ASVC: An Automatic Security Vulnerability Categorization Framework Based on Novel Features of Vulnerability Data

Tao Wen[1], Yuqing Zhang[1,2], Qianru Wu[2], and Gang Yang[2]
[1] State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China;
[2] National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing 101408, China
Email: wentao_beijing@126.com; {zhangyq, wuqr, yang}@nipc.org.cn

Abstract — Security vulnerabilities are a main cause of network security. Vulnerability classification gives us a better understanding of the essence of vulnerabilities, which help propose efficient solutions. However, applying Vulnerability Categorization Standard (VCS) to manually categorize vulnerabilities is impracticable since it is time-consuming and subjective. To address this issue, a new framework named Automatic Security Vulnerabilities Categorization Framework (ASVC) is proposed based on Text Mining. To further improve the accuracy, a new rule for extraction of features of Text Mining is proposed. ASVC abstracts the categorization of vulnerabilities into a process of Text Mining, and categorize vulnerabilities automatically according to a VCS. Finally, VCS of Common Weakness Enumeration is applied to three main Vulnerability Databases based on ASVC in a fast way, about 1000 vulnerabilities per hour. The accuracy of the categorization is 86.8%, 8.3% higher than previous works.

Index Terms—Security vulnerability, vulnerability categorization, vulnerability database, information security, asvc, text mining

I. INTRODUCTION

With the development of network technologies, more and more security issues become prominent. Security vulnerabilities are a main cause of network security in recent years [1]. Delivering the integrated and correct vulnerabilities timely to security researchers, software vendors, system administers and users cannot only raise the safety awareness of the relevant personages, but also provide information for them to give patches and corresponding solutions [2].

Research [3] shows that lots of vulnerabilities are similar in some attributes. With the categorization, they can fully grasp various vulnerabilities and understand the essence of them better [4], eliminate the vulnerabilities and find unknown ones efficiently and increase the security of systems and software. At the same time, categorization can also provide detailed vulnerability information, attack manners and countermeasures for network attacks. Therefore, categorization of vulnerabilities is the precondition of vulnerability analysis in great depth [5].

Researchers have paid great attention to Vulnerability Categorization, so far it has 30 years of history [6] [7]. With the continuous appearance of new vulnerabilities and the increasing number of vulnerabilities, Vulnerability Categorization Standard (VCS) is also developed constantly [8]. MITRE Corporation summarized the advantages and disadvantages of previous works, proposed Common Weakness Enumeration (CWE) [9], which is the most authoritative and comprehensive VCS [10]. CWE provides a unified, measurable set of software weaknesses, and contains nearly all types of known vulnerabilities.

A. Problems of the Existing Categorizations

Although CWE has so many advantages, only NVD [11] uses the CWE in public since it is difficult to apply CWE to any Vulnerability Database. The reason is that CWE is unavailable in following cases:

- The information is not comprehensive [3]. It happens that the persons who categorize vulnerabilities are not the founders of vulnerabilities or the categorizers do not grasp all the necessary information about vulnerabilities, then they will determine the types of vulnerabilities by their subjective experience. In such cases, the categorization does not reach an objective standard.
- The founder of vulnerabilities does not understand the standard of the CWE [12]. In common cases, the categorizers are the founders of vulnerabilities; however, they do not master the standard of the categorization, then it will be difficult for them to categorize the vulnerabilities he found.
- Batch classification [13]. It happens that there are tens of thousands of vulnerabilities in a Vulnerability Database. When the administrator of Vulnerability Database wants to use the latest standard of the vulnerability categorization, he needs to consider the workload of the categorization first and guarantee the manual categorization to be objective.

Unfortunately, the situations mentioned above often appear in practice. The information about latest released vulnerabilities needs to be improved and the time span is often a couple of days or several months. The information about vulnerabilities is not comprehensive but there is a

Manuscript received December 27, 2014; revised February 27, 2015.
This work was supported by The Natural Science Foundation of China under Grant No.61272481.
Corresponding author email: wentao_beijing@126.com.
doi:10.12720/jcm.10.2.107-116

pressing need to determine the types of vulnerabilities in this time.

Recently, some researchers have made some preliminary explorations on Automatic Vulnerability Categorization Framework [14]-[17], which can apply a VCS to a specific Vulnerability Database. However, these Automatic Vulnerability Categorization Frameworks have three drawbacks:

- The taxonomic features of Text Mining of Automatic Vulnerability Categorization Frameworks are not universal, which cause not good enough operability. For example, some of them take the severity as the features of vulnerabilities to classify them [16]. However, the difficulty of obtaining the severity of vulnerabilities is the same as obtaining the categories of them.
- Some Automatic Vulnerability Categorization Frameworks only use the field of Description as the taxonomic feature and accuracy of them is low [15].
- Those Automatic Vulnerability Categorization Frameworks do not use the standard of CWE [15] but use the standard with fewer categories in the experiments instead (vulnerability categories are less than 10).

B. Contribution

In order to address the issues mentioned above, considering the ability of Text Mining in automatically finding the known information in the past and predicting the unknown information, we propose an automatic categorization framework of security vulnerabilities based on Supervised Learning Theory. The contributions are as follows (here we assume the VCS is CWE adopted by NVD):

- We propose a new automatic categorization framework of vulnerabilities termed Automatic Security Vulnerability Categorization (ASVC). It can classify vulnerabilities in the Target Vulnerability Database (Target VDB) in a batch. Advantages of ASVC include: (a) the steps are automatic, and millions of vulnerabilities can be classified fast; (b) As the process based on statistics of a large number of vulnerabilities, ASVC avoids the manual subjectivity; (c) Because ASVC extracts values from simple Description, Date and affected Vendors of vulnerabilities, which makes it be very suitable for the case of insufficient information, for instance, the latest ones.
- To improve the accuracy of categorization and the usability of ASVC, we propose a new method to extract the taxonomic features. The taxonomic features consist of three items: a. the field of Description, Title and Vendor of the vulnerability in Target VDB; b. the field of Description of the vulnerability entries of NVD; c. the field of Published Date of the vulnerability entries of Target VDB. Worth mentioned, the features which are used in a, b, c can be given in nearly all the Vulnerability Databases, so the universality of our method is very good.
- In order to optimize the result of CWE, we verify the best algorithms and parameters of vulnerability classification. We find that the number of the features is chosen from the first 20 values from CHI since if less than 20, the accuracy and the coverage rate will be low and if more than 20 the computation cost will increase.
- We collect and collate four representative Vulnerability Databases, NVD, OSVDB [18], X_Force [19] and Securityfocus [20], which contain 300 thousand of vulnerabilities totally. Then we apply the VCS (i.e. CWE, which NVD only adopted) to other three non-NVD Databases. In comparison with our framework, the accuracy of the method [15] which only uses the field of Description as features is 78.5%. However, our accuracy is 86.8% when using the method proposed in this paper. Finally, based on CWE, we explain the reason of errors in ASVC.

C. Organization

The rest of this paper is organized as follows. In Section 2, we review the related works in the categorization of vulnerabilities. In Section 3, we give the details of our categorization framework. In Section 4, we give the experiment results and the evaluation. Finally, in Section 5, we provide a conclusion of the paper and give the future work.

II. Related Work

A. Traditional Manual Categorizations of Vulnerabilities

In 1976, Ref. [6] proposed a manual categorization system termed Research into Secure Operating System (RISOS). The categorization mainly aims at vulnerabilities in the system operation, and it has seven categories. The subsequent Protection Analysis (PA) project increased the universal property of the categorization and it has four major categories and four smaller categories [7]. PA is more universal than RISOS. The aim of the PA is that everyone can find vulnerabilities using this model. Ref. [21]-[22] proposed a categorization scheme for vulnerabilities of Unix system for the first time. Subsequently, Ref. [23]-[25] improved the RISOS and PA, and proposed several categorization schemes respectively. However, these schemes are not perfect as a universal categorization scheme. In 2006, the presentation of the CWE had a remarkable significance in history [26]. It becomes the standard of the categorization of vulnerabilities. In the next few years, although new categorization schemes were proposed, there were no obvious breakthroughs [10].

B. Common Weakness Enumeration (CWE) Reformed by NVD

CWE summarizes the advantages and disadvantages of existing categorization standards and lists hundreds of

vulnerability categories. NVD selects 19 most practical vulnerability categories as the categorization standard of NVD and *Design_Error*, a new vulnerability category, is added. At present, only NVD takes advantage of the categorization standard of CWE in the known Vulnerability Databases. CWE, which is reformed by NVD, is the most authoritative currently.

Fig. 1 shows some of the vulnerability categories in CWE and the relationship among them. In Fig. 1, the shaded parts are the categories which have been used by NVD, such as CWE-22 and CWE-59 are the categories used in NVD, while CWE-21 is not used in NVD.

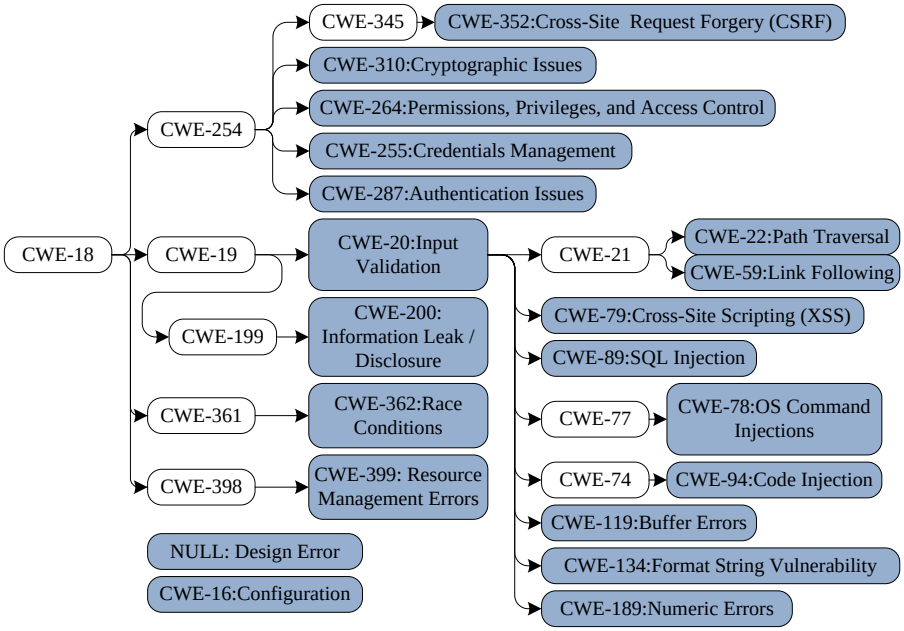


Fig. 1. The structure tree of the relationship among the vulnerability categories

C. Automatic Vulnerability Categorization

In the research of the automatic vulnerability categorization, Ref. [27]-[30] used the clustering algorithm to classify vulnerabilities without supervision. These works discussed how to find the hidden mode automatically. There were also some works on the research of the categorization with supervision. Ref. [14] classified vulnerabilities in Securityfocus. The algorithms they used include SVM and Bayes, and the taxonomic features they used are all the fields, such as Summary, Status, Severity Rank and so on. Ref. [12]-[13] took advantage of Bayes categorization algorithm to classify vulnerabilities in NVD. The features they used are Product and CVSS. However, for almost all the Vulnerability Databases, only the fields of Description, Vendors and Date are included, and the other fields are only included in particular databases. Therefore, in order to guarantee the usability of the automatic categorization, only the three fields can be used as the taxonomic features. Based on the categorization of NVD, Ref. [15] proposed a categorization framework. The field of Description is the taxonomic feature, but the feature is too simple, leading to low accuracy of categorization when CWE is used.

In a word, the research on manual categorization is mature. CWE represents the most comprehensive and reasonable VCS. However, there is still a big challenge in implementing the VCS in a batch. The research center of the categorization of the vulnerability has been transferred from the categorization of the vulnerability to the automatic categorization.

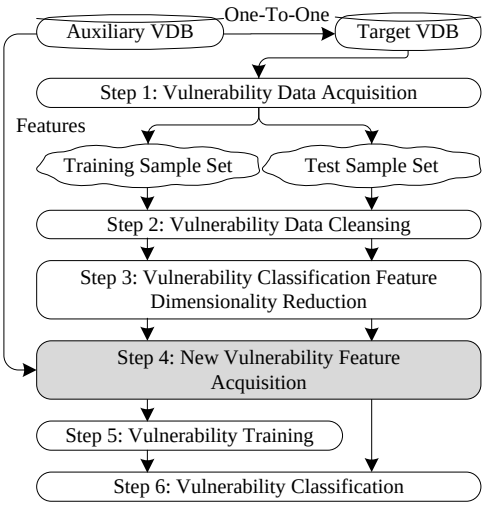


Fig. 2. The flow chart of ASVC

III. ASVC: An Automatic Security Vulnerability Categorization Framework

In this section, we introduce our automatic categorization framework. The idea of the framework is abstracting the categorization of vulnerabilities into a process of Text Mining.

We select the standard of CWE, which has been reformed by NVD, as VCS. According to its characteristics, NVD chooses 19 typical vulnerability categories from CWE and adds other one category as a supplement which is not from CWE. We will use these 20 (19 plus 1) vulnerability categories as VCS. It means we automatically classify vulnerabilities in Target Vulnerability Database (Target VDB) according to the 20 categories. According to the standard, the whole categorization can be done in a batch and fast way.

According to the characteristics of vulnerabilities, we show the procedure of ASVC in Fig. 2. The detailed procedure is as follows (Shadow step in Fig. 2 is the core algorithm in the whole process).

A. Vulnerability Data Acquisition

In this paper, we aim at classifying vulnerabilities in Target VDB, so we need to obtain the data of vulnerabilities first.

Furthermore, we also need to obtain the data in NVD as an Auxiliary Vulnerability Database (Auxiliary VDB). In order to verify the accuracy of the classification algorithm, we only choose vulnerabilities, which are in one-to-one correspondence between Auxiliary VDB and Target VDB as Training Data. Worth mentioned, "one-to-one vulnerabilities" are vulnerabilities which are repeated in both of the Vulnerability Databases, for example the vulnerability X is recorded in Auxiliary VDB and marked N1 and it is also recorded in Target VDB and marked T1, then we say N1 and T1 have a one-to-one relationship. The aim of choosing the one-to-one vulnerabilities is to know the actual categories of vulnerabilities in advance.

Besides choosing the one-to-one vulnerabilities, we also need to obtain all the vulnerabilities in NVD, so we can get the feature set of *FT_DNC* in step 4.

B. Vulnerability Data Cleansing

The vulnerability data which is obtained from Target VDB needs to be pretreated since it cannot be used in Text Mining. The purpose of data cleansing is to get the data in regular forms. After cleansed, the data will be the vector of strings which are stored in the form of single words. The steps of data cleansing are as follows:

- Segment the words. The classification features mainly come from the text of the fields of Description, Title and Vendor which are some words segmented with spaces. The fields with space are separated and stored in the form of single word vectors.
- Remove special symbols, such as the commas, periods, brackets and line breaks and so on. Worth mentioned, not all the special symbols are removed, for instance, the brackets which represent the function call need to be reserved, such as "*save()*" which is the name of a function in a source code, if we remove "()", the original meaning of it will be changed.
- Remove the words without effective information. In order to increase the efficiency of features, we need to remove some words which do not have effective information related to categorization, for example, the words whose length is 1 (a, b, c); and the pure digital words (1234).
- Remove stop words, such as "are" and "what" and so on, since these words have no real effect on the categorization of vulnerabilities.
- Deal with the tense and grammar. Change the passive voice and plural into the original form.

C. Vulnerability Classification Features Dimensionality Reduction

In the Text Mining, the words which have been segmented are usually used as a dimension. The frequency of the words is the value of the dimension. For example, we assume the word sequence of the vulnerability *O1* is: *word*, *sql*, *computer*, and the word sequence of the vulnerability *O2* is: *sql*, *computer*, *function*. Let *O1* and *O2* are a set, then the features include four dimensions: *word*, *sql*, *computer* and *function*, where the value of *O1* is *{1,1,1,0}*, the value of *O2* is *{0,1,1,1}*.

The Description of vulnerabilities consists of a huge number of words. So the dimensionality curse will occur when we classify the text if we do not deal with the words. Such a large dimension is not realistic. Therefore, dimensionality reduction is needed. The frequently-used dimensionality reduction algorithm is Document Frequency (DF) [31], Information Gain (IG) [32], Mutual Information (MI) [33] and Chi-square (CHI) [34] at present. Among them, IG and CHI are the best [35]. However, the problem of IG is that it only can investigate the features which contribute to the whole system, i.e., the global features. It cannot choose the features against individual categories [36]. So IG is only suitable in the case that the number of the sets is close to each other in different categories. However, the vulnerability numbers of each category are significantly different respectively in this paper, so if we use IG algorithm the result will not be perfect. Therefore, we use the CHI algorithm as the dimensionality reduction algorithm. The specific calculation formula is shown in equation (1).

$$\chi^2 = \frac{N(AD-BC)^2}{(A+C)(A+B)(B+D)(B+C)} \quad (1)$$

In the formula, *N* denotes the total number of documents in the statistical sample set, *A* denotes the frequency of occurrence of some word's positive document, *B* denotes the frequency of occurrence of some words' negative document, *C* denotes the frequency of nonoccurrence of some words' positive document, *D* denotes the frequency of nonoccurrence of some words' negative document. Every unique word (i.e., a feature dimensionality) needs to compute a value χ^2 against a category (20 categories in all), for example, when there are 1000 dimensionalities, 20000 values will be computed. These values are in descending order. A certain number of these values are chosen as the features

of this category. However, too many features we choose increase computation complexity, while few features decrease accuracy and coverage rate.

D. New Vulnerability Feature Acquisition

The taxonomic features we used are the fields of Description, Title, Vendor and Date of the vulnerability entries in Target VDB and the field of Description of the vulnerability entries in Auxiliary VDB.

We found almost all the Vulnerability Databases have the fields of Description, Title, Vendor and Date. It means the universality of these four fields is the best. So we extract the feature set of these four fields.

As we know, tens of thousands of vulnerabilities in NVD have been classified according to the standard of CWE, however, the vulnerabilities we extracted are only a few thousands which have the one-to-one relationship in NVD and Target VDB. So we take the fields of Description of all the vulnerabilities in NVD as an independent data set. It can provide more universal information and make a correction on the feature sets from Target VDB.

The feature sets to be extracted are as follows:

- *FT_DO* (Description Only as Features). *FT_DO* denotes the field of Description in Target VDB. Applying the CHI dimensionality reduction algorithm, we extract the first n feature words in each category (20 categories in all) of the vulnerability in Target VDB. The field of Description has good universality (every vulnerability has the field of Description).
- *FT_TV* (Title and Vendor as Features). *FT_TV* denotes the field of Title and Vendor in Target VDB. We need to mix and remove the duplicate words which have been segmented from Title and Vendor of each vulnerability in Target VDB. Then applying the CHI dimensionality reduction algorithm, extract the first n feature words of each category from the mixed word sequence.
- *FT_DT* (Date). *FT_DT* denotes the field of Date. We first need to get the releasing time of each vulnerability, then we make 1/1/1900 as 0, the number increases one if the days increase one. For example, we define 2/1/1900 as 1, and then the Date of 1/1/1901 will be 365 and so on. Use the Date which has been transformed as a feature.
- *FT_DNC* (Description in Auxiliary VDB as Features). *FT_DNC* denotes the fields of NVD Description. Applying the CHI dimensionality reduction algorithm, extract the first n feature words of each category (20 categories in all) of the vulnerability entry in NVD. These words are the feature words of NVD Description. Worth mentioned, we need to deal with all the vulnerabilities which have been classified in NVD not only the one-to-one correspondence vulnerability entries which have been extracted.

E. Vulnerabilities Training and Classification

Compared with the Bayes algorithm [37] and the Random Forests algorithm [38], SVM algorithm [39] used in this paper has high accuracy but is time-consuming. Considering the requirement of time we need is lower than that of accuracy in this paper, we use SVM algorithm as the classification algorithm.

SVM algorithm maps the sample space to feature space with a high dimension by nonlinear mapping. The mapping transfers the nonlinear and separable problem in the original sample space transformed into the linear and separable problem in the feature space. Then different kernel functions generate different SVM. Frequently-used kernel functions are: (1) Liner; (2) Polynomial; (3) Radial; (4) Sigmoid.

The kernel function we used is Radial. After data cleansing and feature selection, the Training Set gets the feature vectors as the input of SVM. Through training the Training Set, we get the optimal weight and model. Then we can test the Test Set and compute the categorization of vulnerabilities in the Test Set.

IV. Experiment and Analysis

In the implementation of our system, the operating system we use is Windows 7, the database system is Microsoft SQL Server and algorithm implementation we use is the package of R language. The memory of the system needs to be larger than 8G.

A. Vulnerability Data Acquisition

TABLE I: THE NUMBER OF VULNERABILITIES OBTAINED

Vulnerability Database	The number of vulnerabilities obtained	Training Set	Test Set
OSVDB	98252	9041	4521
Securityfocus	58201	3296	1648
X_Force	84761	5593	2797
NVD	65200	65200	--
Total	301214	17932	8966

The aim of our experiment is to apply the CWE, which is used by NVD to Target VDB. So we select NVD as Auxiliary VDB, select Securityfocus, OSVDB and X_Force as Target VDB. We can easily know the categories of some vulnerabilities in Target VDB which ones can be found in NVD. In practice, all of equivalent vulnerabilities in NVD are chosen as Training Set; however, we only choose parts of these vulnerabilities as Training Set in this paper, and the rests are chosen as Test Set. The purpose is to get accuracy objectively. In this paper, we collect and collate the four Vulnerability Databases, which contain 300 thousands of vulnerabilities totally, the number of vulnerabilities we obtained are shown in Table I. All of the data obtained up to 2013 November.

B. Analysis of the Number of the Features

The total number of the features is the feature words of each category multiplied by 20. Fig. 3 shows the relationship between the number of feature words and the accuracy of the categorization. The y-coordinate is the accuracy and the x-coordinate is the number of the

feature words. From Fig. 3 we can see, the accuracy reaches the maximum when the number of the feature words is 15 and the accuracy is almost unchanged when the number of the feature words increases. Fig. 4 shows the relationship between the number of feature words and the coverage rate of the vulnerability. The y-coordinate is the coverage rate and the x-coordinate is the number of the feature words. From Fig. 4 we can see, the coverage rate is 99.5% when the number of the feature words is 8 and it goes up to 100% when the number of the feature words is greater than 15%. Fig. 5 shows the relationship between the number of feature words and the training time of the categorization. The y-coordinate is the training time and the x-coordinate is the number of the feature words. From Fig. 5 we can see, the training time increases with the number of the features linearly. So considering the accuracy, the coverage rate and the training time, it is optimal when the number of the features is 20 in each category.

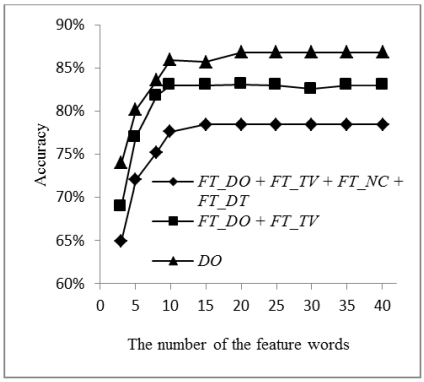


Fig. 3. The distribution of the accuracy varying with the number of the feature words

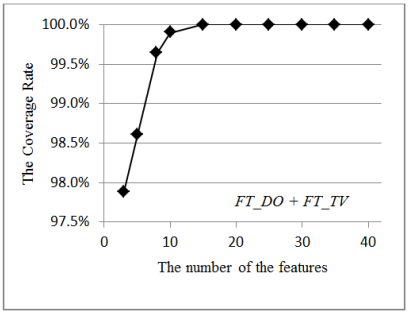


Fig. 4. The distribution of the coverage rate varying with the number of the feature words

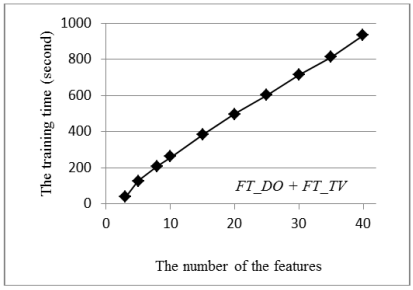


Fig. 5. The distribution of the training time varying with the number of the feature words

C. Analysis of the Features

Table II shows the comparison of the categorization results from the classifier, where the number of the features is 20. The accuracy is computed by equation (2),

$$Accuracy = 1 - \frac{|V_M|}{|V|} \quad (2)$$

In equation (2), the numerator $|V_M|$ represents the number of test cases which have been classified in wrong categories and the denominator $|V|$ represents the number of the test cases. From Table II we can see, following the model of Ref. [15], the accuracy of the categorization is 78.5% if we only use the Description feature of vulnerabilities in Target VDB. The accuracy of the categorization increases greatly if we add the words Title and Vendor. The ultimate accuracy will be 86.8% if we further add the Description in NVD and the Date in Target VDB. The reason is that there are about 60000 vulnerabilities in NVD, which have been classified according to the standard of CWE, so it is more comprehensive and objective to use the vulnerability data in NVD to choose the feature words. The reason why we add Date into the features lies in that every type of vulnerabilities appears at different time and a category of the vulnerability will go through several stages, such as discovery, development, maturity, balance and suppression, so the distribution of vulnerability categories is related to the published time of vulnerabilities.

TABLE II: THE COMPARISON OF THE ACCURACY OF THE CATEGORIZATION BETWEEN DIFFERENT TAXONOMIC FEATURES

Feature Mode	Accuracy
FT_DO	78.5%
FT_DO + FT_TV	83.0%
FT_DO + FT_TV + FT_DNC + FT_DT	86.8%

TABLE III: THE FIELDS APPLIED BY EACH FRAMEWORK

	Description	Vendor	Title	Date	CVSS	Reason	Source	Impact
BNVC [12]		√			√			
LVCM [13]	√	√	√	√	√	√	√	
OSBC [14]	√					√		√
CVCF [15]	√							
ASVC	√	√	√	√				

TABLE IV: THE COMPARISON OF EACH FIELD

	Description	Vendor	Title	Date	CVSS	Reason	Source	Impact
Applicable Rate	100%	100%	100%	100%	68.5%	50.7%	61.8%	56.7%
Accuracy	78.5%	47.9%	53.2%	11.0%	65.6	64.2%	23.7	59.8%

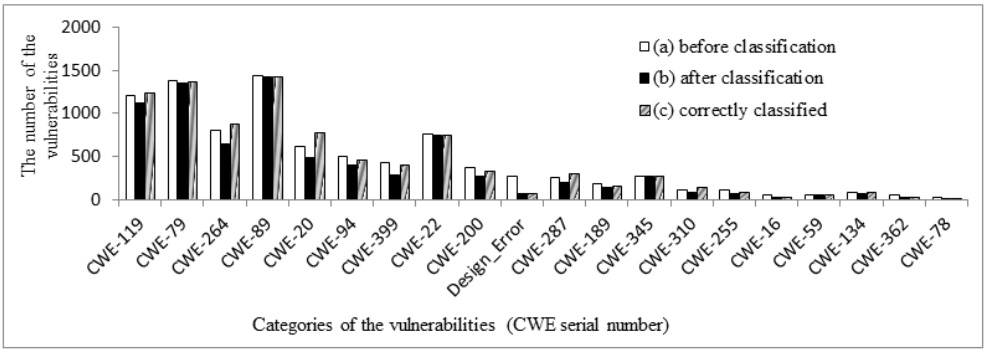


Fig. 6. The number of vulnerabilities in each category

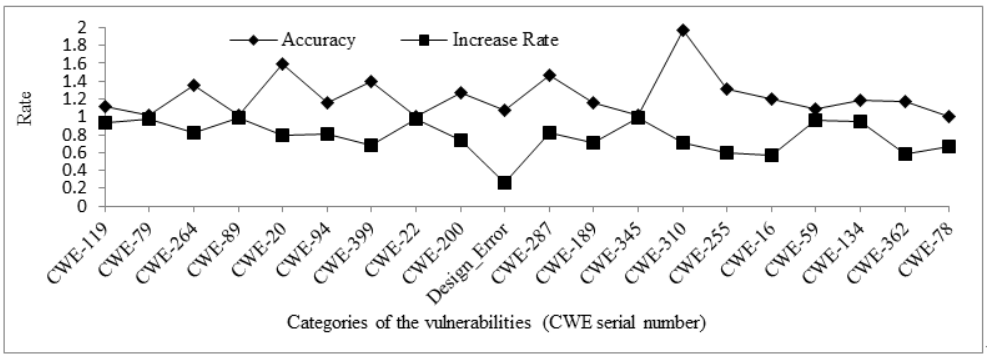


Fig. 7. The accuracy and increase rate of various categories of vulnerabilities

D. The Comparison of Other Frameworks

In this section, we compared ASVC with other similar automatic vulnerability categorization frameworks, which contained BNVC [12] (Bayesian Networks Vulnerability Categorization), LVCM [13] (LDA Vulnerability Classification Mode), OSBC [14] (Open Source Software Bug Characteristics) and CVCF [15] (Common Vulnerabilities Categorization Framework). Above-mentioned frameworks have different vulnerability fields, see in Table III. Each vulnerability field has different characteristics in the aspects of the applicable rate and accuracy, see in Table IV. We computed the applicable rate and accuracy of above-mentioned frameworks. Table VI shows that the applicable rate of CVCF and ASVC are 100%, that is only CVCF and ASVC can be applied widely. Considering accuracy, ASVC is the best framework in the aspect of comprehensive performance.

TABLE V: THE COMPARISON OF EACH FRAMEWORK

	Applicable Rate	Accuracy
BNVC [12]	68.5%	76.1%
LVCM [13]	50.7%	92.9%
OSBC [14]	50.7%	82.5%
CVCF [15]	100%	78.5%
ASVC	100%	86.8%

E. The Analysis of the Concrete Category

Fig. 6 shows the number of vulnerabilities in different categories. The features are *FT_DO* + *FT_TV* + *FT_DNC* + *FT_DT* and the number of the features is 20.

In Fig. 6, the x-coordinate represents the CWE serial number of the vulnerability categories, where the category of *Design_Error* is added by NVD based on CWE and they have no CWE serial number. The y-coordinate represents the number of vulnerabilities and the order is: (a) the number of vulnerabilities, which belong to some category before classification. That is the real classification; (b) the number of vulnerabilities, which have been classified into the category after classification. Actually, some vulnerabilities may be classified in wrong categories; (c) the number of vulnerabilities, which have been correctly classified.

The increasing rate in Fig. 7 represents the ratio of the number of the categories which have been classified into the actual number of the categories. Fig. 8 shows the relationship among (a), (b) and (c). The names of vulnerabilities, which correspond to the serial number of vulnerabilities in Fig. 6 and Fig. 7, are shown in Fig. 1.

Combining Fig. 1, Fig. 6 and Fig. 7, we can draw following conclusions:

- The number of the vulnerability categories used in CWE reformed by NVD can be divided into six clusters, see Table VI.

- The vulnerabilities of *Design_Error* are easy to be classified into other categories. The category of *Design_Error* added by NVD may destroy the integrity of the CWE standard. This conclusion is based on that the category whose increasing rate is the minimum is *Design_Error*. From the results of the classification, the category of *Design_Error* has weak compatibility with the categories of CWE. Since the category of *Design_Error* is distributed into the five clusters uniformly, it means the category of *Design_Error* added by NVD may destroy the integrity of the CWE standard.
- The misclassification rate is high if vulnerabilities are in the same cluster and it is low if vulnerabilities are in different clusters. This conclusion is based on the fact that the increase rate of CWE-20 (Input_Validation) is the highest. The reason is that other categories in the cluster of Input_Validation in CWE belong to the subclasses of CWE-20, so it is probable to misclassify these subclasses into CWE-20.

It is note that, although CWE is the most authoritative vulnerability classification scheme, CWE have not been applied widely, in fact, the version revised by NVD is the only application of CWE. Therefore, although we consider that the CWE revised by NVD is not perfect, there is no better choice. In addition, the main purpose of this paper is to realize the automation of application of classification scheme, rather than design a new scheme. So in order to highlight the focus, we do not make a futher discussion.

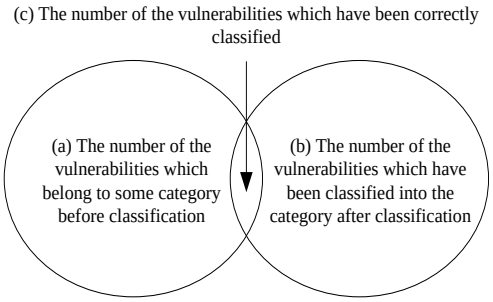


Fig. 8. The relationship among (a), (b) and (c)

TABLE VI: THE CLUSTERS OF THE CATEGORIES OF VULNERABILITIES

The name of the clusters	CWE serial number
Input_Validation	CWE-20, 22, 59, 79, 89, 78, 94, 119, 134, 189
Access_Control_and_Resource_ Management	CWE-399, 352, 310, 264, 255, 287
Information_Leak / Disclosure	CWE-200
Race_Conditions	CWE-362
Configuration	CWE-16
Design_Error	NULL

V. CONCLUSIONS AND FUTURE WORK

Categorizing vulnerabilities manually wastes time. Meanwhile, lots of the information about vulnerabilities is not comprehensive. Therefore, the categorization standards, such as CWE, are difficult to apply.

To address the above problems, we propose an Automatic Security Vulnerability Categorization Framework (ASVC) based on the Text Mining. To improve the accuracy of categorization and the usability of ASVC, we propose a new method to obtain taxonomic features and the result is perfect. What's more, in order to optimize the result of CWE, we also verify the best algorithms and parameters of ASVC.

We classify vulnerabilities in Target Vulnerability Databases, includes OSVDB, Securityfocus and X_Force with CWE, and the accuracy is 86.8%. If we use the categorization method proposed by Ref. [15], the accuracy of the categorization is 78.5%. So, according to the comparison, the categorization method proposed in this paper has better performance. Finally, based on CWE, we explain the reason of errors in ASVC.

Our further work will be in the following aspects: (1) in order to increase the accuracy, we need to improve the algorithm and find more efficient features extraction algorithms; (2) classify vulnerabilities based on the data mining algorithm without supervision.

ACKNOWLEDGMENTS

The authors wish to thank The Natural Science Foundation of China (61272481). The authors also wish to thank the anonymous reviewers for providing constructive suggestions.

REFERENCES

[1] H. Y. Cao, J. J. Zhao, P. D. Zhu, X. C. Lu, and C. L. Zhao, "Worm detection without knowledge base in industrial networks," *Journal of Communications*, vol. 8, no. 11, pp. 716-723, November 2013.

[2] T. Uemura and T. Dohi, "Optimal security patch management policies maximizing system availability," *Journal of Communications*, vol. 5, no. 1, pp. 71-80, Jane 2010.

[3] H. Ghani, J. Luna, A. Khelil, and N. Alkadri, "Predictive vulnerability scoring in the context of insufficient information availability," in *Proc. International Conf. Risks and Security of Internet and Systems*, La Rochelle, 2013, pp. 1-8.

[4] Q. X. Liu and Y. Q. Zhang, "VRSS: A new system for rating and scoring vulnerabilities," *Computer Communications*, vol. 34, no. 3, pp. 264-273, March 2011.

[5] Q. X. Liu, Y. Q. Zhang, and Y. Kong, "Improving VRSS-based vulnerability prioritization using analytic hierarchy process," *The Journal of Systems and Software*, vol. 85, no. 8, pp. 1699-1708, August 2012.

[6] R. Abbott, J. Chin, and J. Donnelley, "Security analysis and enhancements of computer operating systems," in *Proc. US Department of Commerce National Bureau of Standards*, Washington, D.C., 1976, pp. 26-41.

[7] I. R. Bisbey and D. Hollingworth, "Protection analysis: Final report," in *Proc. Marina Del Rey Conf.*, 1978, pp. 1-10.

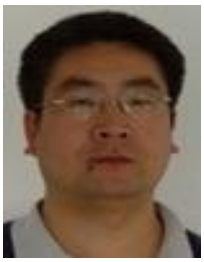
[8] U. Lindqvist and E. Jonsson, "How to systematically classify computer security intrusions," in *Proc. IEEE Symposium Conf. Security and Privacy*, 1997, pp. 154-163.

[9] CWE. Common Weakness Enumeration. [Online]. Available: https://nvd.nist.gov/cwe.cfm.

[10] H. S. Venter, J. H. P. Eloff, and Y. L. Li, "Standardising vulnerability categories," *Computers & Security*, vol. 27, no. 3-4, pp. 71-83, May 2008.

[11] NVD. National Vulnerability Database. [Online]. Available: http://web.NVD.nist.gov/
[12] J. A. Wang and M. Z. Guo, "Vulnerability categorization using bayesian networks," in *Proc. ACM International Conf. 6th Annual Cyber Security and Information Intelligence Research Workshop: Cyber Security and Information Intelligence Challenges and Strategies*, Oak Ridge, 2010.
[13] X. F. Liao, Y. J. Wang, X. B. Fan, and J. Z. Wu. "National security vulnerability database classification based on an LDA topic model," *Journal of Tsinghua University*, vol. 52, no. 10, pp. 1351-1355, 2012.
[14] Z. M. Li, L. Tan, X. H. Wang, S. Lu, Y. Y. Zhou, and C. X. Zhai, "Have things changed now? An empirical study of bug characteristics in modern open source software," in *Proc. ASPLOS 2006 Conf. 1st Workshop on Architectural and System Support for Improving Software Dependability*, San Jose, 2006, pp. 25-33.
[15] Z. Q. Chen, Y. Zhang, and Z. R. Chen, "A categorization framework for common computer vulnerabilities and exposures," *The Computer Journal*, vol. 53, no. 5, pp. 551-580, June 2010.
[16] J. A. Wang, H. Wang, M. Z. Guo, H. Wang, and L. F. Zhou, "Ranking attacks based on vulnerability analysis," in *Proc. 43rd Annual Hawaii International Conf. System Sciences*, Honolulu, 2010, pp. 1-10.
[17] J. A. Wang, M. Guo, H. Wang, M. Xia, and L. F. Zhou, "Ontology-based security assessment for software products," in *Proc. Fifth Annual Cyber Security and Information Intelligence Research Workshop Conf. Cyber Security and Information Intelligence Challenges and Strategies*, Oak Ridge, 2009, pp. 159-168.
[18] OSVDB. Open Source Vulnerability Database. [Online]. Available: http://OSVDB.org/
[19] X. Force. IBM Internet Security Systems Ahead of the threat. [Online]. Available: http://xforce.iss.net/
[20] Securityfocus. [Online]. Available: http://www.securityfocus.com/
[21] M. Bishop, "A taxonomy of UNIX system and network vulnerabilities," Technical Report CSE-9510, 1995.
[22] T. Aslam, I. Krsul, and E. Spafford, "Use of a taxonomy of security faults," in *Proc. West Lafayette Conf.*, 1996, pp. 21-22.
[23] C. E. Landwehr, A. R. Bull, J. P. Mcdemott, and W. S. Choi, "A taxonomy of computer program security flaws," *ACM Computing Surveys*, vol. 26, no. 3, pp. 211-254, September 1994.
[24] S. Kumar and E. Spafford, "A taxonomy of common computer security vulnerabilities based on their method of detection," Technical Report, COAST Laboratory, West Lafayette, 1995.
[25] M. Bishop and D. Bailey, "A critical analysis of vulnerability taxonomies," in *Proc. University of California at Davis Conf.*, 1996, pp. 30-45.
[26] M. Howard, "Improving software security by eliminating the CWE top 25 vulnerabilities," *IEEE Security and Privacy*, vol. 7, no. 3, pp. 21-32, May 2009.
[27] L. L. DeLooze, "Classification of computer attacks using a self-organizing map," in *Proc. 5th Annual IEEE SMC Conf. Information Assurance Workshop*, West Point, 2004, pp. 365-369.
[28] A. Tripathi and U. K. Singh, "Evaluation of severity index of vulnerability categories," *International Journal of Information and Computer Security*, vol. 5, no. 4, pp. 210-219, 2013.
[29] H. S. Venter and J. H. P. Eloff, "Vulnerabilities categories for intrusion detection systems," *Computers and Security*, vol. 21, no. 7, pp. 617-619, 2002.
[30] T. Melanie. A Comparison of Word Frequency and N-Gram Based Vulnerability Categorization Using SOM. [Online]. Available: http://www.cra.org/Activities/craw_archive/cdmp/awards/2008/Tupper/CDMP2008.pdf
[31] D. Karakos, M. Dredze, K. Church, A. Jansen, and S. Khudanpur, "Estimating document frequencies in a speech corpus," in *Proc. IEEE Workshop Conf. Automatic Speech Recognition and Understanding*, Waikoloa, 2011 pp. 407-412.
[32] Z. Gao, Y. J. Xu, F. Y. Meng, Q. Feng, and Z. Q. Lin, "Improved information gain-based feature selection for text categorization," in *Proc. 2014 4th International Conf. Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronics Systems*, Aalborg, 2014, pp. 1-5.
[33] J. Y. Liang, X. P. Liu, K. N. Huang, X. Li, D. G. Wang, and X. W. Wang, "Automatic registration of multisensor images using an integrated spatial and mutual information (SMI) metric," *IEEE Transactions Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 603-615, 2013.
[34] X. H. Niu, F. Shi, J. B. Xia, X. H. Hu, and N. N. Li, "Comparisons among the novel measurements based on chi square criterion for sequence dissimilarity and their applications to phylogeny," in *Proc. Fifth International Conf. Computational and Information Sciences*, Shiyang, 2013, pp. 470-473.
[35] S. L. Liu, X. Chen, W. S. Liu, J. Q. Chen, Q. Gu, and D. X. Chen, "FECAR: A feature selection framework for software defect prediction it cannot choose features against individual categories," in *Proc. IEEE 38th Annual Computer Software and Applications Conf.*, Vasteras, 2014, pp. 426-435.
[36] C. Z. Yang, C. C. Hou, W. C. Kao, and I. X. Chen, "An empirical study on improving severity prediction of defect reports using feature selection," in *Proc. 19th Asia-Pacific, Software Engineering Conf.*, Hong Kong, 2012, pp. 240-249.
[37] Z. F. Chen, R. Molina, and A. K. Katsaggelos, "A variational approach for sparse component estimation and low-rank matrix recovery," *Journal of Communications*, vol. 9, no. 8, pp. 600-611, September 2013.
[38] J. Lv and Y. G. Yan, "Estimating leaf chlorophyll concentration in soybean using random forests and field imaging spectroscopy," in *Proc. Third International Conf. Agro-geoinformatics (Agro-geoinformatics 2014)*, Beijing, 2014, pp. 1-4.
[39] X. M. Liu and J. S. Tang, "Mass classification in mammograms using selected geometry and texture features, and a new svm-based feature selection method," *IEEE Systems Journal*, vol. 8, no. 3, pp. 910-920, September 2014.

Tao Wen is a postgraduate student of Xidian University, China. He received his B.Sc. in Fudan University, in 2007. He received his M.Sc. in Geosciences University, in 2011. Currently, he is studying security information and security vulnerability.



Yuqing Zhang is a professor and supervisor of Ph.D. students of Graduate University of Chinese Academy of Sciences. He received his B.Sc. and M.Sc. in Computer Science from Xidian University, China, in 1987 and 1990 respectively. He received his Ph.D. degree in Cryptography from Xidian in 2000. His research interests include cryptography, wireless security and trust management.

Qianru Wu is a postgraduate student of Information Science and Engineering of Graduate University of Chinese Academy of Sciences, Beijing, China. She received her B.Sc. in Information Management and System Program from Shaanxi Normal University of China, in 2011. Currently, she is studying web application security.

Gang Yang is a postgraduate student of Information Science and Engineering of Graduate University of Chinese Academy of Sciences, Beijing, China. He received his B.Sc. in Wuhan University, in 2014. Currently, he is studying security vulnerability.