Combining sources from CVE and CNNVD: Data analysis in information security vulnerabilities

Ruiying Jin* and Jing Nan²

1School of Economics and Management, Beijing Jiaotong University, Beijing, 100000, China
2School of Computer, Beijing Jiaotong University, Beijing, 100000, China

*Corresponding author’s e-mail:18711052@bjtu.edu.cn (ORCID: 0000-0002-8623-4718)

Abstract. The information technology level of China continues to improve, and various industries are digitalized in different degrees. However, the development of the network has correspondingly brought the trouble of loopholes, which have caused mass economic losses to many enterprises. In the era of big data, it is necessary to understand the characteristics of the ever-emerging vulnerabilities. In this study, we used descriptive statistics and programming techniques to analyze more than 140,000 vulnerability records from CVE and CNNVD. We visualize the development trend of vulnerabilities through descriptive statistics and summarized the characteristics of vulnerabilities in products from manufacturers of different scales. The result shows that large companies usually face more and severer loopholes. Then, multivariate regression is conducted to explore the relationship between vulnerability type, threat type and hazard level. K-means clustering method is employed to select and extract features based on the description paragraphs of vulnerabilities. Finally, we categorize the loopholes according to different patterns, so as to find a breakthrough point for a quicker solution to new vulnerabilities with the existing classified database. The combined sources are proved to be useful for clustering analysis.

1. Introduction
Nowadays, an increasing number of companies face the problem of vulnerability, and many companies suffer a great loss. In 2010, Google announced that it would consider withdrawing from the Chinese market. The announcement said that the important reason for this decision was that Google was attacked by hackers [1]. On December 21, 2011, CSDN, a well-known Chinese domestic programmer website, was attacked by hackers. A large number of user databases were published on the Internet, and more than 6 million plaintext registered E-mails were forced to expose to the public. December 29, 2011 afternoon, following user data leakage of the CSDN and Tianya community, the Internet industry is in jittery, as there are also loopholes and user information leakages in the most important business areas of user data. These all show the significance of network security to the company.

1.1. Cause of the vulnerabilities
There are many causes of vulnerability, which are generally divided into the following categories:
1.1.1. **Protocol vulnerabilities in the network.** Take the most widely used TCP/IP protocol group as an example. It is designed in a trusted environment. It only considers the network interconnection and openness, but not too much security. As a result, the TCP/IP protocol group itself is not secure in the application, which leads to the security and vulnerability of a series of network services based on TCP/IP protocol.

1.1.2. **Vulnerability of application software system.** Due to the design defects of any software, there are more or fewer loopholes, which can cause the vulnerability of the system itself. Usually, there are two kinds of vulnerabilities: one is the security vulnerability caused by the design defects of the operating system itself, which affects the application running on the system; the other is the design vulnerability of the application itself.

1.1.3. **Vulnerability caused by improper configuration.** Due to the incomplete security policy settings, the system sometimes runs when the security policy does not work, and it is difficult for managers to find out until there are problems in the system.

1.1.4. **Management vulnerability.** The lack of management system and the mistakes of management personnel will cause vulnerability [2].

1.2. **CNNVD**

CNNVD is China's National Information Security Vulnerability Database [3], full name as "China National Vulnerability Database of Information Security", hereinafter referred to as "CNNVD", China's Information Security evaluation center is to earnestly implement the functions of Vulnerability collection and risk assessment, and responsible for the construction of National Information Security Vulnerability Database operations, providing the basic services to Information Security in China.

1.3. **CVE**

CVE is a vulnerability database with its full name "Common Vulnerabilities and Exposures" to function the universal vulnerability disclosure [4]. The CVE table is like a dictionary, as has been widely accepted by information security holes or have exposed weaknesses are a Common name. Using a common name helps users share data in separate vulnerability databases and vulnerability assessment tools. If the vulnerability is identified in a vulnerability report and has a CVE name, the user can quickly find information about the solution in any other CVE-compatible database to resolve security issues.

1.4. **Compare and contrast CNNVD and CVE databases**

CNNVD is larger and more comprehensive, and many CVE does not cover several domestic manufacturers in China. For example, many Chinese router manufacturers do not have contact information on CVE, which means the loopholes they encountered are very less likely to be included in CVE. Most of the medium and small business would choose to report to CNNVD oftentimes. China has developed rapidly in the field of information security, and the number of enterprises engaged in network, information and technology has been surging. The influence of them cannot be overlooked. Therefore, a more comprehensive conclusion can be obtained by combining the two databases for analysis.

2. **Data sources and sample selection**

The research team fetches over 140,000 records of data from the CVE and CNNVD databases (including both Chinese and English languages) by web mining technology. The repeated and redundant data are not included in the data file being analyzed, therefore obtaining the final data with 149,310 records.
3. Empirical analysis

Figure 1. Stack Bar Chart of Hazard Level Count for Different Manufacturers

This chart is accomplished by data intelligence software Tableau, the vulnerability count of each vendor with each hazard level obtained from Tableau. After sifting through the number of vulnerabilities, we found that Microsoft, Apple, Oracle, Google, Cisco and other large companies had the largest number of vulnerabilities. Most of their vulnerabilities are medium-risk. As the number of vulnerabilities decreases, the vulnerability levels for smaller companies tend to be more uniform. This is to say that small companies are basically accompanied by low-risk, medium-risk, high-risk or super-risk vulnerabilities.

Figure 2. The speed of different vendors to resolve vulnerabilities

This chart calculates the average number of days each vendor solves vulnerabilities. After analyzing the tableau chart, we found that the average speed of vulnerability resolution varies from vendor to vendor. About a third of the vendors resolved the problem within 100 days, and those vendors were
generally smaller and didn't have a lot of bugs. About a third of the vendors have solved the problem within 100 to 500 days, and those vendors have a lot of large companies with a lot of bugs. A third of the vendors solved the problem in more than 500 days. Some of these vendors were large and had a large number of bugs, while others had a small number of bugs but were slow to fix.

Figure 3. Number of vulnerabilities in different years

This graph shows the trend of the number of vulnerabilities with different levels of hazard over time. Due to the data source collection is up to July 2020, the data for 2020 is not a full-year number of records, contributing to a decrease on the graph. It can be seen that the number of vulnerabilities fluctuated and increased. Through Tableau analysis, after increasing the number of vulnerabilities with different hazard levels over time, we found that the number of medium-risk and high-risk vulnerabilities increased, while the number of low-risk and super-risk vulnerabilities changed steadily.

Figure 4. Number of different types of vulnerabilities
This visualization shows the number of vulnerabilities for different threat types. From the visualization, we find that the number of distant threat vulnerabilities far exceeds the number of other threat types. The number of vulnerabilities of a specific network environment and distant or local threat type is relatively small, while the number of vulnerabilities of local threat type is relatively larger.

4. Multivariate regression

After analyzing the attributes of the data, we want to explore the relationship between vulnerability type, threat type and hazard level. We treat it as a multivariate regression with virtual variables. First, to facilitate the operation, clear the rows that are missing any of the three variables. The code in python is: data = Time1.dropna(axis='index', how='any', subset = ['Hazard level', 'Vulnerability Type', 'Threat Type']). Because there are four levels of hazard, in order to facilitate regression, we convert them into dummy variables from low to high. The types of the two independent variables after data cleansing and grouping were counted as follows:

| Threat Types                                      | Number Counted |
|--------------------------------------------------|----------------|
| SQL injection                                   | 14913          |
| Insufficient logging and monitoring              | 12843          |
| Failed access control                           | 13366          |
| Failed authentication and session management     | 14844          |
| Sensitive information leakage                    | 10850          |
| Buffer overflow                                 | 19131          |
| Design error                                    | 2160           |
| Resource management error                        | 5286           |
| Cross-site scripting vulnerability              | 19094          |
| Directory traversal attack                       | 4061           |

Table 1. Grouped statistics of vulnerability types after data cleaning

| Treat Types                      | Amount |
|----------------------------------|--------|
| Local                            | 19919  |
| Specific Network Environment     | 2165   |
| Distant                          | 107154 |
| Distant or Local                 | 1122   |

Note: the category of “Distant and Local” are both added to “Distant” and “Local”.

| OLS Regression Result                  |
|----------------------------------------|
| Dep. Variable: Hazard Level            | R-squared: 0.266 |
| Model: OLS                             | Adj. R-squared: 0.266 |
| Method: Least Squares                  | F-statistic: 3518 |

R square is more than 0.1, as medium correlated. The F value is far larger than 3, which means it is a significant test. So, we conclude that there exists a weak positive correlation between the three variables.

5. Clustering

5.1. Word cutting by Jieba

We use Jieba package tool, a tool to execute Chinese word cut, to cluster the column of vulnerability description, count word frequency by word segmentation, and import the final result into a .txt document. In a python environment, developers just need to import Jieba and then write the corresponding codes.
Table 4. Result one (translated)

| Keywords              | Amount  |
|-----------------------|---------|
| Version               | 197104  |
| Loophole              | 164695  |
| Attacker              | 86943   |
| Use                   | 76407   |
| Company               | 54863   |
| America               | 50214   |
| Distant               | 49780   |
| Windows               | 45981   |
| Execution             | 38392   |
| Product               | 38113   |
| Arbitrary             | 34372   |
| Security loophole     | 31566   |
| Document              | 31104   |
| Affect                | 30376   |
| Code                  | 27376   |
| Platform              | 25055   |
| Microsoft             | 23760   |
| Server                | 23570   |
| Oracle                | 23546   |
| Components            | 23356   |
| By means of           | 23355   |
| User                  | 23323   |
| Web                   | 22920   |
| Operation system      | 22722   |

In the result, we can see that the most common words are “version” and “vulnerability”. On one hand, it shows that the vulnerability version is updated quickly and the number of versions is large. On the other hand, it reflects the complexity and difficulty of solving vulnerability. The words “attacker” and “company” indicate the source of the vulnerability and the target or maintainer of the vulnerability. The word "United States" is also very frequent, which shows that the United States is still far ahead in the release of vulnerability. “Windows” and “Microsoft” show that operating system vulnerabilities account for a large proportion of all vulnerabilities. The word "Web" indicates that there is a transition from the server to the client in the vulnerability attack center, the system layer and the network layer are gradually expanding to the application layer, and the harm caused by web application security vulnerabilities is prominent.

5.2. Data processing

The data in each txt document for word segmentation, using the Jieba library and stopwords documents for word segmentation, with space splicing, the results are saved in the "word segmentation results 2" folder, named "vulnerability numbers".

5.3. Calculation of Tf-Idf

K-means clustering at the same time, initial clustering was set at 20 centers core package imported are as follows:

```python
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
```

5.4. Results

Start Kmeans:

```python
KMeans(n_clusters=20)
```
inertia = 2454.4594014767536

We find similar patterns through existing data, and we can quickly find the breakthrough and solution when new vulnerabilities appear. A quicker solution will be found with the existing categories.

Figure 5. Output Visualization

6. Conclusion
The overall number of vulnerabilities still keeps increasing, and the number reached a new high in 2019. In the field of information security, both large and small companies are facing huge challenges. Although the technology is more advanced, Amazon and Google's products are at higher risk. There are many types of network vulnerabilities, but the description of such vulnerabilities often contains keywords such as "version", "club", etc. K-means text clustering analysis algorithm is employed to integrate the joint classification of vulnerabilities in CNNVD and CVE, and the aggregation results under 20 categories are obtained. Combining the data from two databases for analysis is proved to be helpful in further understanding the feature of the frequently exposed vulnerabilities.

References
[1] Schmidt, E., Rosenberg, J., Eagle, A., & EBSCO Publishing (Firm). (2014). How google works. NY: Grand Central Publishing, New York.
[2] Dong W. (2020) Application Research of Computer Information Technology in Network Security in the Era of Big Data. J. Information and Computers (Theoretical edition), 20, 32(19):196-198.
[3] China National Vulnerability Database of Information Security. (2020). http://www.cnnvd.org.cn.
[4] Common Vulnerabilities and Exposures. (2020). http://cve.mitre.org.