Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 932397, 9 pages
http://dx.doi.org/10.1155/2014/932397

Research Article

A Software Vulnerability Rating Approach Based on the Vulnerability Database

Jian Luo, Kueiming Lo, and Haoran Qu

School of Software, Tsinghua University, Beijing 100084, China

Correspondence should be addressed to Jian Luo; j-luo10@mails.tsinghua.edu.cn

Received 14 March 2014; Accepted 14 May 2014; Published 29 May 2014

Academic Editor: Xiaoyu Song

CVSS is a specification for measuring the relative severity of software vulnerabilities. The performance values of the CVSS given by CVSS-SIG cannot describe the reasons for the software vulnerabilities. This approach fails to distinguish between software vulnerabilities that have the same score but different levels of severity. In this paper, a software vulnerability rating approach (SVRA) is proposed. The vulnerability database is used by SVRA to analyze the frequencies of CVSS's metrics at different times. Then, the equations for both exploitability and impact subscores are given in terms of these frequencies. SVRA performs a weighted average of these two subscores to create an SVRA score. The score of a vulnerability is dynamically calculated at different times using the vulnerability database. Experiments were performed to validate the efficiency of the SVRA.

1. Introduction

The common vulnerability scoring system (CVSS), developed and maintained by the CVSS special interest group (CVSS-SIG) working under the auspices of the forum for incident response and security teams (FIRST), can be applied to the classification of security vulnerability [1] and the analysis of attack models [2]. CVSS has been adopted by many software vendors and service providers [3]. The US federal government uses it for its National Vulnerability Database [4] and mandates its use in products validated by the security content automation protocol (SCAP) program.

There exist many proprietary schemes for rating software flaw vulnerabilities, most are created by software vendors, but CVSS is the only known open specification. In contrast to other scoring systems, CVSS was designed to be quantitative so that analysts would not have to perform qualitative evaluations of vulnerability severity. Great effort has been directed at developing the specification for CVSS so that any two vulnerability analysts should obtain identical CVSS scores for the same vulnerability. The scores are based on a series of measurements (called metrics) based on expert assessment.

1.1. Overview of CVSS Framework. CVSS provides an open framework for describing the characteristics and impacts of IT vulnerabilities. It contains three groups of metrics (see Figure 1), as explained in [5, 6].

(1) *Base.* It represents the intrinsic and fundamental characteristics of a vulnerability that are time-constant across user environments. An equation is applied to the values of the base metrics to compute a vulnerability's base score.

(2) *Temporal.* It represents the characteristics of a vulnerability that change over time but apply to all instances of a vulnerability in all environments, such as the public availability of an exploit code or a remediation technique. A temporal score for a vulnerability is calculated with an equation that uses both the base score and temporal metric values as parameters.

(3) *Environmental.* It captures the characteristics of a vulnerability that are associated with users IT environment. Since environmental metrics are optional, they each include a metric value that has no effect on the score. An environmental score is calculated with

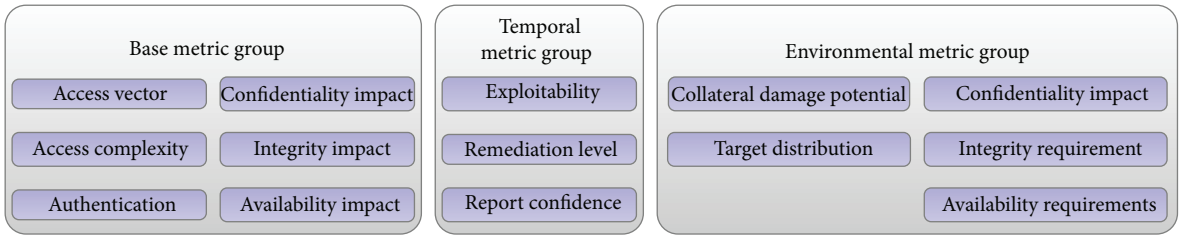

Figure 1: CVSS metric groups.

an equation that uses both the temporal score and the environmental metric values as parameters.

The initial CVSS specification was developed by the National Infrastructure Advisory Council and published in October 2004 [7]. During the analysis and use of the original CVSS version, many deficiencies were found, as explained in [8]. Finalized in 2007, the current version (CVSS v2) was designed to address these deficiencies. The base metric group of CVSS v2 has two subscores.

(1) *Exploitability subscore* E, composed of the access vector (AV), access complexity (AC), and authentication instances (AU), is computed by the following equation:

$$E = 20 \times \mathrm{AV} \times \mathrm{AC} \times \mathrm{AU}. \quad (1)$$

(2) *Impact subscore* I, which expresses the potential damage on confidentiality (CI), integrity (II), and availability (AI), is computed as follows:

$$I = 10.41 \times (1 - (1 - \mathrm{CI}) \times (1 - \mathrm{II}) \times (1 - \mathrm{AI})). \quad (2)$$

Table 1 gives all possible values of the six base metrics in v2, which are used to calculate these two subscores. The overall base score of v2 is expressed in terms of impact (I) and exploitability (E) components by

$$B = \begin{cases} 1.176 \times \left(\frac{3I}{5} + \frac{2E}{5} - \frac{3}{2}\right) & \text{if } I \neq 0, \\ 0 & \text{if } I = 0. \end{cases} \quad (3)$$

The base score is rounded to one decimal place and ranges from 0.0 to 10.0. More details related to CVSS metrics and their scoring computation can be found in the CVSS guide [5].

1.2. Shortcomings of CVSS v2. We downloaded 54,432 vulnerabilities listed in the common vulnerabilities and exposures (CVE) dictionary [9]; this encompasses all valid CVE entries published between 2002 and 2012. The scoring was performed by the national vulnerability database (NVD) [4] in accordance with the v2 specification.

When scoring separates vulnerabilities, they should be scored completely independently of each other and not take into account any interaction. According to v2 scoring tip number 1:

> *"Vulnerability scoring should not take into account any interaction with other vulnerabilities. That is, each vulnerability should be scored independently."*

SPSS, a type of statistical software, was used to perform the Chi-square analysis, and the results indicate *there exist correlations among the six metrics*. For example, Table 2 shows the statistical data related to the frequencies of AV and AC. These data were used as inputs by SPSS, and the results of the Chi-square analysis are given in Table 3. Asymp. Sig. is smaller than the significance level 0.05, indicating that the correlation between AV and AC is significant. Similarly, there exist significant correlations between other metrics, for example, II and AI.

Most importantly, CVSS v2 fails to distinguish different vulnerabilities. As an example, a path disclosure flaw in a web application would be scored as (AV = Network, AU = None, AC = Low, CI = Partial, II = None, AI = None) for a total score of 5.0. A vulnerability that allows an attacker to traverse the file system and read any file accessible by the web server would receive the same score as the path disclosure flaw. These two flaws obviously pose significantly different risks, yet according to CVSS v2 standards, they are identical.

Once the v2 metrics were defined, opinions related to the scoring for each type of vulnerability were collected from the CVSS-SIG members and their organizations. Each of the six metrics had three possible values, resulting in 729 possible vulnerability types. It was not possible to create scores for these 729 types in the range [0.0, 10.0] in a justifiable manner. So, the researchers divided the base metrics into two subgroups: impact and exploitability. Each group had three metrics with three possible values, so only 27 vulnerability types per group had to be scored and ranked. The researchers reached consensus on the approximated rankings and scorings, leading to the creation of lookup tables for impact and exploitability. The CVSS score was computed by a weighted average of exploitability and impact. However, the CVSS community desired an equation instead of lookup tables. So, mathematicians proposed equations (1)–(3) to approximate the lookup tables. In essence, these equations were derived from the designers' experience and statistical results of vulnerability data. As time went on, it became clear

that these equations, as well as the empirical values listed in Table 1, might no longer be applicable.

1.3. Our Design Methodology. To overcome the shortcomings mentioned above, a software vulnerability rating approach (SVRA) is proposed. SVRA takes time as an important parameter. Based on a vulnerability database, it counts the frequencies of the six metrics at any given time point. Then, the three values of each metric are given by their frequencies. As the frequencies change over time, each metric takes different values instead of a constant value.

The process of exploiting a vulnerability is a step-by-step procedure, but the impact is an evolutionary and accumulative process. So, the frequency of the vector (AV, AC, AU) is used to approximate the exploitability, while the frequencies of CI, II, and AI are utilized to calculate the impact. To create an SVRA score from these two subscores, SVRA also performs a weighted average of exploitability and impact, with exploitability having a weight of 0.4 and impact having a weight of 0.6. In terms of design methodology, SVRA is fundamentally different from CVSS v1 and CVSS v2. The score of a vulnerability in SVRA dynamically changes over time, which is not true in v2 or v1.

The rest of this paper is organized as follows. The next section provides the framework of the SVRA. Section 3 describes the analysis of and comparison between CVSS and SVRA, and the experimental results are also reported. Section 4 summarizes our conclusions and highlights some suggestions for future work.

Table 1: Possible values of CVSS base metrics.

AV	0.395 (local)	0.646 (Adj.Net)	1 (network)
AC	0.35 (high)	0.61 (medium)	0.71 (low)
AU	0 (multiple)	0.56 (single)	0.704 (none)
CI, II, AI	0 (none)	0.275 (partial)	0.66 (complete)

Table 2: Input data for Chi-square analysis.

AV	AC	Count	Percentage
Local	High	471	0.87%
Local	Medium	1207	2.22%
Local	Low	5463	10.04%
Adj.Net	High	27	0.05%
Adj.Net	Medium	85	0.16%
Adj.Net	Low	146	0.27%
Network	High	2104	3.87%
Network	Medium	15796	29.02%
Network	Low	29133	53.52%

Table 3: Chi-square tests.

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-square	833.805[a]	4	.000
Likelihood ratio	909.498	4	.000
N of valid cases	54432		

[a]0 cells (0.0%) have expected count less than 5. The minimum expected count is 12.33.

2. Software Vulnerability Rating Approach

This section provides the framework of our software vulnerability rating approach (SVRA), where the base score of a vulnerability is dynamically calculated over time.

2.1. Frequency. Let $T \subseteq \mathbb{R}^+$ be the time domain. Obviously, all vulnerabilities have their own report times. Given $t \in T$, let $\mathscr{V}(t)$, a vulnerability database, be all vulnerabilities whose report time is less than or equal to t, and

$$\Omega = \{(\text{AV}, \text{Local}), (\text{AV}, \text{Adj.Net}), (\text{AV}, \text{Network}), \ldots, (\text{AI}, \text{None}), (\text{AI}, \text{Partial}), (\text{AI}, \text{Complete})\}, \quad (4)$$

which contains 18 elements. Then, for $\forall(m, a) \in \Omega$, a subset $\mathscr{V}(m = a, t)$ is defined as

$$\mathscr{V}(m = a, t) = \{v \in \mathscr{V}(t) \mid v.m = a\}. \quad (5)$$

The frequency of $m = a$ at t is denoted as $f(m = a, t)$ and it can be computed as follows:

$$f(m = a, t) = \begin{cases} \dfrac{\text{card}(\mathscr{V}(m = a, t))}{\text{card}(\mathscr{V}(t))} & \text{if } \mathscr{V}(t) \neq \emptyset, \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where card(X) denotes cardinality of set X.

Note that the report time is unique, and it represents the cut-off point at which a vulnerability belongs to the database or not. So, the report time is chosen as the benchmark to rank the vulnerability database. The other time parameters, such as modified date, cannot rank the database effectively (a vulnerability possibly might not have a modified date, for example).

Figure 2 shows the frequency curves of the six basic metrics at time point $t = 2012$ for different values. For the exploitability metrics, the three curves of AV = Network, AC = Low, and AU = None have a higher position in their coordinate systems. This shows that a vulnerability v falling into the group (AV = Network, AC = Low, AU = None) is more vulnerable. Overall, the curves of exploitability metrics are divergent, while the curves of impact metrics are convergent.

Similarly, for $\forall(m_1, a_1), (m_2, a_2) \in \Omega$, and $m_1 \neq m_2$, the frequency of $(m_1 = a_1, m_2 = a_2)$ at time t can be calculated as follows:

$$f(m_1 = a_1, m_2 = a_2, t) = \frac{\text{card}(\mathscr{V}(m_1 = a_1, m_2 = a_2, t))}{\text{card}(\mathscr{V}(t))}, \quad (7)$$

where $\mathscr{V}(m_1 = a_1, m_2 = a_2, t) = \mathscr{V}(m_1 = a_1, t) \cap \mathscr{V}(m_2 = a_2, t)$. If $\mathscr{V}(t) = \emptyset$, $f(m_1 = a_1, m_2 = a_2, t) = 0$.

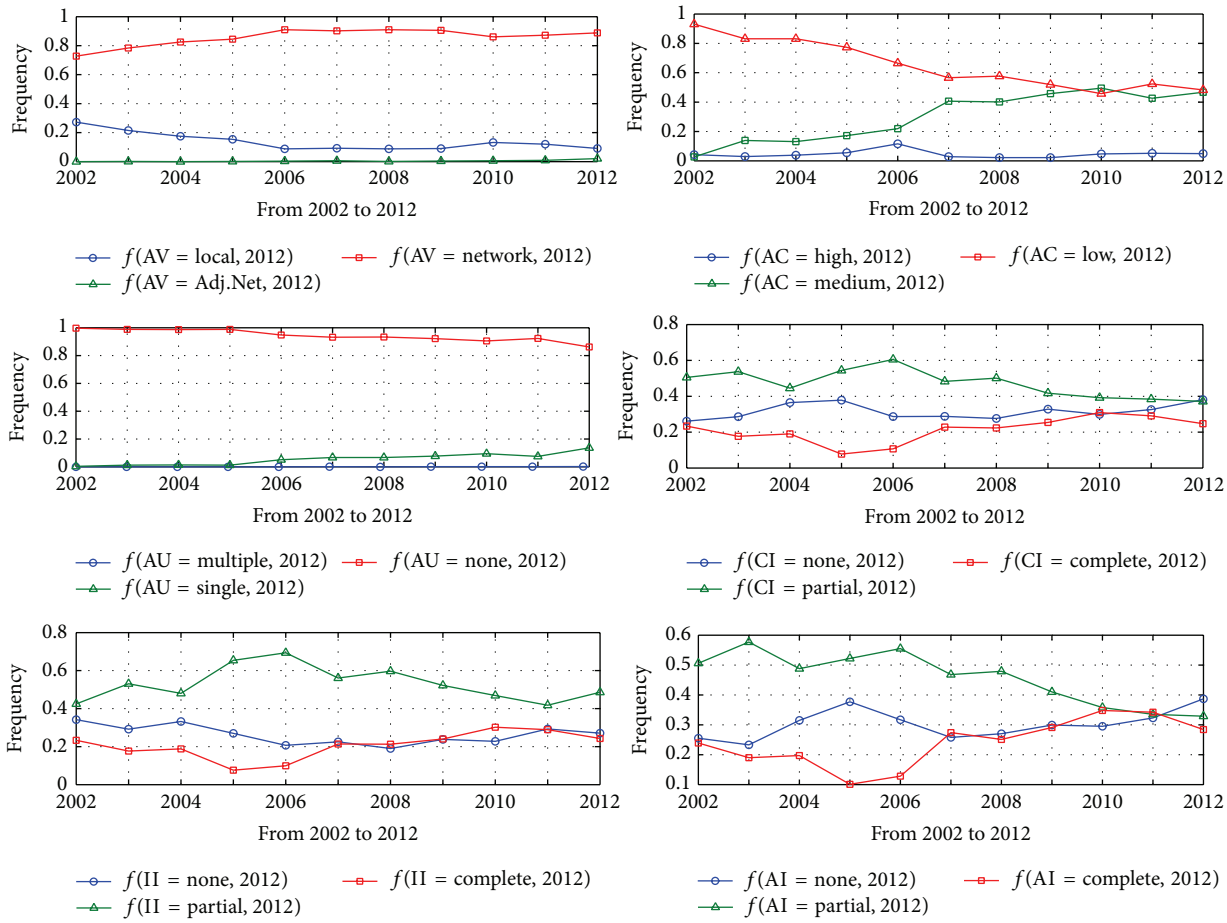

FIGURE 2: The frequency change curves of the six metrics from 2002 to 2012, where $T = \{2002, 2003, \ldots\}$. As can be seen, the curves of AV, AC, and AU are divergent, while the curves of CI, II, and AI have approximate convergence (to the value 1/3). These curves reflect the probability of the occurrence of the metric values of each metric.

2.2. Exploitability Score $E(v,t)$. Consider the correlations among AV, AC, and AU; the equation for exploitability subscore $E(v,t)$ is defined by the following probability:

$$E(v,t) = C(t) \cdot \Pr(v.\mathrm{AV}, v.\mathrm{AC}, v.\mathrm{AU}, t), \tag{8}$$

where $\Pr(\cdot)$ is the probability measure. In contrast to v2, we use probability to define the exploitability score instead of the magic numbers in Table 1. The probability also includes the time point t as its parameter, and it can be given by the root of $n(t)$ solution of its frequency:

$$\Pr(\mathrm{AV}, \mathrm{AC}, \mathrm{AU}, t) = \sqrt[n(t)]{f(\mathrm{AV}, \mathrm{AC}, \mathrm{AU}, t)}. \tag{9}$$

Because the frequencies of the 27 vulnerability types of exploitability metrics are not of the same order of magnitude, $\sqrt[n(t)]{f(\mathrm{AV}, \mathrm{AC}, \mathrm{AU}, t)}$ is used to approximate the probability $\Pr(\cdot)$ instead of $f(\mathrm{AV}, \mathrm{AC}, \mathrm{AU}, t)$. So, there must exist the smallest positive integer n_0 such that

$$\sqrt[n_0]{\min\{f(\mathrm{AV}, \mathrm{AC}, \mathrm{AU}, t) > 0\}} \geq \frac{1}{27}. \tag{10}$$

The basic idea of this equation is that the minimum and nonzero value $\min\{f(\mathrm{AV}, \mathrm{AC}, \mathrm{AU}, t) > 0\}$ is close to 1/27 when the range $[0, 1]$ is divided into 27 classes. Since these subscores are normalized to the range $[0.0, 10.0]$, the coefficient of (8) can be determined by

$$C(t) = \frac{10}{\sqrt[n_0]{\max f(\mathrm{AV}, \mathrm{AC}, \mathrm{AU}, t)}}. \tag{11}$$

For the database $\mathscr{V}(2012)$, the number of each exploitability type is listed in fourth column of Table 4. As can be seen, the value $\min\{f(\mathrm{AV}, \mathrm{AC}, \mathrm{AU}, 2012) > 0\} = 1/54432$ and $\sqrt[4]{1/54432} = 0.0655 > 1/27$, so, $n_0 = 4$. Note that $\max\{f(\mathrm{AV}, \mathrm{AC}, \mathrm{AU}, 2012)\} = 27419/54432$, and

$$C(2012) = \frac{10}{\sqrt[4]{27419/54432}} = 11.87. \tag{12}$$

From (8), the exploitability subscores of SVRA can be calculated; the results are listed in the fifth column of Table 4. CVSS v2 has 9 exploitability vectors with a score of 0, while SVRA has only one. The theoretical distributions of exploitability subscores for both SVRA and CVSS v2 are shown in Figure 3. From a theoretical viewpoint, v2 subscores have much less diversity than SVRA subscores. Figure 4

Table 4: Theoretical exploitability score comparison.

AV	AC	AU	Count[a]	$E(v,t)$	E (v2)
Local	High	Multiple	4	1.1	0
Local	High	Single	30	1.8	1.5
Local	High	None	437	3.6	1.9
Local	Medium	Multiple	2	0.9	0
Local	Medium	Single	83	2.3	2.7
Local	Medium	None	1122	4.5	3.4
Local	Low	Multiple	2	0.9	0
Local	Low	Single	193	2.9	3.1
Local	Low	None	5268	6.6	3.9
Adj.Net.	High	Multiple	1	0.8	0
Adj.Net.	High	Single	4	1.1	2.5
Adj.Net.	High	None	22	1.7	3.2
Adj.Net.	Medium	Multiple	0	0	0
Adj.Net.	Medium	Single	19	1.6	4.4
Adj.Net.	Medium	None	66	2.2	5.5
Adj.Net.	Low	Multiple	1	0.8	0
Adj.Net.	Low	Single	21	1.7	5.1
Adj.Net.	Low	None	124	2.6	6.5
Network	High	Multiple	7	1.3	0
Network	High	Single	213	3.0	3.9
Network	High	None	1884	5.1	4.9
Network	Medium	Multiple	5	1.2	0
Network	Medium	Single	902	4.3	6.8
Network	Medium	None	14889	8.6	8.6
Network	Low	Multiple	6	1.2	0
Network	Low	Single	1708	5.0	8.0
Network	Low	None	27419	10.0	10.0

[a]A total of 54432 vulnerabilities in $\mathscr{V}(2012)$ at $t = 2012$.

Table 5: Values of $n(t)$ and $C(t)$ from 2002 to 2012.

Year	$n(t)$	$C(t)$	Card($\mathscr{V}(t)$)	Mean($E(v,t)$)
2002	3	11.39	6662	3.2
2003	3	11.44	8157	2.8
2004	3	11.45	10796	2.7
2005	3	11.53	15409	2.7
2006	3	11.72	22389	2.6
2007	4	11.42	28823	3.1
2008	4	11.52	35808	3.1
2009	4	11.62	40655	3.1
2010	4	11.73	45516	3.1
2011	4	11.79	49724	3.0
2012	4	11.87	54432	2.9

shows how exploitability subscores change over time for two vulnerability types: (AV = Network, AC = High, AU = Single) and (AV = Local, AC = Low, AU = None). For SVRA, the exploitability subscore may increase, decrease, or remain unchanged. Table 5 lists the $n(t)$ and $C(t)$ values from 2002 to 2012. When the amount of vulnerability data $\mathscr{V}(t)$ increases, the changes for $C(t)$ and $n(t)$ are minor. This

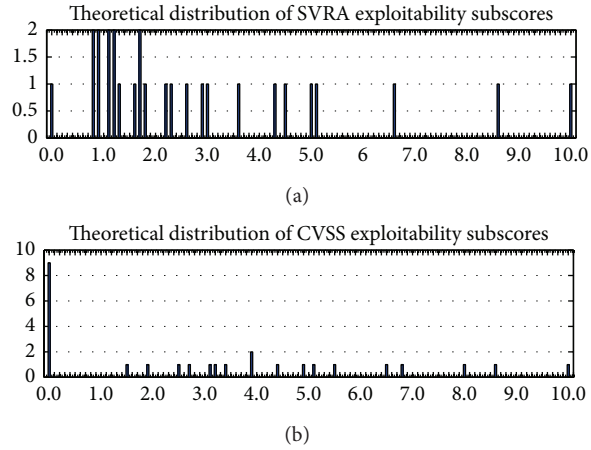

Figure 3: Theoretical distributions of exploitability subscores for SVRA (a) and CVSS v2 (b), where $T = \{2002; 2003, \ldots\}$ and $t = 2012$.

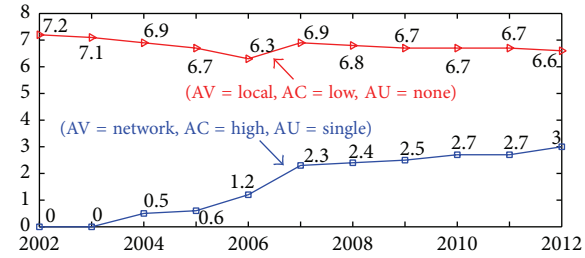

Figure 4: Exploitability subscore curves from 2002 to 2012 for two vulnerability types; the exploitability of a vulnerability changes when $\mathscr{V}(t)$ increases over time.

indicates that (8) has better stability and can dynamically compute $E(v,t)$ when $\mathscr{V}(t)$ changes over time.

2.3. Impact Score $I(v,t)$. The impact caused by a vulnerability varies. Ideally, the sum of all categories of impact can be used to measure the impact subscore $I(v,t)$. However, there exist correlations among the impact metrics CI, II, and AI, so the equation for impact subscore is defined as

$$I(v,t) = D(t) \cdot \left[1 - \prod_{m \in S_i} (1 - \gamma(v.m,t))\right], \quad (13)$$

where $S_i = \{\text{CI, II, AI}\}$ and

$$\gamma(m,t) = \begin{cases} \frac{1}{3} f(m,t) & \text{if } m = \text{None} \\ \frac{1}{3} f(m,t) + 0.167 & \text{if } m = \text{Partial} \\ \frac{1}{3} f(m,t) + 0.333 & \text{if } m = \text{Complete.} \end{cases} \quad (14)$$

From the frequency curves of CI, II, and AI in Figure 2, the approximate convergence value 1/3 is chosen as the

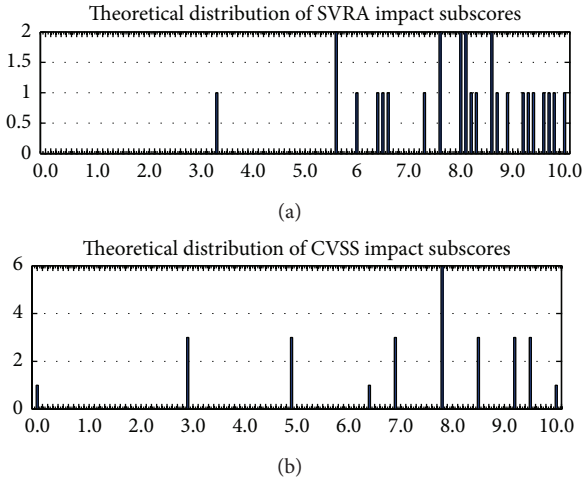

Figure 5: Theoretical distributions of impact subscores for SVRA (a) and CVSS v2 (b), where $T = \{2002, 2003, \ldots\}$ and $t = 2012$.

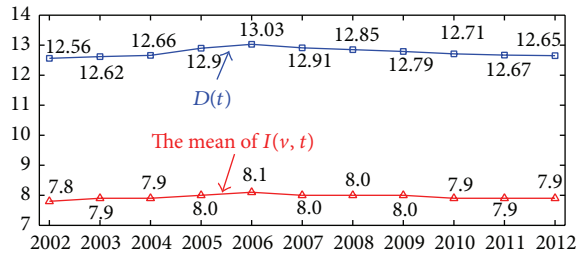

Figure 6: $D(t)$ and mean$(I(v,t))$ curves from 2002 to 2012; (13) has better stability and can dynamically compute $I(v,t)$ when $\mathscr{V}(t)$ changes.

coefficient of $\gamma(m,t)$. Let $\beta(v,t) = 1 - \prod_{m \in S_i}(1 - \gamma(v.m,t))$; this definition makes use of an idea similar to the inclusion-exclusion principle, and the parameter can be determined by the following equation:

$$D(t) = \frac{10}{\max \beta(v,t)}. \tag{15}$$

For database $\mathscr{V}(2012)$, the computing results for $\beta(v,t)$ are listed in the fourth column of Table 6. As can be seen, $\max \beta(v, 2012) = 0.7908$, so $D(2012) = 10/0.7908 = 12.65$. By (13), the impact subscores of SVRA can be calculated, and they are listed in the fifth column of Table 6. CVSS v2 has one impact vector (None, None, None) with a score of 0, while SVRA has none. For SVRA, the mean for the theoretical score is 7.9 and the median is 8.1; the standard deviation is 1.58 and the skew is −0.97. This represents a significant change from v2, which has a mean of 7.0, a median of 7.8, a standard deviation of 2.53, and a skew of −1.11. The theoretical distributions of the impact subscores for both SVRA and CVSS v2 are shown in Figure 5. From a theoretical viewpoint, v2 subscores have much less diversity than SVRA subscores.

Figure 6 shows the curves of the two variables $D(t)$ and mean$(I(v,t))$ from 2002 to 2012. When vulnerability data

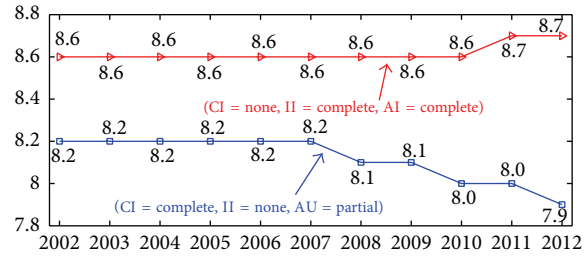

Figure 7: Impact subscore curves from 2002 to 2012 for two vulnerability types; the impact of a vulnerability changes when $\mathscr{V}(t)$ increases over time.

Table 6: Theoretical impact score comparison where $t = 2012$.

CI	II	AI	$\beta(v,t)$[a]	$I(v,t)$	I (v2)
None	None	None	0.2619	3.3	0.0
None	None	Partial	0.4412	5.6	2.9
None	None	Complete	0.5187	6.6	6.9
None	Partial	None	0.4734	6.0	2.9
None	Partial	Partial	0.6013	7.6	4.9
None	Partial	Complete	0.6566	8.3	7.8
None	Complete	None	0.5175	6.5	6.9
None	Complete	Partial	0.6347	8.0	7.8
None	Complete	Complete	0.6854	8.7	9.2
Partial	None	None	0.4453	5.6	2.9
Partial	None	Partial	0.5800	7.3	4.9
Partial	None	Complete	0.6382	8.1	7.8
Partial	Partial	None	0.6042	7.6	4.9
Partial	Partial	Partial	0.7004	8.9	6.4
Partial	Partial	Complete	0.7419	9.4	8.5
Partial	Complete	None	0.6374	8.1	7.8
Partial	Complete	Partial	0.7255	9.2	8.5
Partial	Complete	Complete	0.7635	9.7	9.5
Complete	None	None	0.5093	6.4	6.9
Complete	None	Partial	0.6285	8.0	7.8
Complete	None	Complete	0.6800	8.6	9.2
Complete	Partial	None	0.6499	8.2	7.8
Complete	Partial	Partial	0.7350	9.3	8.5
Complete	Partial	Complete	0.7717	9.8	9.5
Complete	Complete	None	0.6792	8.6	9.2
Complete	Complete	Partial	0.7572	9.6	9.5
Complete	Complete	Complete	0.7908	10.0	10.0

[a] $\beta(v,t) = 1 - \prod_{m \in S_i}(1 - \gamma(v.m,t))$.

$\mathscr{V}(t)$ increases, the changes for $D(t)$ and mean$(I(v,t))$ are minor. This indicates that (13) has better stability and can dynamically compute $I(v,t)$ when $\mathscr{V}(t)$ changes over time. Figure 7 shows how impact subscores change over time for two impact types: (CI = None, II = Complete, AI = Complete) and (CI = Complete, II = None, AU = Partial). For SVRA, the impact subscore can increase, decrease, or remain unchanged when $\mathscr{V}(t)$ changes over time.

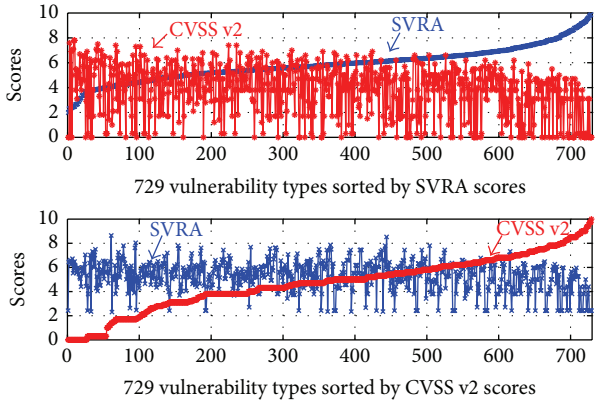

FIGURE 8: A theoretical score comparison of SVRA with CVSS v2 for 729 vulnerability types, where $T = \{2002, 2003, \ldots\}$ and $t = 2012$.

2.4. Base Score $B(v,t)$. Given time $t \in T$ and a vulnerability $v \in \mathscr{V}(t)$, the new equation of base score is as follows:

$$B(v,t) = \begin{cases} \dfrac{3I(v,t)}{5} + \dfrac{2E(v,t)}{5} & \text{if } I(v,t) \neq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

Figure 8 shows a comparison between SVRA and CVSS v2. Some vulnerability types have high SVRA scores but low v2 scores, and others have high v2 scores but low SVRA scores. We examined the theoretical distributions of SVRA and CVSS v2 scores; see Figure 9. For SVRA, the mean for the theoretical scores is 5.9, the median is 5.8, the standard deviation is 1.34, and the skew is 0.22. This represents a significant change from v2, which has a mean of 4.7, a median of 4.9, a standard deviation of 2.20, and a skew of −0.28. This illustrates that SVRA has superior numerical normality and stability. Figure 10 illustrates the changing trends of two vulnerability types (Type 1 and Type 2), as well as the mean of the SVRA base scores from 2002 to 2012. The CVSS v2 base scores of Type 1 and Type 2 are 6.6 and 5.6, respectively. However, the SVRA base score of Type 1 first decreases and then settles in the range [7.7, 7.9]. The $B(v,t)$ of Type 2 increases by 1.1 from 4.9 to 6.0. At first, the rise of the mean mean$(B(v,t))$ is rapid, and then the increase slows down. So, when new vulnerabilities are added into the database $\mathscr{V}(t)$, many of $\mathscr{V}(t)$ become more and more serious. In real life, two or more vulnerabilities may be combined to form a critical issue. The Google Chrome Pwnium full exploits are excellent examples, in which strings of vulnerabilities are combined into a full sandbox escape, resulting in arbitrary code execution. So, these curves reflect the fact that vulnerabilities interact with each other.

3. Experimental Analysis and Comparisons

This section describes our experimental analysis of the SVRA and CVSS v2 base scores for 54,432 vulnerabilities listed in CVE. Figure 11 shows a comparison between SVRA and CVSS v2. For the v2 experimental scores, the mean is 6.3, the median is 6.8, the standard deviation is 2.02, and the skew

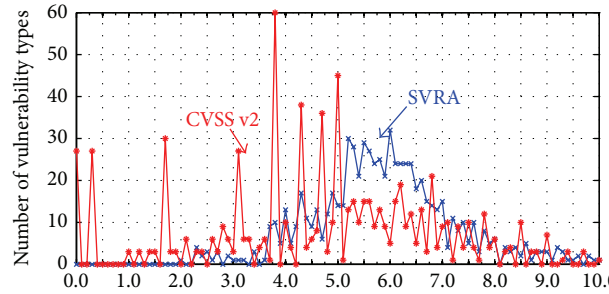

FIGURE 9: Theoretical distributions of SVRA and CVSS v2, where $T = \{2002, 2003, \ldots\}$ and $t = 2012$.

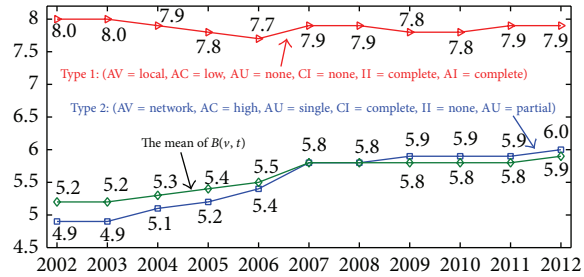

FIGURE 10: The base score curves from 2002 to 2012 for two vulnerability types and the mean of $B(v,t)$. This indicates the base score of a vulnerability changes when $\mathscr{V}(t)$ increases over time, where $T = \{2002, 2003, \ldots\}$.

is −0.001. This represents an increase of 1.6 in the mean and 1.9 in the median from the theoretical data. Approximately 58.43% of the scores are above 5.0, 16.58% are at 5.0, and 24.98% are below 5.0. For the SVRA experimental scores, the mean is 8.1 and the median is 8.0. This represents an increase of 2.2 in both the mean and median from the theoretical data. The standard deviation is 1.26 and the skew is −0.39. Of the scores, approximately 99.09% are above 5.0, 0.21% are at 5.0, and 0.69% are below 5.0. This is consistent with the CVSS-SIG's goal to have the majority of scores above 5.0.

The national vulnerability database (NVD) [4] generates a base score for each vulnerability and then assigns a ranking based on the score. The rankings are Low (0.0 to 3.9), Medium (4.0 to 6.9), and High (7.0 to 10.0) [10]. The motivation for having these rankings is to help organizations prioritize their mitigations of new vulnerabilities. Table 7 lists the comparison results of several vulnerabilities among the four authoritative security organizations (Secunia in Denmark, FrSIRT in France, ISS X-Force in the USA, and CVSS). As can be seen, SVRA coincides with the majority of the organizations. For the vulnerabilities CVE-2007-1497 and CVE-2007-2242, SVRA adjusts the CVSS rankings from High to Medium.

We also performed rankings for the theoretical data, as shown in Table 8. There exists a dramatic change in the CVSS v2 and SVRA, with SVRA having more Medium and High vulnerabilities but fewer Low vulnerabilities.

There are times when a vulnerability is scored as a 0.0 by CVSS v2 standards. These are often vulnerabilities that do

Table 7: Comparison of rating results.

Example	AV	AC	AU	CI	II	AI	Secunia	FrSIRT	X-Force	CVSS v2	SVRA
CVE-2007-1497	N	L	N	P	N	N	Medium (3/5)	Medium (2/4)	Medium (2/3)	High (3/3)	Medium (2/3)
CVE-2007-1754	N	M	N	C	C	C	High (4/5)	High (3/4)	High (3/3)	High (3/3)	High (3/3)
CVE-2007-1748	N	L	N	C	C	C	High (4/5)	Critical (4/4)	High (3/3)	High (3/3)	High (3/3)
CVE-2007-3338	N	L	N	C	C	C	Medium (3/5)	Critical (4/4)	High (3/3)	High (3/3)	High (3/3)
CVE-2007-3680	L	L	N	C	C	C	Low (2/5)	Medium (2/4)	High (3/3)	High (3/3)	High (3/3)
CVE-2007-2242	N	L	N	N	N	C	Medium (3/5)	Medium (2/4)	Low (1/3)	High (3/3)	Medium (2/3)

[a]"(m/n)" denotes the mth level of n severity levels; the bigger the m of a vulnerability, the higher severity level ranking of the vulnerability.

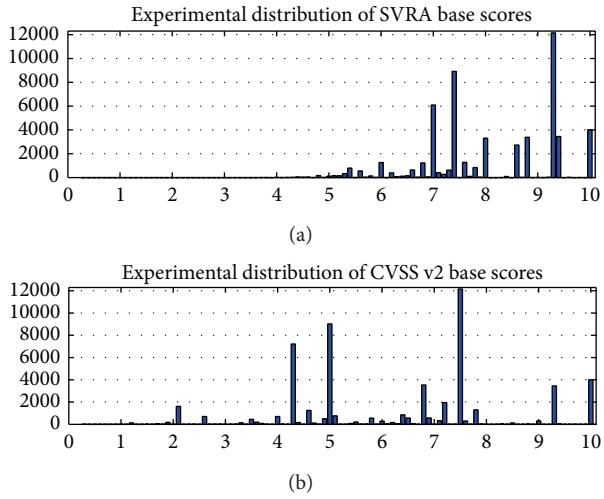

Figure 11: Experimental SVRA scores (a) and CVSS v2 scores (b), where $T = \{2002, 2003, \ldots\}$ and $t = 2012$.

Table 8: NVD severity rankings for theoretical data.

Rank	CVSS v2		SVRA	
	Count	Frequency	Count	Frequency
Low	251	34.43%	51	7.0%
Medium	370	50.75%	556	76.27%
High	108	14.82%	122	16.73%

pose some threat, albeit a limited one. Nevertheless, if the issue is considered a vulnerability by the industry, this should be reflected through the assignment of a real score. One such example is arbitrary site redirection. Per current CVSS v2 scoring rules, this would yield (AV = Network, AC = Medium, AU = None, CI = None, II = None, AI = None) with an SVRA score of 5.4.

4. Conclusions

The CVSS empirical values given by CVSS-SIG cannot distinguish software vulnerabilities that have identical scores but different severities. In this paper, a software vulnerability rating approach (SVRA) is proposed based on a vulnerability database. With the SVRA, the frequencies of CVSS metrics are analyzed at different times. The equations for both exploitability and impact subscores are given in terms of these frequencies. To create an SVRA score, SVRA performs a weighted average of these two subscores. As the frequency changes over time, each metric takes different values instead of the constant empirical value. The score of a vulnerability is dynamically computed at different time points using the vulnerability database. The theoretical and experimental results illustrate the efficiency of the SVRA.

Although the SVRA was developed for the base metric group, the approach can be extended to the temporal metric group and the environmental metric group. Further work will include predicting whether vulnerability severity changes so much over time that future modifications to the SVRA may be needed.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Funds NSFC61171121 and the Science Foundation of Chinese Ministry of Education—China Mobile 2012.

References

[1] M. Bozorgi, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond heuristics: learning to classify vulnerabilities and predict exploits," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '10)*, pp. 105–113, July 2010.

[2] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using Bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 61–74, 2012.

[3] Forum of Incident Response and Security Teams, "CVSS Adopters," http://www.first.org/cvss/eadopters.html.

[4] National Institute of Standards and Technology, "National vulnerability database," http://nvd.nist.gov/.

[5] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system (CVSS)," 2011, http://www.first.org/cvss/cvss-guide.html.

[6] K. Scarfone and P. Mell, "An analysis of CVSS version 2 vulnerability scoring," in *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM '09)*, pp. 516–525, October 2009.

[7] National Infrastructure Advisory Council, "Common vulnerability scoring system," 2004, http://www.first.org/cvss/cvss-dhs-12-02-04.pdf.

[8] G. Reid, P. Mell, and K. Scarfone, "Cvss-sig version 2 history," Forum of Incident Response and Security Teams, June 2009, http://www.first.org/cvss/history.html.

[9] MITRE Corporation, "Common vulnerabilities and exposures (cve)," August 2009, http://cve.mitre.org/.

[10] National Institute of Standards and Technology, "National vulnerability database cvss scoring," August 2009, http://nvd.nist.gov/cvss.cfm.