API. We believe this increased timeout will provide ample time for the API to complete its tasks without causing unnecessary delays or timeouts.

Post GSoC:

continue contributing to the project and work on bugs/issues.

Other commitments:

In mid-July, I have some minor university assessments, but other than that, my schedule is free. However, these assessments may affect my ability to allocate enough time to the ongoing project. To ensure that I deliver the project on time, I have devised a plan to dedicate extra time during the other weeks. This approach will help me balance my commitments and ensure that the project timeline is not jeopardized.

CVE-BIN-TOOL is the only organization I am applying for.