Common Vulnerabilities Finder Tool (CVEInspector)

Dr. S Venkatesan and Manpreet Singh
Indian Institute of Information Technology Allahabad,
Department of Information Technology, Prayagraj,INDIA-211015
venkat@iiita.ac.in IIT2020025@iiita.ac.in

1 . Introduction

Common vulnerabilities refer to the known weaknesses or flaws in computer systems, networks, software, or applications that can be exploited by attackers. These vulnerabilities may arise due to programming errors, design flaws, misconfigurations, or outdated software components. Exploiting these vulnerabilities can lead to unauthorized access, data breaches, system compromise, and other security incidents.

Knowing common vulnerabilities of a system is vital for effective risk management, prioritizing mitigation efforts, maintaining compliance, responding to security incidents, and proactively defending against potential threats. It allows organizations to make informed decisions, allocate resources effectively, and implement security measures that address the specific vulnerabilities their systems may be exposed to.

2 . Different available CVE finders

- **National Vulnerability Database (NVD)**: The NVD, maintained by the National Institute of Standards and Technology (NIST) in the United States, is a comprehensive database of CVEs. It provides detailed information about vulnerabilities, including their descriptions, impact, severity ratings, and references to applicable patches or fixes.
- **Common Vulnerabilities and Exposures (CVE) website**: The official CVE website, operated by the MITRE Corporation, offers a search interface to access the CVE database. It allows users to search for specific CVEs, browse through vulnerability entries, and obtain additional details.
- **Common Platform Enumeration (CPE)**: The CPE is a standardized naming scheme for software applications, operating systems, and hardware devices. It is linked to CVE entries, enabling users to search for CVEs based on specific products or vendors. The NVD and CVE websites provide CPE-based search functionality.
- **Vulners**: Vulners is a popular vulnerability database and search engine that aggregates information from various sources, including NVD, MITRE, Exploit Database, and more. It provides a unified interface to search for vulnerabilities, exploits, and related security content.
- **ExploitDB-py** :It is a powerful Python library that provides developers and security researchers with seamless access to the renowned Exploit Database (ExploitDB). With its extensive collection of exploits, payloads, and vulnerabilities, ExploitDB-py simplifies

the process of identifying weaknesses in software systems. Users can search, retrieve, and utilize exploit code effortlessly, staying up to date with the latest security vulnerabilities and fixes. The library offers a user-friendly interface for easy integration into Python projects.

3 . CVEInspector(New Tool)

As the name suggests, the "CVEinspector" tool is a vulnerability scanner that aims to identify potential vulnerabilities based on keywords extracted from an input file. By leveraging regular expressions, it analyzes the provided file for patterns matching software versions and extracts relevant keywords that might indicate the presence of vulnerabilities.

To ensure the latest information is available, CVEinspector pulls the most recent changes from the CVE (Common Vulnerabilities and Exposures) database, stored in a local repository. This ensures that the tool has access to the most up-to-date vulnerability data.

Once armed with the keywords, CVEinspector diligently searches the vulnerabilities database for files associated with these keywords. By cross-referencing the keywords against the database, it aims to pinpoint potential vulnerabilities that may pose risks to the systems or software being analyzed.

The tool generates an output file, "output.txt," listing the identified keywords and the corresponding **CVE numbers from the NVD database** that potentially contain relevant vulnerability information. This report assists users in further investigation and mitigation efforts to safeguard their systems from potential threats.

3.1 Keywords Extraction of CVEInspector

3.1.1 Input File: CVE Inspector takes an input file as its primary source of data. This file contains information, such as software versions or version-related details, that could potentially indicate the presence of vulnerabilities.

3.1.2 Pattern Matching: CVE Inspector uses regular expressions to search for patterns in the input file. These two patterns are used:

a. Pattern 1: The first pattern, "pat1," is designed to capture words or phrases of the format "<software> version <version_number>". It uses the regular expression (\w+(?:\.\w+)*)\s(?:version|v)\s(\w+(?:\.\w+)*) to match and extract these software versions and related keywords.

b. Pattern 2: The second pattern, "pat2," captures words or phrases of the format "<software> <version_number>". It employs the regular expression ([a-zA-Z]+(?:\.[a-zA-Z]+)*)\s(\d+(?:\.\d+)*) to identify and extract these software versions and relevant keywords.

b. Custom Patterns : To add or remove patterns, users can edit the file "regular_expression.txt". These patterns should be of the python regular expression format

Regular_expression.txt

```
(\w+(?:\.\w+)*)\s(?:version|v)\s(\w+(?:\.\w+)*)
([a-zA-Z]+(?:\.[a-zA-Z]+)*)\s(\d+(?:\.\d+)*)
(\w+(?:\.\w+)*)\s(?:version|v)(\d+(?:\.\d+)*)
```

3.1.3 Ignoring Irrelevant Matches: After extracting potential keywords using the patterns, CVE Inspector checks against the keywords present in the "ignore" file. If a keyword is found in the ignore list, it is excluded from further processing, ensuring that only relevant keywords are considered. Users can modify the "ignore.txt" file to add or remove keywords from the ignore keywords list.

3.2 Database Used in CVEInspector

CVEInspector utilizes the National Vulnerability Database (NVD) as its database for vulnerability information. The NVD, maintained by the National Institute of Standards and Technology (NIST) in the United States, is a comprehensive and authoritative source of CVE (Common Vulnerabilities and Exposures) data.
To ensure that CVEInspector has access to up-to-date vulnerability information, it incorporates the functionality of updating the CVE database monthly (after every 30 days).

3.3 CVE numbers Generation By CVEInspector

1. CVEInspector leverages the extracted keywords to search for corresponding vulnerabilities in the NVD database.
2. By matching the keywords against the vast collection of CVE entries, CVEInspector identifies potential vulnerabilities.
3. The tool utilizes an efficient cross-referencing mechanism to map the keywords to their associated CVE numbers.
4. CVEInspector generates a list of CVE numbers that are potentially linked to the identified vulnerabilities.
5. Each CVE number represents a unique vulnerability entry in the NVD database, providing detailed information about the associated security issue.
6. The generated list of CVE numbers serves as a valuable reference for further investigation and mitigation efforts.
7. With CVE numbers in hand, users can access comprehensive vulnerability information, including impact severity, affected software versions, and available patches or mitigations.

3.4 Results of CVEInspector

This is the output file (CVE numbers) generated by CVEInspetor on scanning the boot logs of linux based network router.

Output.txt

```
Possible vulnerabilities of Linux 5.0.0
  CVE-2018-6705.json
  CVE-2018-6704.json
Possible vulnerabilities of gcc 8.2.0
Possible vulnerabilities of hci 0x100
Possible vulnerabilities of OpenSSL 3.0
  CVE-2022-2097.json
  CVE-2022-2068.json
  CVE-2022-2274.json
  CVE-2022-2906.json
  CVE-2022-1343.json
  CVE-2022-1473.json
  CVE-2022-1434.json
  CVE-2022-1292.json
  CVE-2022-0778.json
  CVE-2022-3358.json
  CVE-2022-3602.json
  CVE-2022-4450.json
  CVE-2021-4160.json
  CVE-2021-4044.json
Possible vulnerabilities of GENET 5.0
Possible vulnerabilities of Broadcom 2835
Possible vulnerabilities of Linux 5.10.63
Possible vulnerabilities of iproc 3
Possible vulnerabilities of tuple 0
Possible vulnerabilities of usb 1
Possible vulnerabilities of 0300 1.3
Possible vulnerabilities of NETLINK 0.30
Possible vulnerabilities of ctnetlink 0.93
Possible vulnerabilities of Ebtables 2.0
Possible vulnerabilities of Support 1.8
  CVE-2007-6539.json
  CVE-2009-4434.json
  CVE-2009-4433.json
  CVE-2006-4884.json
Possible vulnerabilities of nf_conntrack_rtsp 0.6.21
Possible vulnerabilities of nf_nat_rtsp 0.6.21
Possible vulnerabilities of support 0.3
Possible vulnerabilities of class 2.0
```

4 . Comparative Analysis

The main differentiating features between our tool, CVEInspector, and other available tools are:

1. **Automatic Keyword Extraction:** CVEInspector has the capability to automatically extract keywords from the input file, eliminating the need for manual keyword input. This streamlines the vulnerability scanning process and saves time.
2. **Offline CVE Data:** CVEInspector searches for vulnerabilities using the recent CVE data offline. This approach ensures that users have access to up-to-date vulnerability information without relying on online APIs.
3. **Faster Scanning:** By leveraging offline CVE data, CVEInspector offers faster vulnerability scanning compared to tools that rely on online APIs. This efficiency enables users to perform scans more quickly and efficiently.
4. **API Limitations:** Many online API-based tools, such as the CVE website by MITRE and Vulners, have limitations, such as a limited number of requests per hour. CVEInspector eliminates these limitations by using offline data, allowing users to perform scans without being restricted by API usage restrictions.

Comparison Table

Feature	CVEInspector	CVE website by MITRE	Vulners	ExploitDB-py
Automatic Keyword Extraction	Yes	No	No	No
Offline	Yes	No	No	Yes
Faster Scanning of Multiple Keywords	Yes	No	No	No
API Limitations	None	Yes	Yes	None

5 . Conclusion

CVEInspector distinctive features provide significant advantages in vulnerability scanning and analysis. By automatically extracting keywords from the input file and searching through an offline but up-to-date CVE database, CVE Inspector streamlines the process and saves valuable time. The tool's offline functionality allows for faster scanning compared to relying on online APIs, which often have limitations and restrictions. CVE Inspector ensures users can perform scans efficiently, without being constrained by API usage limits.

6 . References

https://github.com/CVEDB/cve-monitor

https://nvd.nist.gov/vuln/vulnerability-detail-pages

https://cve.mitre.org/cve/search_cve_list.html