CVE Binary Tool:

Checker creation Helper Script
Recommending Safe package versions

Project Information

Abstract:

- ➔ To create Helper Scripts which would simplify the process of checker creation.
- ➔ To recommend users the safe package versions for corresponding vulnerable packages which are found while scanning

Detailed Description:

Checker Creation Help Script

cve-bin-tool looks at strings found in binary files to see if they match certain vulnerable versions of vulnerable components such as libpng, libxml2 ,...

Since, cve-bin-tool relies on contributors to build checkers for each package. This in turn requires them to know various file manipulation methods, which are easier to use on linux than on windows. To simplify this process, I propose to create a checker creation helper script, which would take **.rpm**, **.deb**, **.tar.gz** or other files that represent packaged versions of the software to be detected, including the **product name** and **version number** that we expect to find in each as a set of inputs.

The script would automate the process mentioned in the checkers/readme.md and output the following results:

- `CONTAINS_PATTERNS` - list of commonly found human-readable strings in the binary of the product
- `FILENAME_PATTERNS` - list of different filename for the product
- `VERSION_PATTERNS` - list of version patterns found in binary of the product.
- `VENDOR_PRODUCT` - list of vendor-product pairs for the product as found in NVD.

Motivation to create this

This would help new contributors to understand what "common" filenames, version-patterns look like. This would also enable windows users to contribute more easily.

Recommending Safe Package Versions

Currently, cve-bin-tool outputs the vulnerable package name, version and vendor-product name. An upgrade to this would be to also provide the user with the list of all safe/fixed packages for the corresponding vulnerable packages found while scanning.

This could be done by taking a vulnerable package (which was found in the scan) and web-scraping for the latest vulnerable version and then comparing it with all it's released versions and then giving the safe version as output. If there is no safe version available for a package, then we could output it's latest version or the version with a low CVSS score.

For packages with multiple dependencies (initially we could start with dependencies < 3), we could do something similar as above by storing these in python dictionaries.

Example: Suppose there are 3 packages A, B, C which are dependent on each other. If A-2.3.0 depends on B-4.0.0 and B-4.0.0 depends on C-1.2.3, but A-2.3.0 depends on C-1.3.0. Now if B-4.0.0 is compatible with C-1.3.0, we print the output of recommended packages to install as A-2.3.0, B-4.0.0, C-1.3.0 (not C-1.2.3).

Doing this for files with > 3 dependencies will be harder, but an exciting challenge :)

Motivation to create this

This would allow developers to quickly be able to upgrade to latest safe packages and also provide information about multiple dependencies.

Weekly Timeline:

Community bonding (May 17, 2021 - June 7, 2021)

- Understanding the code base
- Learning things related to project
- Understanding what weak signatures/”signature needs work” means and how to encounter them
- Create a list of checkers to add during this period
- Create an overall outline

Week 1 (June 7, 2021 - June 13, 2021)

- Start to create the helper script
- Automating the files downloading and extracting process and testing it

Week 2 (June 14, 2021 - June 20, 2021)

- Implementing the regex finding process and testing it

Week 3 (June 21, 2021 - June 27, 2021)

- Polishing the code and Giving a proper output to the created script
- Testing it on various existing checkers for validation of the script

Week 4 (June 28, 2021 - July 4, 2021)

- Continue to test the helper script
- Getting feedback from contributors and implementing any required changes

Week 5 (July 5, 2021 - July 11, 2021)

- Adding various checkers with the help of the script from the checker list initially created

Week 6 (July 12, 2021 - July 18, 2021)

- Start to work on “Recommending safe package versions”
- Implementing the Web Scraping process and finding the various version strings from multiple databases and testing it

Week 7 (July 19, 2021 - July 25, 2021)

- Working on the code that would compare the safe and vulnerable packages
- Testing it on custom and various packages

Week 8 (July 26, 2021 - August 1, 2021)

- Giving a proper output for the safe packages versions
- Testing and Getting feedback
- Starting to work on the packages with multiple dependencies

Week 9 (August 2, 2021 - August 8, 2021)

- Continuing to work on the code for multiple dependencies
- Testing and Getting feedback

Week 10 (August 9, 2021 - August 15, 2021)

- Polishing the code
- Preparing for submission of the project

Final Week

- Submitting the project

Post GSoC

An extended goal for the “Recommending Safe Packages” would be to automate the process of upgrading to those safe packages via the click of a button. I plan on contributing to this upgrade and learn various new things in the process :)

Other Commitments:

During Weekdays

On weekdays, I would be having my college classes.So, depending on college timings, I would be working in the following hours:

> 1700 - 2200 IST : I would be doing most of my work during these hours (with 50-60 min break (around 1930 - 2030 IST for dinner and other activities))

I would be able to contribute more during my half-day classes (the schedules for my classes are yet to be announced).

During Weekends

I would be able to give more time during weekends. Depending on the work from college, I would be able to work in the following hours:

> 1000 - 1330 IST : I would be working during these hours.
>
> 1600 - 1930 IST : Depending upon the amount of work from college, this time may vary accordingly.

As of any other commitments, there would be my 1st Semester exams during 10 April to 15 May (which luckily do not collide with anything), I wouldn't be able to contribute more during this period. During my 2nd Semester internal exams (schedules are yet to be announced), I would be less active during this period (probably 3 days).

Code Contributions:

https://github.com/intel/cve-bin-tool/pulls?q=is%3Apr+author%3Apeb-peb

About Me:

Name	: Harsh
Github ID	: peb-peb
College	: Bangalore Institute of Technology, Bangalore
Program	: B.E. in Computer Science and Engineering
Year	: 1st Year
Expected Graduation Date	: 2024
Time Zone	: Indian Standard Time, UTC +5:30