An Analysis of CVSS Version 2 Vulnerability Scoring[1]

Karen Scarfone
National Institute of Standards and Technology (NIST)
karen.scarfone@nist.gov

Peter Mell
National Institute of Standards and Technology (NIST)
mell@nist.gov

Abstract

The Common Vulnerability Scoring System (CVSS) is a specification for measuring the relative severity of software vulnerabilities. Finalized in 2007, CVSS version 2 was designed to address deficiencies found during analysis and use of the original CVSS version. This paper analyzes how effectively CVSS version 2 addresses these deficiencies and what new deficiencies it may have. This analysis is based primarily on an experiment that applied both version 1 and version 2 scoring to a large set of recent vulnerabilities. Theoretical characteristics of version 1 and version 2 scores were also examined. The results show that the goals for the changes were met, but that some changes had a negligible effect on scoring while complicating the scoring process. The changes also had unintended effects on organizations that prioritize vulnerability remediation based primarily on CVSS scores.

1. Introduction

The Common Vulnerability Scoring System (CVSS) is a specification for documenting the major characteristics of vulnerabilities and measuring the potential impact of vulnerability exploitation [3]. The motivation for developing CVSS was to provide standardized information for organizations to use to prioritize vulnerability mitigation. CVSS is 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).

CVSS has been widely adopted by the information technology community. CVSS is mandated for use in evaluating the security of payment card systems worldwide [9]. The U.S. Federal government uses it for its National Vulnerability Database [7] and mandates its use by products in the Security Content Automation Protocol (SCAP) validation program [6]. CVSS has also been adopted by dozens of software vendors and service providers [1].

There are many proprietary schemes for scoring software flaw vulnerabilities, most created by software vendors, but CVSS is the only known open specification. CVSS is also distinguished from other scoring systems in that CVSS was designed to be quantitative so that analysts would not have to perform qualitative evaluations of vulnerability severity. Significant effort has been put into developing the specification for CVSS so that any two vulnerability analysts should produce identical CVSS scores for the same vulnerability. In addition, CVSS is designed to provide visibility into how a score was calculated. Each CVSS score is provided with a CVSS vector. This vector includes metrics that categorize several characteristics of a vulnerability. The vector provides details on the nature of the vulnerability that help CVSS users to understand why a vulnerability received a particular score. These two attributes of CVSS, quantitative analysis and transparency through vectors, lend the specification to research and analysis. Large publicly available CVSS data sets from the National Vulnerability Database [7] further enable this research.

The initial CVSS specification was developed by the National Infrastructure Advisory Council [5] and published in October 2004. As [10] explains, the original specification did not undergo widespread peer review, and adopters raised several concerns about it. The CVSS-SIG worked from April 2005 to June 2007 on identifying problems with version 1 and determining how best to solve them, which ultimately led to the release of version 2. The goal for this paper is to determine how effectively version 2 (v2) has addressed the version 1 (v1) problems. The analysis is based on a review of the v2 specification and the results of an experiment scoring 11,012 recent vulnerabilities using both v1 and v2. Section 2 provides background on CVSS, and Section 3 discusses the v1 problems and the methodology used to

[1] Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States.

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

create v2. Section 4 provides an overview of the analysis process, and Sections 5 through 8 present the results of the analysis in four categories: base scores, subscores, vulnerability characteristics, and severity rankings. Section 9 provides conclusions for the work.

2. Background

CVSS uses three groups of metrics to calculate vulnerability scores. Base metrics are vulnerability attributes that are constant over time and across all implementations and environments. Temporal metrics are vulnerability attributes that change over time but which apply to all instances of a vulnerability in all environments (e.g., the public availability of 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. Environmental metrics are vulnerability attributes that are organization and implementation-specific, such as how prevalent a target is within an organization. An environmental score is calculated with an equation that uses both the temporal score and the environmental metric values as parameters.

The focus of our research is base metrics. An equation is applied to their values to calculate a vulnerability's base score. There are six base metrics in CVSS v2. The first three metrics relate to exploitability. AccessVector measures the range of exploitation (e.g., can it be launched over the network or only locally). Authentication measures the level to which an attacker must authenticate to the target before exploiting the vulnerability. AccessComplexity measures how difficult it is to exploit the vulnerability once the target is accessed. These three metrics, which collectively measure how readily an attacker can attempt to exploit a vulnerability, comprise an exploitability subvector from which an exploitability subscore can be calculated.

In addition to the three exploitability metrics, v2 also has three base metrics related to impact. ConfImpact measures the level to which vulnerability exploitation can impact the target's confidentiality, and IntegImpact and AvailImpact capture the same information for integrity and availability, respectively. The impact metrics collectively measure the extent to which an attacker can compromise a computer's security by exploiting a particular vulnerability. The three impact metrics form the impact subvector, from which an impact subscore can be determined.

Table 1 lists the possible values for each metric in CVSS v1 and v2, along with the abbreviations (in parentheses) for each metric and metric value. CVSS v1 had three additional base metrics, called impact bias metrics, that set the relative importance of the three impact metrics (ConfImpact, IntegImpact, and AvailImpact). The impact bias metrics were converted from base metrics to environmental metrics in v2.

The equation for calculating the base score in v1 is round_to_1_decimal (10 * AccessVector * AccessComplexity * Authentication * ((ConfImpact * ConfImpactBias) + (IntegImpact * IntegImpactBias) + (AvailImpact * AvailImpactBias))). The base score ranges between 0.0 and 10.0. To calculate the v2 base score, the three exploitabilty metrics are combined into an exploitability subscore using the equation (20 * AccessVector * AccessComplexity * Authentication). The three impact metrics are combined into an impact subscore using the equation (10.41 * (1 – (1 – ConfImpact) * (1 – IntegImpact) * (1 – AvailImpact))). The base score is calculated from the subscores using the following equation: (round_to_1_decimal(((0.6 * Impact) + (0.4 * Exploitability) – 1.5) * f(Impact))), where f(impact) = 0 if Impact=0, 1.176 otherwise.

Table 1. Possible values for base metrics

Metric Name	Possible Values
AccessVector (AV), v1	Remotely (R): 1.0 Requires local authentication or physical access (L): 0.7
AccessVector (AV), v2	Network (N): 1.0 Adjacent network (A): 0.646 Requires local access (L): 0.395
AccessComplex-ity (AC), v1	Low (L): 1.0 High (H): 0.8
AccessComplex-ity (AC), v2	Low (L): 0.71 Medium (M): 0.61 High (H): 0.35
Authentication (Au), v1	Not required (NR): 1.0 Required (R): 0.6
Authentication (Au), v2	Not required (N): 0.704 Single instance (S): 0.56 Multiple instances (M): 0.45
ConfImpact (C), IntegImpact (I), AvailImpact (A), v1	Complete (C): 1.0 Partial (P): 0.7 None (N): 0.0
ConfImpact (C), IntegImpact (I), AvailImpact (A), v2	Complete (C): 0.660 Partial (P): 0.275 None (N): 0.0

3. CVSS v2 design methodology

The designers of CVSS v1 postulated an equation that appeared reasonable and then assigned metric values for the equation elements using trial and error.

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

This resulted in a useful scoring specification, but adopters of v1 noted several deficiencies [2, 10]. These included base scores that were not properly reflecting the true severity of vulnerabilities, and less diversity in scores than expected (i.e., too many vulnerabilities having the same score). To address this, CVSS v2 was to correct the errant scores, which would also cause a higher average value for scores, and to improve score accuracy and diversity by making the metrics more granular and adjusting the metric values and equations. (For CVSS, "accuracy" refers to relative accuracy. CVSS scores are intended to provide a relative comparison of vulnerability severity, not exact measurements.) Another goal was to ensure that analysts would produce consistent and accurate v2 scores, so CVSS should not be made more complicated than necessary.

CVSS v2 was designed using a more rigorous process than v1, which did not undergo extensive peer review. The first step in designing v2 was to evaluate the v1 metrics and propose changes that would enable users to better characterize the security-relevant aspects of a vulnerability. The impact bias metrics were removed from the base metric (and moved to the environmental metric), and more granularity was added to the AccessVector, AccessComplexity, and Authentication metrics.

Once the v2 metrics were defined, opinions were gathered from the CVSS-SIG members and their organizations on what score each type of vulnerability should have. There were six v2 metrics, each of which had three possible values, resulting in 729 possible vulnerability types. It was not possible to rank, much less score, 729 vulnerability types in a justifiable manner. The problem was simplified by placing the base metrics into two groups, impact and exploitability, and generating approximated subscores for each group. Each group contained three metrics with three possible values, so only 27 vulnerability types per group had to be ranked and scored. The CVSS-SIG members reached consensus on the approximated rankings and scorings, resulting in the creation of lookup tables for exploitability and impact. To create a CVSS score from these two subscores, the researchers performed a weighted average of exploitability and impact, with exploitability having a weight of 0.4 and impact having a weight of 0.6. These weights were a simplified version of the weights effectively employed by v1, 0.428 and 0.572 [10].

At this point the CVSS-SIG could have used the lookup tables for v2 scoring. However, the CVSS community desired an equation instead of lookup tables. Therefore, mathematicians proposed equations that approximated the lookup tables. The resulting equations underwent beta testing and a number of small scoring inconsistencies were encountered. To address these, modifications were made to particular metric input values, and then further beta testing was performed to ensure that the scores were as expected. The final change was to the equation itself; experts felt that the entire scoring distribution had been shifted too high. To lessen this, the researchers subtracted 1.5 from the base score and then multiplied the result by 1.176. This caused the desired shift downwards while maintaining the score range of 0.0 to 10.0 and keeping the scores for the different types of vulnerabilities in the same order. This concluded the design of v2, and it was finalized in June 2007.

CVSS v2 has already been determined to meet the accuracy goals of the CVSS-SIG, because the CVSS-SIG extensively examined many test cases when designing v2 to confirm score accuracy. There was also a small experiment conducted [2] during v2's development that involved calculating scores for 1156 vulnerabilities using both v1 and a pre-final version of v2. That experiment focused on analyzing the average of scores and score diversity, and it found improvements in both from v1. This paper has similar goals as the [2] experiment, but it uses the final version of v2, it examines a much larger data set, and it performs a far more detailed and thorough analysis of the experimental data.

4. Analysis overview

We performed a theoretical analysis of the CVSS v2 base score equation and metrics. We generated theoretical scoring distributions for v1 and v2 by considering all the possible sets of metric values and calculating the corresponding scores and the frequency of each score. First, we counted the number of possible combinations of metric values: for v1 there are 864, and for v2 there are 729. However, vulnerabilities with all impact metrics set to None are not possible in practice because each vulnerability must have some impact, so we subtracted those and had final counts of 832 for v1 and 702 for v2. Next, we calculated the score for each combination and counted the frequencies of each of the 101 possible score values (0.0-10.0). We then used this as the basis for analyzing the theoretical scoring distribution. We also generated theoretical scoring distributions for the impact and exploitability subscores using a similar process.

We also performed an experimental analysis of CVSS scores. In the experiment, we calculated v2 base scores for 11,012 vulnerabilities listed in the Common Vulnerabilities and Exposures (CVE) dictionary [4]. This encompassed all valid CVE entries published between June 20, 2007 and April 30, 2009. The scoring

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

was performed by the National Vulnerability Database (NVD) [7] in accordance with the v2 specification [3].

For the experiment, we mapped the v2 metrics assigned to each CVE entry back to their v1 equivalents. Three base metrics had no changes in options, so no mapping was needed. Three other base metrics had more granular options in v2 that could be mapped to broader v1 options. The mappings are shown in Table 6. The impact bias metrics from v1 were dropped from the base metrics for v2, and the [2] study indicated that in v1 they affected scoring less than 1% of the time, so we chose to disregard them.

A small percentage of the experimental data is assumed to have scoring errors. Errors can occur from research sources, such as incorrect or incomplete information in vulnerability announcements, or analyst misinterpretation of vulnerability information. Errors can also occur by analysts misunderstanding the CVSS scoring guidelines or having differing assumptions, such as the default privileges under which a vulnerable application is typically run. One of the CVSS-SIG's goals in developing v2 was to make the scoring process clearer for analysts to improve score consistency [10]. However, the scoring process is sufficiently complex that some misinterpretations likely still occur and cause occasional scoring discrepancies. The true error rate in the experimental data cannot be readily quantified because there is no authoritative source of CVSS scores, but there are extensive quality assurance efforts in place, with the analysts checking each others' work and the researchers providing guidance whenever the analysts are unsure of the proper scoring. The analysts are knowledgeable about general security and have been specifically trained on vulnerability characteristics and CVSS scoring. Also, the scoring interface does not have default settings, so there should not be a default bias. The error rate should be sufficiently small so as not to affect the results of this experiment.

5. Base score analysis

This section describes our theoretical and experimental analysis of the v1 and v2 base scores.

5.1. Theoretical score distribution

We examined the theoretical distributions of v1 and v2 scores. For v2, the mean for the theoretical scores is 5.4, the median is 5.6, the standard deviation is 1.82, and the skew is -0.34. This is a significant change from v1, which had a mean of 3.6, a median of 3.3, a standard deviation of 1.91, and a skew of 0.81. This shift from v1's characteristics is consistent with the CVSS-SIG's goal to have higher scores, with the majority of scores being over 5.0. Figures 1 and 2 show the frequency of each possible score in the theoretical distributions for v1 and v2 scores, respectively.

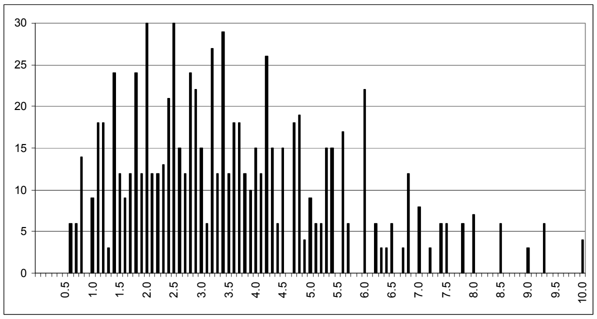

Figure 1. Theoretical distribution of v1 scores

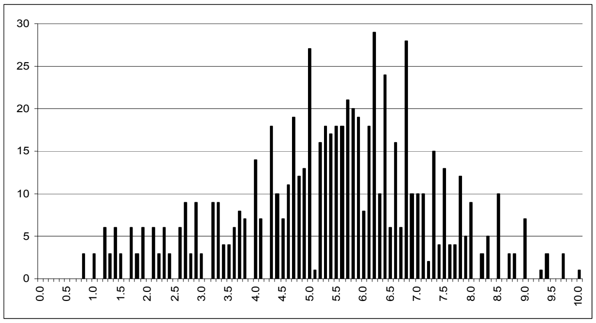

Figure 2. Theoretical distribution of v2 scores

We also mapped all the possible v2 vectors back to their v1 counterparts and looked at the score differences for each of the 702 vectors. This is different from the v1 mean and median described above, which were based on the 832 possible vectors for v1: this is a strict one-to-one comparison of the v1 and v2 scores for all the v2 vectors. From v1 to v2, scores increased an average of 2.1, with a median change of +2.3 and a standard deviation of 1.23. Of the 702 vectors, 664 (94.6%) had higher v2 scores, 31 (4.4%) had higher v1 scores, and 7 (1.0%) had the same v1 and v2 scores. This indicates that v2 should generally produce higher scores than v1.

5.2. Experimental score distribution

We analyzed the base scores for the experimental data. Figure 3 shows how many vulnerabilities had each possible v1 score. The mean was 5.1 and the median 5.6. This was an increase of 1.5 in the mean and 2.3 in the median from the theoretical data. The standard deviation was 2.62 and the skew 0.11.

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

Approximately 45% of the scores were below 5.0 and the other 55% above 5.0, with none at exactly 5.0.

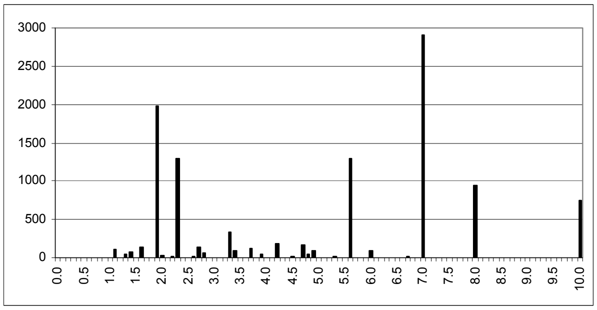

Figure 3. Experimental v1 scores

For the v2 experimental scores, shown in Figure 4, the mean was 6.6, the median 6.8, the standard deviation 1.91, and the skew -0.05. This was an increase of 1.2 in the mean and 1.2 in the median from the theoretical data. Of the scores, approximately 25% were below 5.0, 10% were at 5.0, and 65% were above 5.0. This is consistent with the CVSS-SIG's goal to have the majority of scores above 5.0.

Both the v1 and v2 results show that their experimental scores are significantly higher than their theoretical scores, with the differences being more pronounced for v1.

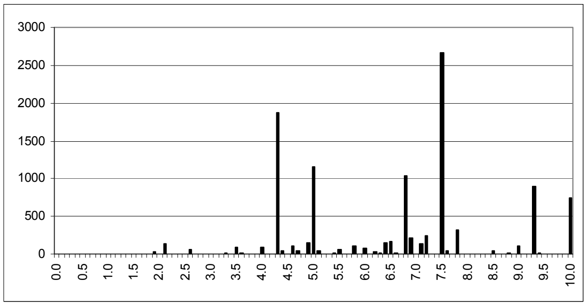

Figure 4. Experimental v2 scores

The means and medians indicate that actual v2 scores are significantly higher than v1 scores. To investigate this further, we compared each v2 experimental score to the score achieved by mapping the v2 vector back to v1. Of the 11,012 vulnerabilities, 10,072 (91.5%) had higher v2 scores, 743 (6.7%) had the same v1 and v2 scores, and 197 (1.8%) had higher v1 scores. This is further confirmation that v2 has met the CVSS-SIG's goal of increasing base scores.

5.3. Score diversity

Score diversity refers to the relative variety of scores. To look at score diversity, we started by reviewing theoretical v1 and v2 scores for all the possible vectors. There are more possible vectors than scores, but this does not necessarily indicate that every score has a corresponding vector. We confirmed the finding from [2] that not all base scores can occur: in v1, 66 of the 101 scores are possible, and in v2, 75 of the 101 scores are possible. Having more scores possible in v2 than v1 helps to support greater score diversity, although it does not ensure it.

We also looked at the diversity of the experimental data. The 11,012 vulnerabilities in the data set produced 35 distinct v1 scores (53% of the 66 possible scores) and 51 distinct v2 scores (68% of the 75 possible scores), again showing the increased diversity of v2 over v1. We also looked to see how diverse the vectors in the experimental data were, and of the 702 possible v2 base vectors, only 143 (20%) were represented. The 10 most common vectors, listed in Table 2, comprised over 77% of all vulnerabilities. Table 2 also shows the v1 and v2 scores for the most common vectors.

Table 2. Most common v2 vectors in experiment

Freq count and %	AV/AC/Au	C/I/A	v1	v2
2662 (24.2)	N L N	P P P	7.0	7.5
1527 (13.9)	N M N	N P N	1.9	4.3
999 (9.1)	N M N	P P P	5.6	6.8
896 (8.1)	N M N	C C C	8.0	9.3
743 (6.7)	N L N	C C C	10.0	10.0
577 (5.2)	N L N	P N N	2.3	5.0
443 (4.0)	N L N	N N P	2.3	5.0
251 (2.3)	L L N	C C C	7.0	7.2
240 (2.2)	N L N	N N C	3.3	7.8
217 (2.0)	L M N	C C C	5.6	6.9

These results differ significantly from the theoretical score distribution, and an analysis of similar results from v1 experimental data in 2006 [2] had found this to be caused by certain types of vulnerabilities occurring much more often than others. Analysis of our experimental data, as shown in Table 2, reaches the same conclusion. Also, because CVSS treats confidentiality, integrity, and availability as equally important, vectors that are identical except for which of these attributes are impacted have the same base scores. For example, vectors 6 and 7 in Table 2 are the same except that one has a partial impact to confidentiality and the other a partial impact to availability.

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

Table 3 presents the ten most commonly occurring scores in the v1 and v2 data. The two most frequent scores encompassed 44.5% of v1 vulnerabilities and 41.2% for v2. The ten most frequently occurring scores encompassed 90.1% of v1 vulnerabilities and 83.8% of v2 vulnerabilities. These are additional indications of the improved diversity of v2 scores over v1.

Table 3. Most common scores in experiment

v1 score	v1 freq count and %	v2 score	v2 freq count and %
7.0	2916 (26.5)	7.5	2662 (24.2)
1.9	1979 (18.0)	4.3	1872 (17.0)
2.3	1293 (11.7)	5.0	1153 (10.5)
5.6	1291 (11.7)	6.8	1038 (9.4)
8.0	948 (8.6)	9.3	896 (8.1)
10.0	745 (6.8)	10.0	743 (6.7)
3.3	331 (3.0)	7.8	321 (2.9)
4.2	183 (1.7)	7.2	251 (2.3)
4.7	163 (1.5)	6.9	217 (2.0)
2.7	140 (1.3)	6.5	167 (1.5)

Most of the scores that appear in Table 3 also appear in Table 2, and their frequencies are similar. For example, in Table 2 the first vector has a v2 score of 7.5 and occurs 2662 times, and in Table 3 the most common v2 score is 7.5 and it also occurs 2662 times. So every instance of a 7.5 score in v2 has the same vector. This particular vector corresponds to vulnerabilities that can be exploited remotely, with low complexity and no authentication. The impact of exploiting this vector is a partial impact to confidentiality, integrity, and availability, which in most cases means that the attacker can gain user-level access. The next most common vector involves a partial impact to integrity through network access, medium attack complexity, and no authentication. This most often corresponds to cross-site scripting vulnerabilities, which have been quite prevalent the past few years.

6. Subscore analysis

To better understand the composition of the v1 and v2 scores, we performed theoretical and experimental analysis of the v2 impact and exploitability subscores.

6.1. Theoretical score distribution

There are 27 possible exploitability vectors, which map to 23 exploitability subscores. There are 26 possible impact vectors, but they only map to 9 impact subscores. So from a theoretical viewpoint, impact subscores have much less diversity than exploitability subscores. Figure 5 shows the frequency of each possible score in the theoretical distribution for the impact vectors. Approximately 23% of the impact vectors had subscores below 5.0. The mean was 7.3 and the median 7.8, the range was 2.9 to 10.0, the standard deviation was 2.12, and the skew was -0.88, indicating that most of the impact subscores are high values. Since an impact subscore is 60% of a base score, this is likely why the theoretical base scores have higher-than-expected values.

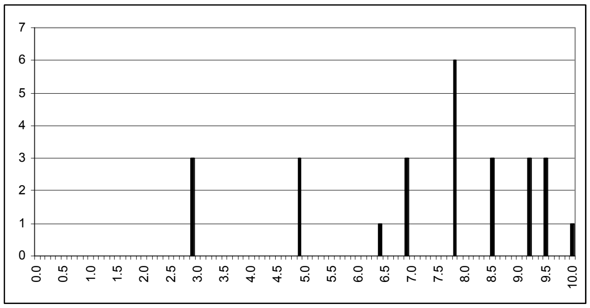

Figure 5. Theoretical distribution of v2 impact subscores

Next, we analyzed the theoretical distribution of exploitation subscores. As shown in Figure 6, two-thirds of the exploitation vectors had subscores below 5.0. The mean for the subscores was 4.3 and the median 3.9, the range was 1.2 to 10.0, the standard deviation was 2.20, and the skew was 0.83. This indicates that the exploitation subscores are somewhat low values, although not as far from the midpoint as the impact subscores were.

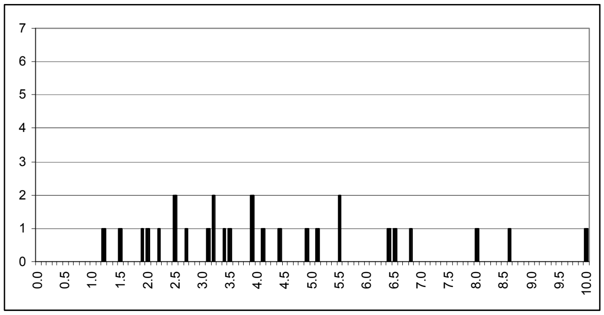

Figure 6. Theoretical distribution of v2 exploitation subscores

6.2. Experimental score distribution

We analyzed the subscores from the experimental data to gain a better understanding of the differences between the theoretical and experimental scores. For

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

the impact subscores, the mean was 6.1 and the median 6.4, the range 2.9 to 10.0, the standard deviation 2.57, and the skew 0.18. For the exploitability subscores, the mean was 8.6 and the median 8.6, the range 1.5 to 10.0, the standard deviation 1.94, and the skew -1.75. These results differed substantially from the theoretical analysis, which indicated means of 7.3 for impact subscores and 4.3 for exploitation subscores. As with the base scores, we also looked at the subscores based on an assumption of an ideal mean of 5. For the impact subscores, 35% were below 5 and 65% were above 5; for the exploitability subscores, 12% were below 5 and 88% were above 5. Figures 7 and 8 show the experimental distribution for impact and exploitation subscores, respectively.

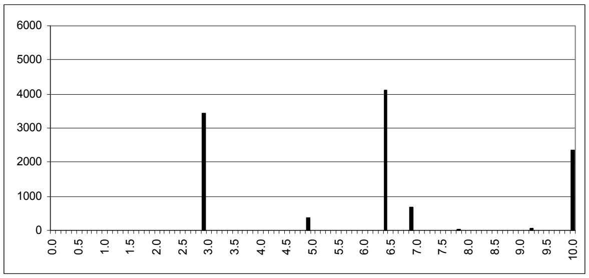

Figure 7. Experimental distribution of v2 impact subscores

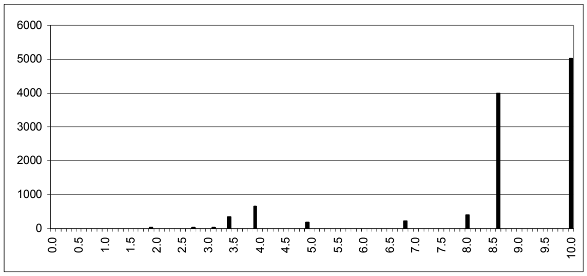

Figure 8. Experimental distribution of v2 exploitation subscores

The deviations from the theoretical distributions indicate that some subvectors are occurring more often than others. To further investigate this, we looked at the most common subvectors and their scores.

6.3. Experimental subscore diversity

We examined the diversity of the experimental subscores. The 11,012 vulnerabilities in the data set produced 21 distinct exploitability subvectors (of 27 possible). The two most common exploitability subvectors comprised 82% of the vulnerabilities, and the ten most common (shown in Table 4) comprised over 99%. The two most common vectors have high scores, 10.0 and 8.6. Since these comprise over 82% of the vulnerabilities, the prevalence of these two vectors is likely the main cause of the experimental subscores being higher than the theoretical subscores.

Table 4. Common exploitability subvectors in experiment

#	AV	AC	Au	Subscore
5045 (45.8%)	N	L	N	10.0
4005 (36.4%)	N	M	N	8.6
624 (5.7%)	L	L	N	3.9
408 (3.7%)	N	L	S	8.0
356 (3.2%)	L	M	N	3.4
231 (2.1%)	N	M	S	6.8
186 (1.7%)	N	H	N	4.9
41 (0.4%)	L	L	S	3.1
35 (0.3%)	L	H	N	1.9
28 (0.3%)	N	H	S	3.9

Next, we looked at the impact subscores. The experimental data had 21 distinct impact subvectors (of 26 possible). The two most common subvectors comprised over 58% of the data, and the ten most common comprised over 99%. Table 5 shows the experimental results for the ten most common impact subvectors. The most common has a mid-range score (6.4) and the next two have scores at the high and low ends of the range. This helps explain why impact subscores do not have a strong bias. Table 5 also shows examples of subvectors mapping to the same score.

Table 5. Common impact subvectors in experiment

#	C	I	A	Subscore
4108 (37.3%)	P	P	P	6.4
2347 (21.3%)	C	C	C	10.0
1827 (16.6%)	N	P	N	2.9
870 (7.9%)	P	N	N	2.9
747 (6.8%)	N	N	P	2.9
522 (4.7%)	N	N	C	6.9
208 (1.9%)	P	P	N	4.9
139 (1.3%)	C	N	N	6.9
132 (1.2%)	N	P	P	4.9
24 (0.2%)	N	C	C	9.2

We were surprised that the impact subvectors showed more diversity than the exploitability subvectors. From our experience with CVSS scoring,

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

we knew that few actual vulnerabilities had split impacts, which refers to a vulnerability with both Complete and Partial impacts (e.g., a Partial impact to Confidentiality and a Complete impact to Availability). We specifically looked for split impacts in the experimental data and found only 29 such vulnerabilities. Of the 26 theoretically possible impact subvectors, 12 have split impacts, so we expected that in the data the impact subvectors would have little diversity compared to exploitability subvectors. However, given the results in Tables 4 and 5, we conclude that impact subvectors have greater diversity than exploitability subvectors, and that exploitability subscores have greater diversity than impact subscores.

7. Vulnerability characteristics

To further investigate subvector and subscore diversity, this section presents experimental results for the diversity of the individual CVSS metric values.

7.1. Exploitability characteristics

Table 6 shows the percentage of vulnerabilities assigned to the exploitability metric values. Much of the motivation for making these three metrics more granular in v2 (three options each instead of two) was to improve score diversity and accuracy. However, Table 6 indicates that these changes have not affected a large number of vulnerabilities. The new Multiple option for Authentication was only used three times in the entire experiment set, and the new Adjacent Network option for Access Vector was used only 0.3% of the time. The new option for Access Complexity has been more heavily used, but has still only affected 2.4% of the vulnerabilities.

Table 6. Value frequencies for exploitability metrics

Metric	Value	v1 %	v2 %
AccessComplex-ity	Low	55.7	55.7
	Medium	N/A	41.9
	High	44.3	2.4
Authentication	NR/None	93.3	93.3
	Required	6.7	N/A
	Single	N/A	6.7
	Multiple	N/A	0.0
AccessVector	Remote	90.2	N/A
	Network	N/A	89.9
	Adj. Network	N/A	0.3
	Local	9.8	9.8

We examined the data for the vulnerabilities that used the new options. First, we examined the entries with AccessComplexity set to High. Without the High option (setting them to Medium instead), these vulnerabilities would have exploitability subscores that were on average 3.2 higher (median 3.7, range 1.2 to 3.7, standard deviation 0.84) and base scores that were on average 1.5 higher (median 1.7, range 0.5 to 1.8, standard deviation 0.38). This indicates that the use of the Medium and High options is succeeding at distinguishing the severity of vulnerabilities, albeit for only 2.4% of the total vulnerabilities.

Next, we examined the vulnerabilities with AccessVector set to Adjacent Network to see how their values would differ if the Adjacent Network option did not exist (using Network instead). Their exploitability subscores would have been an average of 2.9 higher (median 3.1, range 1.4 to 3.5, standard deviation 0.74) and their base scores an average of 1.4 higher (median 1.4, range 0.6 to 1.7, standard deviation 0.37). This is almost as large a difference as the AccessComplexity Medium and High options described above, but it affects only 0.3% of vulnerabilities.

Although three data points have little meaning, we note that the vulnerabilities with Multiple set for Authentication would have had exploitability subscores an average of 1.2 higher and base scores an average of 0.5 higher if the Multiple option had not existed (Single would have been used instead). Given the rarity of the Multiple Authentication option in the experimental data and the small change in scores, its addition to CVSS seems to have been ineffectual at improving score accuracy and diversity, while making the scoring process more complicated.

The rarity of certain exploitability metric values, as shown in Table 6, explains much of the unexpected lack of diversity in exploitability subvectors and scores. The absence of Multiple Authentication effectively reduces the number of likely vectors from 27 to 18, and the low frequencies of other metric values indicate that additional subvectors will occur very rarely.

While developing v2, the CVSS-SIG had extensive discussions about how the exploitability metrics could be defined more granularly to improve diversity, and the options deemed most feasible were added, yet they affected relatively few vulnerabilities in practice. It seems likely that it would be difficult to find feasible new ways to further distinguish vulnerabilities from each other by authentication, access vector, or access complexity without adding significant complexity to vulnerability analysis. Most CVE entries may have similar base characteristics in terms of exploitability, making further distinctions difficult at best.

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

7.2. Impact characteristics

Table 7 shows the percentage of vulnerabilities assigned to the impact metric values. The Table 7 data applies to both v1 and v2 because they use the same impact metrics. The data shows that each possible value has a relatively similar likelihood, which supports greater diversity of subvectors and scores.

Table 7. Frequencies of each value for impact metrics

Metric	Complete %	Partial %	None %
Confidentiality	22.8	47.4	29.8
Integrity	21.9	57.2	20.9
Availability	26.5	45.5	28.0

The impact metrics could be made more granular to better differentiate vulnerabilities. The Complete and None values are absolutes and cannot be made more granular, but Partial covers the entire range of impacts between the extremes of Complete and None. Having four or more possible values for each impact metric would probably improve score diversity, but it would also increase the complexity of scoring, which could make scoring less consistent among analysts. The current three values are easily distinguished from each other and it seems unlikely that multiple categories within Partial would be as easy to distinguish.

8. Severity rankings

The National Vulnerability Database (NVD) [7] 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) [8]. The motivation for having these rankings is to help organizations prioritize their mitigations of new vulnerabilities. We did rankings for the theoretical data, shown in Table 8. There has been a dramatic change in the distributions from v1 to v2, with v2 having less than a third as many Low vulnerabilities but over twice as many Medium and High vulnerabilities. The v1 scores are heavily biased to lower values and the v2 scores are more evenly distributed, with the majority of scores in the Medium range and the Low and High scores occurring with nearly equal frequency.

Table 8. NVD severity rankings for theoretical data

Rank	v1 freq	v2 freq
Low	517 (62%)	142 (20%)
Medium	259 (31%)	440 (63%)
High	55 (7%)	120 (17%)

We also applied the rankings to the experimental data; Table 9 shows the results. Most vulnerabilities scored using v1 have Low or High scores, not Medium. The v2 data in Table 8 shows a clear shift from the v1 pattern; the v2 data has few vulnerabilities with Low scores, and a rather even split between Medium and High scores. There are significant differences between the theoretical and experimental rankings, which corresponds to the previous analysis on theoretical and experimental vectors and scores.

Table 9. NVD severity rankings for experimental data

Rank	v1 freq	v2 freq
Low	4490 (41%)	393 (4%)
Medium	1912 (17%)	5388 (49%)
High	4610 (42%)	5231 (47%)

An interesting note on the change in rankings from v1 to v2 involves the Payment Card Industry (PCI) data security standard [9]. It requires that systems not have any vulnerabilities with CVSS scores of 4.0 or greater (except for vulnerabilities that could only cause a denial of service, which comprise fewer than 12% of the vulnerabilities in NVD), with the scores taken from NVD. This corresponds to vulnerabilities ranked as Medium or High by NVD. The shift from v1 to v2 scoring has caused the percentage of vulnerabilities ranked by NVD as Medium or High to go from 59% to 96%. This means that systems subject to PCI's standard are allowed to have far fewer vulnerabilities under v2 scoring than under v1. It is likely that many other organizations have similar policies and have also been significantly affected by the shift in scores.

9. Conclusions

Our analysis has demonstrated that v2 has higher average scores and greater score diversity than v1. The score average and median are in the range desired by the CVSS-SIG. The more granular values added to the exploitability metrics have collectively had a relatively small effect on score diversity, while increasing the complexity of the scoring process.

978-1-4244-4841-8/09/$25.00 ©2009 IEEE

For further improvement of CVSS, our recommendations to the CVSS-SIG are listed below. A caveat is that changes to CVSS could negatively impact its use and reputation. Since v2 has already been widely adopted, changes to it might force CVSS users to modify their processes and applications that use CVSS data. Some changes might even necessitate rescoring existing vulnerabilities, which is generally infeasible since there are so many vulnerabilities. Still, changes that simplify the specification or make it more valuable to end users may be worthwhile.

1. Remove the Multiple value from the Authentication metric. It was only used in 3 vulnerabilities out of 11,012, so it complicates scoring without adding value in differentiating vulnerabilities.
2. Evaluate the Adjacent Network value from the Access Vector metric to determine if its benefits to score accuracy and diversity outweigh the complexity that it adds to the scoring process.
3. If the CVSS-SIG wants to further increase base score diversity and accuracy, investigate the value of dividing the Partial value for the three impact metrics into multiple categories to improve score accuracy and diversity. We suspect that making the impact metrics more granular will significantly complicate scoring and increase the scoring error rate, so we caution the CVSS-SIG to carefully consider these disadvantages.
4. Before finalizing changes to CVSS, conduct vulnerability scoring with the proposed changes to ensure that the effect is as intended. Both this paper and [2] have shown that the experimental data may differ significantly from the theoretical data.

Future work will include studying a larger set of vulnerabilities to see how actual CVSS metrics change over several years. This would help in predicting whether vulnerability characteristics change so much over time that future modifications to CVSS may be needed. We also plan on comparing the NVD data to other sets of CVSS data to identify any significant discrepancies in scoring among organizations.

10. References

[1] Forum of Incident Response and Security Teams, "CVSS Adopters," http://www.first.org/cvss/eadopters.html (current 08/2009).

[2] P. Mell and K. Scarfone, "Improving the Common Vulnerability Scoring System," *IET Information Security*, London, England, Sept. 2007, pp. 119-127.

[3] P. Mell, K. Scarfone, and S. Romanosky, "A Complete Guide to the Common Vulnerability Scoring System Version 2.0," Forum of Incident Response and Security Teams, June 2007, http://www.first.org/cvss/cvss-guide.html (current 08/2009).

[4] MITRE Corporation, "Common Vulnerabilities and Exposures (CVE)," http://cve.mitre.org/ (current 08/2009).

[5] National Infrastructure Advisory Council, "Common Vulnerability Scoring System," http://www.first.org/cvss/cvss-dhs-12-02-04.pdf (current 08/2009)

[6] National Institute of Standards and Technology, "Security Content Automation Protocol (SCAP)," http://scap.nist.gov/ (current 08/2009).

[7] National Institute of Standards and Technology, "National Vulnerability Database," http://nvd.nist.gov/ (current 08/2009).

[8] National Institute of Standards and Technology, "National Vulnerability Database CVSS Scoring," http://nvd.nist.gov/cvss.cfm (current 08/2009).

[9] Payment Card Industry Security Standards Council, "Payment Card Industry (PCI) Data Security Standard: Technical and Operational Requirements for Approved Scanning Vendors (ASVs), Version 1.1", https://www.pcisecuritystandards.org/tech/supporting_documents.htm (current 08/2009).

[10] G. Reid, P. Mell, and K. Scarfone, "CVSS-SIG Version 2 History," Forum of Incident Response and Security Teams, June 2007, http://www.first.org/cvss/history.html (current 08/2009)