# Associating the Severity of Vulnerabilities with their Description

Dimitrios Toloudis, Georgios Spanos, and Lefteris Angelis(✉)

Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
toloudisd@gmail.com, {gspanos,lef}@csd.auth.gr

**Abstract.** Software vulnerabilities constitute a major problem for today's world, which relies more than ever to technological achievements. The characterization of vulnerabilities' severity is an issue of major importance in order to address them and extensively study their impact on information systems. That is why scoring systems have been developed for the ranking of vulnerabilities' severity. However, the severity scores are based on technical information and are calculated by combining experts' assessments. The motivation for the study conducted in this paper was the question of whether the severity of vulnerabilities is directly related to their description. Hence, the associations of severity scores and individual characteristics with vulnerability descriptions' terms were studied using Text Mining, Principal Components and correlation analysis techniques, applied to all vulnerabilities registered in the National Vulnerability Database. The results are promising for the determination of severity by the use of the description since significant correlations were found.

**Keywords:** Information security · Software vulnerability · Text mining · Statistical analysis

## 1 Introduction

It is well known that information is the most valid asset of the contemporary society. Due to the ever-increasing growth of technological achievements and the globalization of society, the need for collecting, storing and processing information has become extremely important. Within this scenery, the security of data and procedures in today's information systems holds a particularly significant role. Governments, industries as well as individuals pay great effort and invest a lot of assets to preserve the Confidentiality, Integrity and Availability (CIA triad) of information. Those three concepts constitute the main pillars of Information Security.

Information Security (InfoSec) has to deal with many primary or secondary problems but one of the most important may be considered the existence of software vulnerabilities. According to the Common Vulnerabilities and Exposures (CVE) reference system [1], "*An information security "vulnerability" is a mistake in software that can be directly used by a hacker to gain access to a system or network*." In other words, software vulnerability is a bug or a flaw in the code/design of a software program that can be exploited by an unauthorized user in order to intercept, alternate or


J. Krogstie et al. (Eds.): CAiSE 2016, LNBIP 249, pp. 231–242, 2016.
DOI: 10.1007/978-3-319-39564-7_22

interrupt information flow. Hence, the intruder can have access to sensitive information, misguide the user or even take control of an entire critical infrastructure.

Unfortunately, it is practically impossible for any development team to produce vulnerability-free software. Mistakes, big or small, may appear during the requirements gathering, in the design of the software, in the development/coding, even during the training of the final user. As software starts to be used, operators or experts find vulnerabilities that developers have to fix. But the number of the faults is enormous and the need for fast solutions is urgent. That is why, specific characteristics of vulnerabilities are measured and scores that represent their severity are computed. This way, IT security managers can prioritize them and deal initially with the vital and then with the relatively harmless ones.

The most widely used scoring system is the Common Vulnerability Scoring System (CVSS), version 2 [2] that constitutes the improvement of CVSS version 1 [3] and is used also from the National Vulnerability Database (NVD) [4]. A third version of CVSS with different scoring factors has been presented in December 2014 but since is not yet applied by NVD for all the vulnerabilities but only for the newest ones (after December 2015), it can not be examined in the present analysis. A relatively new scoring system, which uses alternative approach regarding the weighing of the vulnerability characteristics in comparison with CVSS, is the Weighted Impact Vulnerability Scoring System (WIVSS) [5] while an improvement of WIVSS has been published in [6]. Other known scoring systems constitute: the Vulnerability Rating Scoring System (VRSS) [7], the improvement of VRSS [8] and the PVL [9]. Furthermore, there are many vulnerability databases that gather data for all vulnerabilities or just for a specific category. The most famous and world widely used is the National Vulnerability Database (NVD), which holds a variety of data (score, dates, individual characteristics, description and more) for all kind of vulnerabilities.

The severity of the vulnerabilities is apparently an issue of major importance and their scoring is performed taking into account technical information coming from experts' characterizations. However the question that motivated the current research is whether the description of each vulnerability, which is registered in the database as text, is informative about the severity, i.e. whether the words used to describe the vulnerabilities are correlated with the overall severity scoring. Therefore, the purpose of this paper is to perform a text analysis of the vulnerability descriptions existing in the NVD -which was used as the data source for this research- and to extract useful relationships between the terms that were derived from the text mining and the characteristics/scores (the scores are provided from CVSS version 2 and the improvement of WIVSS) of vulnerabilities, using correlation analysis and principal component analysis.

The rest of this paper is organized as follows. Section 2 describes the related work, in Sect. 3, the database along with the vulnerability characteristics and scorings systems (CVSS, WIVSS) are described, while in Sect. 4 the results of the analysis are presented. Finally, Sect. 5 summarizes the conclusions.

## 2 Related Work

In order to characterize the severity of a vulnerability, experts have to evaluate and define its characteristics. This is a particularly time-consuming process, especially if someone takes into consideration the enormous number of new vulnerabilities and the great need for their immediate confrontation. That is why many scientists scrutinize the characteristics, the life cycle and the behavior of vulnerabilities, finding correlations between them and developing prediction models. In those researches, a variety of "tools" are being used, like statistical and data mining methods.

One of the above techniques is the Text Mining (TM), which can remove needless words and numbers and replace characters or terms so that finally can extract metadata into word-frequency-count Document-Term Matrices (DTM). This specific technique is constantly gaining ground among the scientific community by being increasingly used. Several studies, papers and books have been written for the use of TM and even more for researches in which the methodology was used.

Representative examples that use TM in biomedical research are the following: Theodosiou et al. [10] in 2007, presented a study about an alternative of gene functional annotation throw-out classification modeling and validation. For that purpose, they used TM in biomedical articles, excluding non-informative words and extracting useful metadata. Similar use of TM was made for another analysis of biomedical article datasets in 2008 [11] in which, Non-Linear Canonical Correlation Analysis (NLCCA) was used for exploring the correlation among the variables (words) of multiple representations of biomedical documents. Finally, Janasik et al. [12] published a paper in 2009, about the use of TM in qualitative research and the self-organizing map (SOM) method as well as the inference quality improvement that this implementation may achieve.

Regarding the Information Security field -which is also the research field of the present study- Hovsepyan et al. [13] transformed the source code of many programs into plain text, totally ignoring its complexity, its churn, its size or other characteristics. Thus, via the use of TM, they managed to predict vulnerabilities with fairly good results. Furthermore, in 2012, Liu et al. [14] achieved promising results on analysis and automatic classification of network vulnerabilities, applying TM in data, retrieved from a variety of online sources. Moreover, Nishanth et al. [15] used TM and data mining techniques in order to analyze and classify the risk levels of phishing attacks in financial firms. Using either MLP, PNN or DT, the achieved accuracy was above 80 %. Chen et al. [16] classified not only the risk levels of phishing attacks but also its impact to market value of the attacked firms, by using TM and data mining in phishing alerts and firms' financial data and they also distinguished variables with significant impact in the seriousness of the attacks. Finally, Wang et al. [17] conducted both quantitative and qualitative analysis in order to measure the financial impact of the information security incidents reported in firm financial reports. Initially they examined the influence of the amount of announcements on stock prices and then they correlated the impact with specific term that derived from the TM methodology. This way, they developed a model helpful in evaluation of disposed information of firms' incidents reports.

In this paper, we analyze the textual descriptions of vulnerabilities using TM techniques and associate the most important terms or groups of terms with the vulnerability characteristics and severity.

## 3 Database, Characteristics and Scoring Systems

For this study, logs of all vulnerabilities up to 5 Aug 2015 were retrieved from NVD [4]. The total number was higher than 70,000 (specifically 70,678) and they contained, among others, the values of vulnerability characteristics and a brief description for every one of them. It must be highlighted here, that the whole set of vulnerabilities that exist in NVD was used for this research, instead of an easier-to-manage sample, like vulnerabilities from one year or of those that affect specific software. This approach targeted in more accurate results and more general view.

For every vulnerability we can identify some characteristics that describe the way that a flaw can be exploited and the impact that may have to the affected systems. Their values are being determined by experts, either through their experience or by conducting specific measurements. Those characteristics are:

- **Access Vector**: defines the way a vulnerability can be exploited. The values that can take are: *Local*, *Adjacent Network* or *Network*.
- **Access Complexity**: defines how difficult is to exploit the vulnerability. The values that can take are: *High*, *Medium* or *Low*.
- **Authentication**: defines the level of user authentication levels needed for the exploitation of the vulnerability. The values that can take are: *Multiple*, *Single* or *None*.
- **Confidentiality Impact**: defines how much, the exploitation of vulnerability, can influence the confidentiality of the system. The values that can take are: *None*, *Partial* or *Complete*.
- **Integrity Impact**: defines how much, the exploitation of vulnerability, can influence the integrity of the system. The values that can take as above are: *None*, *Partial* or *Complete*.
- **Availability Impact**: defines how much, the exploitation of vulnerability, can influence the availability of the system: The possible values also are *None*, *Partial* or *Complete*.

The above characteristics are used for the computation of unified scoring systems that represent the severity of vulnerabilities. The most famous and widely used scoring system is the Common Vulnerability Scoring System (CVSS), which was originally developed by the National Infrastructure Advisory Council (NIAC) in 2004 [3]. In 2007, an improved version was released [2]. Nowadays, responsible for CVSS is the Forum of Incident Response and Security Teams (FIRST) [18] and the Common Vulnerability Scoring System-Special Interest Group (CVSS-SIG) [19].

Another scoring system, that improves CVSS in terms of diversity of values, is the Weighted Impact Vulnerability Scoring System (WIVSS). It was originally developed in 2013 by Spanos et al. [5], considering different weights for the Impact Metrics in contrast to CVSS, which considers the same weights for all Impact Metrics. In 2015,

the second version of WIVSS was published [6], improving further the value diversity of the previous version. Both CVSS and WIVSS have the same scoring range (0.0–10.0) with scale step of 0.1. Detailed information regarding the computational formulas of the above scoring systems can be found in [6].

## 4 Correlation Analysis

The first step of the correlation analysis was to isolate the useful terms from the vulnerability descriptions. By the use of Text Mining numbers, commonly used words with no useful meaning and words like brands and software names were removed. Also, the remaining words were transformed in order to remove word endings, to convert upper to lower cases or to unify words with similar meaning. Finally, a Document Term Matrix (DTM) was created, containing the number of appearances for every term, in every vulnerability description. From that matrix, words with very low frequency of appearance were excluded and the result was a DTM with 33 words, which are shown in Fig. 1, sized according to their frequencies. Also, Table 1 contains the term frequencies (percentage form) in vulnerability descriptions.


**Fig. 1.** Word cloud of vulnerability description

The representation of Term Frequency - Inverse Document Frequency (TF-IDF) [20] has been selected for the representation of data in the DTM. TF-IDF is a numerical statistic that is used in text mining and reflects the importance of a word in a document, taking into consideration its general appearance frequency in a group of documents. It is widely used for the recognition and exclusion of useless terms, like stop words, for document categorization and summarization and also by search engines. As its name shows, TF-IDF is the combination of the Term Frequency statistic and the Inverse Document Frequency technique.

Additionally, the CVSS and WIVSS scores along with the characteristics of vulnerabilities were added to the previous matrix in order to conduct the correlation analysis. For the same reason, the vulnerability characteristics (Access Vector, Access

**Table 1.** The term frequencies

| Term | Frequency | Term | Frequency | Term | Frequency |
|---|---|---|---|---|---|
| *allow* | 95.43 % | *earlier* | 13.57 % | *paramet* | 22.87 % |
| *arbitrari* | 52.63 % | *execut* | 33.84 % | *remot* | 77.52 % |
| *attack* | 80.84 % | *file* | 21.09 % | *script* | 14.68 % |
| *authent* | 11.60 % | *function* | 10.18 % | *server* | 13.62 % |
| *buffer* | 10.68 % | *html* | 13.13 % | *servic* | 23.42 % |
| *caus* | 21.90 % | *inform* | 12.09 % | *unspecifi* | 20.45 % |
| *code* | 21.68 % | *inject* | 22.08 % | *use* | 11.19 % |
| *command* | 13.54 % | *local* | 12.97 % | *user* | 22.55 % |
| *craft* | 15.60 % | *multipl* | 11.93 % | *vector* | 19.43 % |
| *crosssitscript* | 13.21 % | *obtain* | 12.24 % | *vulner* | 49.57 % |
| *denial* | 21.04 % | *overflow* | 11.20 % | *web* | 19.01 % |

Complexity, etc.) that are essentially ordinal variables in the sense that the order of their values reflects the vulnerability severity, were represented by numerical values, as shown in Table 2.

**Table 2.** Representation of values of the characteristics

| Metric vector | Representation of values |
|---|---|
| Access Vector (AV) | 1 = Local, 2 = Adjacent Network, 3 = Network |
| Access Complexity (AC) | 1 = High, 2 = Medium, 3 = Low |
| Authentication (Auth) | 1 = Multiple, 2 = Single, 3 = None |
| Confidentiality Impact (CI) | 1 = None, 2 = Partial, 3 = Complete |
| Integrity Impact (II) | 1 = None, 2 = Partial, 3 = Complete |
| Availability Impact (AI) | 1 = None, 2 = Partial, 3 = Complete |

Finally, correlation analysis was applied to the enhanced data matrix, using the non-parametric Spearman's rho correlation coefficient [21], which can be also used to test the correlation between ordinal and continuous variables. Spearman's correlation coefficient takes values in the interval [−1, +1] and a value close to −1 or +1 respectively shows negative or positive monotonic correlation. All the correlation coefficients for all the pairs of variables formed as a combination of one variable-word/term and one variable-characteristic/score are shown in Table 3.

The analysis showed that the vast majority of correlations are statistically significant, as the significance (p-value) of the corresponding test is almost everywhere less than $a = 0.01$. However, the majority of them are very weak or weak (absolute value of Spearman's rho is less than 0.4). Note that rules-of- thumb characterize as "moderate" the correlation when the absolute value of Spearman's rho is in the interval [0.4, 0.6], "strong" for [0.6, 0.8] and very strong for [0.8, 1.0]. We decided to consider and comment only correlations with rho coefficient greater than 0.3 since we believe that even a weak or a moderate correlation can imply the existence of an informative word

**Table 3.** Spearman's rho correlation coefficients (terms vs characteristics and scores)

| | AV | AC | Auth | CI | II | AI | CVSS | WIVSS |
|---|---|---|---|---|---|---|---|---|
| ***allow*** | -0.052 | -0.056 | -0.019 | 0.074 | 0.061 | 0.082 | -0.048 | -0.042 |
| ***arbitrari*** | 0.231 | 0.133 | -0.057 | 0.016 | 0.218 | 0.013 | 0.174 | 0.235 |
| ***attack*** | **0.301** | 0.023 | **-0.354** | 0.093 | 0.123 | 0.11 | 0.15 | 0.086 |
| ***authent*** | 0.111 | -0.003 | **0.648** | 0.085 | 0.079 | 0.059 | -0.109 | -0.078 |
| ***buffer*** | -0.005 | -0.042 | -0.06 | -0.139 | -0.166 | -0.086 | 0.237 | 0.236 |
| ***caus*** | 0.033 | -0.073 | -0.064 | -0.212 | -0.277 | 0.044 | 0.034 | -0.199 |
| ***code*** | 0.128 | 0.031 | -0.069 | -0.135 | -0.172 | -0.108 | **0.426** | **0.434** |
| ***com-mand*** | 0.071 | -0.188 | 0.005 | 0.201 | 0.163 | 0.226 | 0.218 | 0.206 |
| ***craft*** | -0.115 | 0.21 | -0.039 | -0.098 | -0.113 | -0.061 | 0.084 | 0.058 |
| ***crosssites cript*** | 0.164 | **0.422** | 0.004 | -0.128 | **0.337** | -0.096 | **-0.415** | **-0.344** |
| ***denial*** | 0.039 | -0.065 | -0.062 | -0.221 | -0.284 | 0.047 | 0.038 | -0.204 |
| ***earlier*** | 0.052 | -0.054 | -0.031 | 0.062 | 0.071 | 0.08 | 0 | -0.005 |
| ***execut*** | 0.176 | -0.093 | -0.067 | 0.057 | 0.014 | 0.094 | **0.537** | **0.538** |
| ***file*** | -0.067 | -0.051 | -0.041 | 0.072 | -0.042 | -0.004 | 0.003 | 0.051 |
| ***function*** | -0.058 | -0.031 | -0.019 | -0.049 | -0.071 | 0 | 0.012 | -0.015 |
| ***html*** | 0.161 | **0.4** | 0.005 | -0.14 | **0.304** | -0.108 | **-0.396** | **-0.328** |
| ***inform*** | -0.161 | 0.057 | -0.027 | 0.19 | 0.004 | 0.021 | -0.114 | -0.053 |
| ***inject*** | 0.218 | 0.197 | 0 | 0.099 | **0.441** | 0.131 | -0.166 | -0.13 |
| ***local*** | **-0.701** | -0.112 | -0.032 | -0.129 | -0.18 | -0.156 | -0.173 | -0.054 |
| ***multipl*** | 0.105 | 0.082 | -0.022 | 0.016 | 0.136 | 0.044 | 0 | 0.009 |
| ***obtain*** | -0.185 | 0.044 | -0.019 | 0.204 | -0.01 | 0.017 | -0.121 | -0.051 |
| ***overflow*** | 0 | -0.038 | -0.06 | -0.149 | -0.173 | -0.092 | 0.25 | 0.248 |
| ***paramet*** | 0.209 | 0.028 | -0.064 | 0.223 | **0.337** | 0.199 | 0.008 | 0.03 |
| ***remot*** | **0.526** | -0.007 | 0.032 | 0.166 | 0.199 | 0.182 | 0.084 | 0.013 |
| ***script*** | 0.143 | **0.358** | -0.004 | -0.099 | **0.315** | -0.077 | **-0.366** | -0.297 |
| ***server*** | -0.142 | 0.03 | 0.018 | 0.028 | -0.024 | 0.031 | -0.014 | -0.017 |
| ***servic*** | 0.037 | -0.068 | -0.041 | -0.221 | -0.288 | 0.026 | 0.041 | -0.184 |
| ***unspecifi*** | 0.054 | -0.005 | 0.142 | -0.11 | -0.088 | -0.122 | 0.044 | 0.046 |
| ***use*** | -0.039 | -0.021 | -0.018 | -0.016 | -0.047 | -0.024 | -0.008 | 0.002 |
| ***user*** | **-0.489** | -0.095 | **0.434** | -0.058 | -0.094 | -0.102 | -0.259 | -0.139 |
| ***vector*** | 0.071 | -0.013 | 0.122 | -0.102 | -0.064 | -0.107 | 0.042 | 0.055 |
| ***vulner*** | 0.216 | 0.129 | 0.035 | 0.02 | 0.202 | 0.013 | 0.022 | 0.063 |
| ***web*** | 0.189 | **0.363** | -0.008 | -0.099 | 0.235 | -0.111 | **-0.316** | -0.253 |

for the severity of vulnerabilities. All of them are of course statistically significant ($p < 0.001$) and are especially highlighted in Table 3.

We can identify positive correlation between Access Vector and the term "*attack*" (0.301), while negative correlation exists with the term "*user*" (−0.489). More important are the strong negative correlation (−0.701) between Access Vector and the

term "*local*" and the positive correlation with the term "*remote*" (0. 526). These relationships are quite anticipated since Access Vector defines the way that a vulnerability can be exploited, i.e. locally or via network.

Moreover, notable positive correlations were found among Access Complexity and the terms "*crosssitescript*" (0.422), "*html*" (0.400), "*script*" (0.358) and "*web*" (0.363). These terms are related to the injection of scripts in web to exploit a vulnerability, so their relationship with less access complexity is reasonable.

Furthermore, according to the results of correlation analysis, Authentication has positive correlation with the term "*authent*" (0.648), which refers to what the metric measures. Also, Authentication is also correlated with the terms "*user*" (0.434) and "*attack*" (−0.354), positively and negatively respectively.

Continuing with the Impact metrics, there are not any notable correlations between the vulnerability description terms and the Confidentiality/Availability Impact but Integrity Impact seems to be positively correlated with five terms. These terms are: "*inject"* (0.441), "*crosssitscript*" (0.337), "*html*" (0.304), and "*script*" (0.315) and this is an indication that the injection of scripts in web is related to the defacement of websites. Finally the term "*paramet*" (0.337) is also correlated with the Integrity Impact.

Regarding the vulnerability scoring systems, they are correlated with all description terms quite similarly, although in some very weak correlations there is not even agreement in the sign. So, CVSS and WIVSS are positively correlated with the term "*code*" (0.426 and 0.434 respectively) and with the term "*execut*" (0.537 and 0.538). These correlations reflect that the execution of code to exploit vulnerabilities concerns more severe vulnerabilities. In contrary, the terms "*crosssitscript*" and "*html*" are negatively correlated with the two scoring systems (−0.415, −0.396 for CVSS and −0.344, −0.328 for WIVSS). Thus, these terms, which are related to the injection of scripts in the web (as mentioned above), are not correlated with severe vulnerabilities. Finally, CVSS is negatively correlated with the terms "*script*" (−0.366) and "*web*" (−0.316), which were found positively correlated with Access Complexity and Integrity Impact.

In order to consider the internal correlation structure of DTM with respect to characteristics and severity, the variables representing the terms of the description were analyzed by principle component analysis (PCA) with varimax rotation of the axes [22]. PCA produces uncorrelated linear combinations of the original variables. The new variables (or components) account for decreasing amounts of the total variation (i.e. the first component explains the maximum variance, and so on) and their estimations can be used for variable reduction and representation of the data points in lower dimensions. The components essentially form groupings of the participating variables with highly correlated variables within each group.

We tried several different PCA settings in order to find a good model and after excluding from the analysis the terms: {*craft*, *earlier*, *file*, *function*, *multipl*, *paramet*, *server*, *use*, *vulner*}, due to their low contribution to the model, we concluded in a model with 9 components which explains 83.08 % of the total variation. Each principal component extracted is highly correlated with a number of terms either positively or negatively. The nine components in descending order of importance (% of the variance they explain) together with the terms correlated with them are given in Table 4. The sign (+) or (−) following each term shows a positive or negative correlation.

**Table 4.** Results of PCA

| Component number | Variance explained (%) | Terms correlated with the component |
|---|---|---|
| 1 | 16.57 | *crosssitscript*(+), *html*(+), *script*(+), *web*(+), *inject*(+) |
| 2 | 12.13 | *denial*(+), *caus*(+), *servic*(+) |
| 3 | 10.04 | *local*(−), *user*(−), *attack*(+), *remot*(+) |
| 4 | 8.89 | *buffer*(+), *overflow*(+) |
| 5 | 8.31 | *obtain*(+), *inform*(+) |
| 6 | 7.92 | *allow*(+), *arbitrary*(+), *execut*(+) |
| 7 | 7.73 | *vector*(+), *unspecifi*(+) |
| 8 | 5.80 | *command*(+), *code*(−) |
| 9 | 5.69 | *authent*(+) |

Note that the model after the exclusion of 9 terms, explains the correlation structure of 24 terms.

It is interesting to see in Table 4 how the description terms are grouped in subsets according to their internal correlation. For example, the most important component in the dataset is Component #1 which explains 16.57 % of the total variability and is positively correlated with the terms {*crosssitscript*, *html*, *script*, *web*, *inject*} (these terms are "loaded" on the 1st component according to the standard PCA terminology). So PCA in our case is a way to "summarize" many terms together, essentially by finding new, latent variables that are correlated to subsets of them. The values of these new variables can be estimated and can be used for further analysis. We estimated these values by the Anderson-Rubin method, so new standardized (mean = 0 and standard deviation = 1) and uncorrelated among them variables were produced.

In Table 5 we provide the Spearman correlation coefficients between the components as formed and estimated from the correlation structure among description terms and the vulnerability characteristics and scores. Almost all correlations were found statistically significant ($p < 0.01$) but also most of them are very weak or weak. Notable correlations are between:

- Access Vector and the 3rd component (0.517), which represents the group of terms {*local*, *user*, *attack*, *remot*}. Note that the first two terms are loaded negatively on the component while the other two positively, so the anticipated interpretation is that higher values of the AV are correlated with the presence of *attack* and *remot* but with the absence of *local* and *user*.
- Authentication and the 9th component (0.349), which includes only one term, *authent*. The correlation between Authentication and *authent* was found also previously.
- CVSS and 1st component (−0.417), 3rd component (0.353) and 5th component (−0.352). As previously noticed, CVSS is correlated negatively with vulnerabilities related to injection of scripts in web and the 1st component concerns these vulnerabilities. Furthermore, the positive correlation with 3rd component depicts that

more severe vulnerabilities (according to CVSS) are those concerning remote attacks (terms: *remot* and *attack*) and not local users (terms: *local* and *user*). Moreover, the negative correlation with 5th component reflects that CVSS does not scores highly Confidentiality oriented vulnerabilities (terms *obtain* and *information*).

- WIVSS and 1st component (−0.429), 2nd component (−0.405) and 5th component (−0.334). Similar behavior with CVSS, regarding the correlations with 1st and 5th component (although, the negative correlation with 5th component seems somehow weird). Additionally, the negative correlation with 2nd component is reasonable since these terms (*denial*, *caus* and *servic*) are met in Availability oriented vulnerabilities and WIVSS considers Availability as the less severe factor in the CIA triad.

**Table 5.** Spearman's rho correlation coefficients (components vs characteristics and scores)

| Component number | AV | AC | Auth | CI | II | AI | CVSS | WIVSS |
|---|---|---|---|---|---|---|---|---|
| **1** | -0.025 | 0.236 | -0.050 | -0.129 | 0.124 | -0.057 | **-0.417** | **-0.429** |
| **2** | -0.049 | 0.055 | -0.101 | -0.176 | -0.128 | -0.019 | -0.264 | **-0.405** |
| **3** | **0.517** | -0.068 | -0.286 | 0.150 | 0.062 | 0.148 | **0.353** | 0.249 |
| **4** | 0.092 | 0.196 | 0.154 | -0.108 | 0.033 | -0.107 | 0.039 | 0.104 |
| **5** | -0.147 | 0.024 | -0.138 | 0.096 | 0.000 | 0.025 | **-0.325** | **-0.334** |
| **6** | 0.077 | -0.030 | -0.027 | 0.078 | 0.067 | 0.076 | 0.173 | 0.196 |
| **7** | -0.060 | -0.012 | -0.042 | -0.076 | -0.044 | -0.096 | -0.088 | -0.077 |
| **8** | -0.039 | -0.060 | -0.020 | 0.048 | 0.180 | 0.179 | -0.183 | -0.262 |
| **9** | 0.296 | -0.060 | **0.349** | 0.103 | 0.068 | 0.103 | -0.185 | -0.244 |

Overall, we can clearly see that the description terms either single or in groups appear to be correlated with the technical characteristics and the severity scores of vulnerabilities. Although the correlations are not strong, the findings are interesting in the sense that the descriptions contain certain terms or combinations of terms that are quite informative for several security aspects. In order to strengthen our previous results, we further conducted two simple linear least squares regression analyses: In both of them we considered as independent variables the 9 component scores found by PCA while as dependent variables we considered in the first model the CVSS score and in the second model the WIVSS score. The purpose was to see how a severity score, which in a sense summarizes the vulnerability characteristics (such as CVSS and WIVSS), is correlated with all the components together, which also are used for summarizing many terms. In both models all components were found significant ($p < 0.001$), while the r-square statistic for the CVSS model was 0.383 and for the WIVSS model was 0.361. That essentially means that by a simple linear model based on PCA components we can explain the 38 % of CVSS and the 36 % of WIVSS variation.

## 5 Conclusion

In this paper we considered and analyzed software vulnerabilities that constitute one of the most critical issues of computer security. Using a methodology with ever-increasing popularity and acceptance as the Text Mining, we transformed the vulnerability descriptions from text to numerical data and we obtained a data matrix, called Document Term Matrix, which we subsequently used in order to perform correlation analysis between the most frequently appeared terms of vulnerability descriptions and the vulnerability characteristics/scores. The results revealed that there are many worth-mentioning correlations among the above terms, either single or in groups, and the characteristics/scores. However, the nature of this dependence deserves further investigation. Although, simple linear models contribute in understanding a moderate amount of the severity, the fitting of more advanced, probably non-linear models seems to be necessary in order to express adequately the relation between severity and description terms.

The knowledge derived from the present work is useful for researchers in the field of Information Security, but also for IT security managers who can be aided in decision making, regarding the severity and the characteristics of a vulnerability by analyzing small descriptions that exist in vulnerability databases. Although automated severity characterization has its own risks, the diagnosis of severity by statistical tools can be useful aid for human decisions. In this paper we explored and showed that there are serious potentials in the utilization of the description in this regard.

The text analysis provided in the present paper, although was applied on the data of NVD (which is a technical database), the conclusions are generic and could help to the characterization of vulnerabilities by descriptions registered in other structured or non-structured data sources (journal articles, websites and blogs, etc.).

As future research, we plan to combine text mining and machine learning techniques in order to construct powerful diagnostic models, using training data from the wealthy NVD vulnerability source and having as ultimate goal the accurate and, if possible, automated assessment of the vulnerability severity and characteristics.

## References

1. CVE – Terminology. https://cve.mitre.org/about/terminology.html
2. Mell, P., Scarfone, K., Romanosky, S.: A complete guide to the common vulnerability scoring system version 2.0 (2007). https://www.first.org/cvss/v2/guide
3. Schiffman, M., Cisco, C.I.A.G.: A complete guide to the common vulnerability scoring system (cvss) (2005). http://www.first.org/cvss/v1/guide
4. NVD – National Vulnerability Database. https://nvd.nist.gov/
5. Spanos, G., Sioziou, A., Angelis, L.: WIVSS: a new methodology for scoring information systems vulnerabilities. In: Proceedings of the 17th Panhellenic Conference on Informatics, pp. 83–90. ACM, New York (2013)
6. Spanos, G., Angelis, L.: Impact metrics of security vulnerabilities: analysis and weighing. Inf. Secur. J. Gobal Perspect. **24**(1–3), 57–71 (2015)

7. Liu, Q., Zhang, Y.: VRSS: a new system for rating and scoring vulnerabilities. Comput. Commun. **34**(3), 264–273 (2011)
8. Liu, Q., Zhang, Y., Kong, Y., Wu, Q.: Improving VRSS-based vulnerability prioritization using analytic hierarchy process. J. Syst. Softw. **85**(8), 1699–1708 (2012)
9. Wang, Y., Yang, Y.: PVL: a novel metric for single vulnerability rating and its application in IMS. J. Comput. Inf. Syst. **8**(2), 579–590 (2012)
10. Theodosiou, T., Angelis, L., Vakali, A., Thomopoulos, G.N.: Gene functional annotation by statistical analysis of biomedical articles. Int. J. Med. Inform. **76**, 601–613 (2007)
11. Theodosiou, T., Angelis, L., Vakali, A.: Non-linear correlation of content and metadata information extracted from biomedical article datasets. J. Biomed. Inform. **41**, 202–216 (2008)
12. Janasik, N., Honkela, T., Bruun, H.: Text mining in qualitative research: application of an unsupervised learning method. Organ. Res. Methods **12**, 436–460 (2009)
13. Hovsepyan, A., Scandariato, R., Joosen, W., Walden, J.: Software vulnerability prediction using text analysis techniques. In: 4th International Workshop on Security Measurements and Metrics, pp. 7–10. ACM, New York (2012)
14. Liu, C., Li, J., Chen, X.: Network vulnerability analysis using text mining. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ACIIDS 2012, Part II. LNCS, vol. 7197, pp. 274–283. Springer, Heidelberg (2012)
15. Nishanth, K., Ravi, V., Ankaiah, N., Bose, I.: Soft computing based imputation and hybrid data and text mining: the case of predicting the severity of phishing alerts. Expert Syst. Appl. **39**, 10583–10589 (2012)
16. Chen, X., Bose, I., Leung, A., Guo, C.: Assessing the severity of phishing attacks: a hybrid data mining approach. Decis. Support Syst. **50**, 662–672 (2011)
17. Wang, T.-W., Rees, J., Kannan, K.: Reading the disclosures with new eyes: bridging the gap between information security disclosures and incidents. In: 7th Workshop on the Economics of Information Security (WEIS), Hanover, NH (2008)
18. FIRST.org/FIRST - Improving security together. http://www.first.org
19. CVSS-SIG Team. https://www.first.org/cvss/v2/team
20. Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets, vol. 1. Cambridge University Press, Cambridge (2012)
21. Sheskin, D.J.: Handbook of Parametric and Non-parametric Statistical Procedures. Chapman & Hall/CRC, Boca Raton (2004)
22. Bartholomew, D.J., Steele, F., Moustaki, I., Galbraith, J.I.: The Analysis and Interpretation of Multivariate Data for Social Scientists. Chapman & Hall/CRC, Boca Raton (2002)