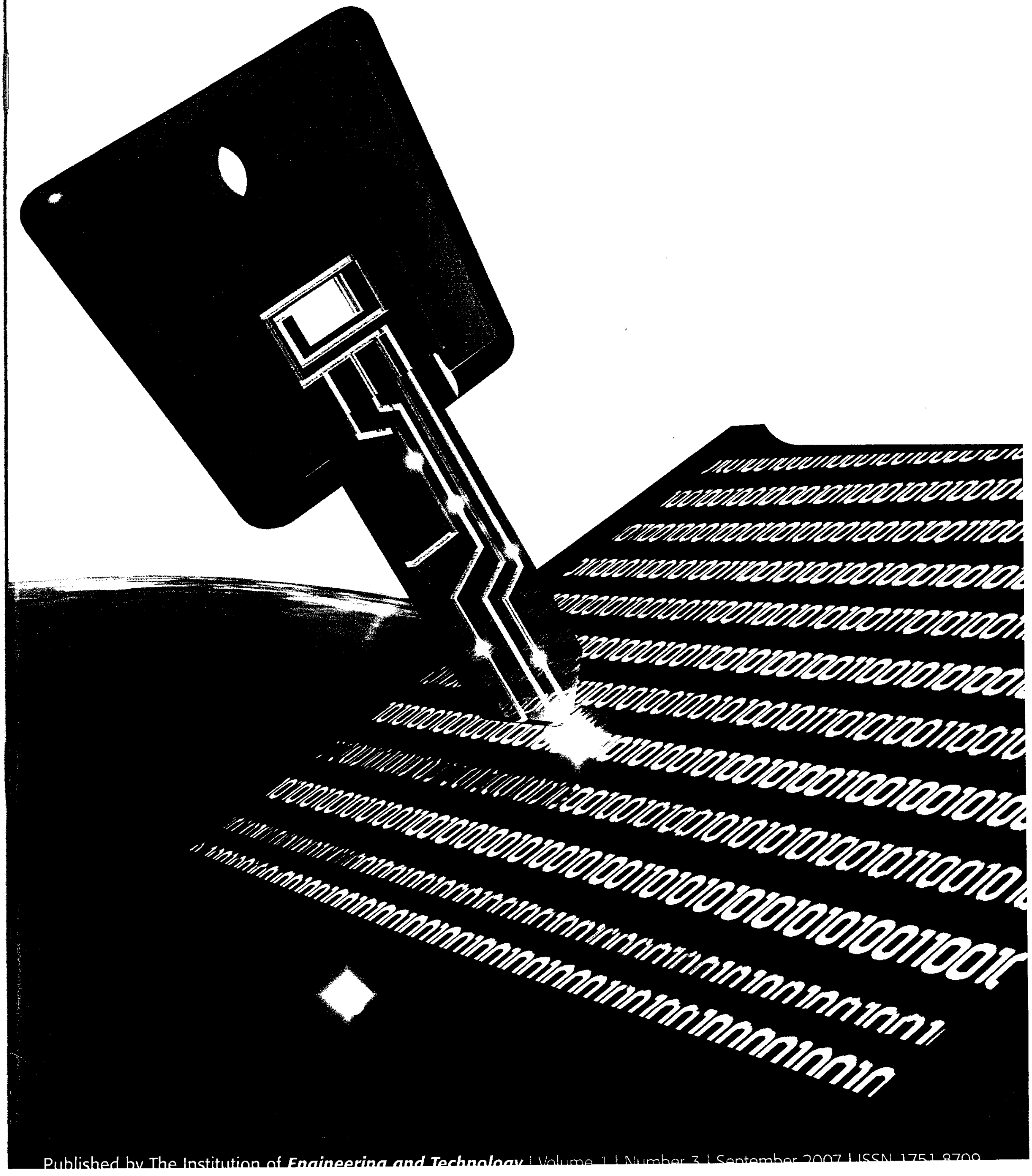
The Knowledge Network

Formerly IEE Proceedings Information Security



Published by The Institution of ***Engineering and Technology*** | Volume 1 | Number 3 | September 2007 | ISSN 1751-8709

Improving the Common Vulnerability Scoring System

P. Mell and K. Scarfone

Abstract: The Common Vulnerability Scoring System is an emerging standard for scoring the impact of vulnerabilities. The results of an analysis of the scoring system and that of an experiment scoring a large set of vulnerabilities using the standard are presented. Although the scoring system was found to be useful, it contains a variety of deficiencies that limit its ability to measure the impact of vulnerabilities. The study demonstrates how these deficiencies could be addressed in subsequent versions of the standard and how these changes are backwards-compatible with the existing scoring efforts. In conclusion a recommendation for a revised scoring system and an analysis of experiments that demonstrate how the revision would address deficiencies discovered in the existing version of the standard are presented.

1 Introduction

The Common Vulnerability Scoring System (CVSS) is an emerging standard for scoring the impact of vulnerabilities [1]. It was originally developed by the United States government National Infrastructure Advisory Council. It is currently being promoted and developed by the international forum for incident response and security teams (FIRST) and has been adopted by a variety of organisations [2].

This paper analyses the effectiveness of the CVSS version 1.0 standard on the basis of a review of the standard itself and the results of an experiment scoring 6831 vulnerabilities using the standard. Improvements to the current version of the standard are then proposed.

According to the FIRST documentation, CVSS provides a 'universal language to convey vulnerability severity and help determine urgency and priority of response' and it 'solves [the] problem of multiple, incompatible scoring systems in use today'. Although it does convey these benefits, we have discovered deficiencies with the scoring metrics and equation that limit CVSS's effectiveness. Fortunately, these deficiencies are correctable, and it is our hope that they will be addressed in the upcoming version of CVSS. Suggested solutions to these problems should allow existing scored vulnerabilities to be updated automatically to a new system without requiring additional human analysis. (This assumes, as is commonly done, that scoring entities provide detailed score component information along with their raw CVSS scores.)

2 Background

CVSS uses three groups of metrics to calculate vulnerability scores:

- Base metrics: vulnerability attributes that are constant over time and across implementations. A base score for a vulnerability is calculated by applying a formula to the values of the base metrics.
- Temporal metrics: vulnerability attributes that change over time but are the same across implementations. A temporal score for a vulnerability is calculated by applying a formula to the base score and the values of the temporal metrics.
- Environmental metrics: vulnerability attributes that are organisation and implementation-specific. An environmental score is calculated by applying a formula to the temporal score and the values of the environmental metrics.

The focus of our research is the base score. Table 1 lists the base metrics and the possible values for each metric.

There is an additional base metric for the relative importance of the three impact metrics (confidentiality, integrity and availability). In some cases, the nature of a target is such that one of the impact areas is more important than the others: an example in the CVSS standards guide [1] is of confidentiality being more important for an encrypted file system than availability. Giving greater importance to one impact area is known as an impact bias. The base metric and possible values for the impact bias are listed in Table 2.

To calculate the base score, the impact and impact bias values are first combined using the formula (ConfImpact * ConfImpactBias) + (IntegImpact * IntegImpactBias) + (Avail Impact * AvailImpactBias). The result of this formula is a value between 0 and 1. This value is multiplied by (10 * AccessVector * AccessComplexity * Authentication). The final result is a base score ranging between 0.0 and 10.0. The complete formula is: BaseScore = round_to_1_decimal (10 * Access Vector * AccessComplexity * Authentication * ((ConfImpact * Conf ImpactBias) + (IntegImpact * IntegImpactBias) + (Avail Impact * AvailImpactBias))).

CVSS was published in 2005 as a 'first-generation' open-scoring system, and the developers stated that they were seeking feedback on the scoring. Until now, no organisation has calculated a large number of CVSS scores and performed analysis of that scoring to determine the typical distribution of scores. The motivation behind this project was to perform such an analysis as a means of understanding the effectiveness of the scoring system and identifying potential improvements to CVSS.

doi:10.1049/iet-ifs:20060055

Paper first received 17th March 2006 and in revised form 23rd February 2007

The authors are with National Institute of Standards and Technology (NIST), 100 Bureau Drive, Stop 8930, Gaithersburg, MD 20899-8930, USA

E-mail: mell@nist.gov

Table 1: Base metrics

Metric name and description	Possible values
AccessVector – how the vulnerability could be exploited	remotely (1.0) only with local authentication or physical access (0.7)
AccessComplexity – how difficult it is to exploit the vulnerability	low (1.0) high (0.8), such as a victim needing to perform certain actions
Authentication – if the attacker has to authenticate after gaining access to the target to exploit the vulnerability	not required (1.0) required (0.6)
ConfImpact – how exploitation could impact the target's confidentiality	complete (1.0) partial (0.7), such as unauthorised access to a limited set of files none (0.0)
IntegImpact – how exploitation could impact the target's integrity	complete (1.0) partial (0.7), such as unauthorised changes to a limited set of files none (0.0)
AvailImpact – how exploitation could impact the target's availability	complete (1.0) partial (0.7), such as targets experiencing slowdowns or short outages none (0.0)

3 Other vulnerability scoring systems

There are several other vulnerability scoring systems, but CVSS is the only one that provides full details of how its scores are created. Partial details are available for several alternate scoring systems, each of which uses a set of metrics combined in various ways to calculate scoring. This section reviews several vulnerability scoring systems as a foundation for providing insights into possible changes to CVSS.

The United States computer emergency readiness team (US-CERT) vulnerability scoring system assigns a score between 0 and 180 for each vulnerability [3]. However, the description of the scoring system explains that the scores are approximate and that they should be used to make general comparisons of the relative severity of vulnerabilities. The scores themselves are not made available to the public; instead, they are used internally by US-CERT to determine which vulnerabilities are severe enough to merit the publication of security advisories. No scoring formula is made publicly available, although it is known that it considers several factors.

The SANS Institute has a vulnerability scoring system that is used by a group of security researchers to assign severity ratings to certain vulnerabilities [4]. No numeric scores are released – simply ratings of Critical, High, Moderate or Low. SANS provides recommended remediation timeframes for each rating category. Like the US-CERT scoring system, the SANS system generates ratings by considering several factors and not weighing them all equally. The formulas used to generate the scores and ratings are not publicly available.

Microsoft has a vulnerability rating system known as the Microsoft Security Response Center Security Bulletin Severity Rating System [5]. Each vulnerability is assigned one of four ratings: Critical, Important, Moderate or Low. Microsoft does not list the factors or formulas used to determine the ratings.

Qualys has a vulnerability rating system that classifies each possible and confirmed vulnerability as having a severity level from 1 to 5 [6]. No other information on the Qualys ratings is available.

4 Theoretical analysis of the standard

This section provides an analysis of deficiencies with the CVSS base-score equation and metrics based on an

Table 2: Impact bias metrics

	ConfImpactBias	IntegImpactBias	AvailImpactBias
Normal – all three impacts are equally important	0.333	0.333	0.333
Confidentiality – the confidentiality impact is more important than the others	0.5	0.25	0.25
Integrity – the integrity impact is more important than the others	0.25	0.5	0.25
Availability – the availability impact is more important than the others	0.25	0.25	0.5

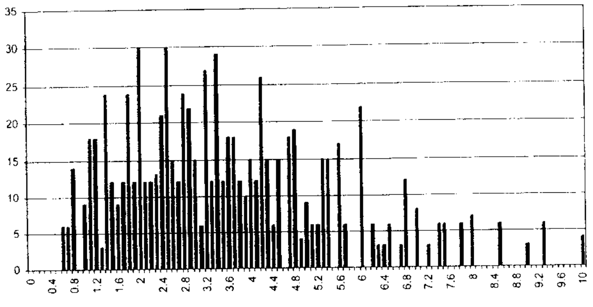


Fig. 1 *Distribution of CVSS base scores in theoretical data*

inspection of their logical and mathematical properties. Subsequent sections discuss an experiment scoring actual vulnerabilities using the standard.

4.1 Scoring distribution

The scoring distribution analysis examines the possible sets of variable inputs to the CVSS base-score equation. Because there are three metrics with two possible values each, three metrics with three possible values each and one metric with four possible values, there are 864 permutations of inputs; however, 32 of these reflect impossible conditions in which vulnerabilities are rated to have no impact on confidentiality, integrity or availability. Thus, the analysis is based on the 832 possible sets of inputs, as well as the 101 possible base-score values (0.0–10.0). Fig. 1 shows how the 832 possible input sets map to the base scores. The median of the scores is 3.2 and the average is 3.475. Fig. 2 shows the same data as Fig. 1, but it is grouped into ten-score ranges to more clearly show the overall shape and left-skewed nature of the score distribution.

4.2 Scoring distribution analysis

The theoretical distribution of CVSS version 1 scores has its peak in the 2.1–3.0 range, so it is shifted to the left considerably from the anticipated peak of 5.0–6.0. This indicates that CVSS base-scores will be biased to low values. This is problematic because practical experience with typical vulnerabilities shows that the majority of them have moderate-to-high impact and thus are deserving of scores greater than 5.0.

Another important finding from Fig. 1 is that not all of the base scores can occur. Calculating the base score for each of the 832 possible input sets yielded only 66 of the 101 possible base scores. Of the 66 possible scores, only 23 (35% of all possible scores) are greater than 5.0, and the median of the scores is 3.95. This means that much greater score diversity is possible at the lower end of the score range; for example, from 1.1 to 3.0, all 20 values are possible, but from 7.1 to 9.0, only 7 of the 20 values are possible. Thus, those vulnerabilities that are given a high score will likely have less distinguishable scores because there are relatively few distinct scores available at the high end of the scoring range. This limits the ability of CVSS scores to discriminate among different high-impact vulnerabilities.

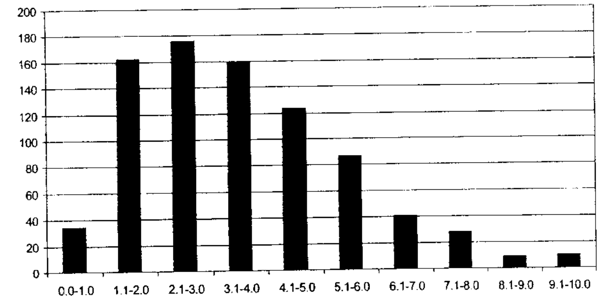


Fig. 2 *Distribution of CVSS base scores in theoretical data, grouped by tens*

4.3 Multiplicative equation problem

We analysed the CVSS base-score equation to look for the causes of the low score bias in the theoretical score distribution. The equation first performs a weighted average of three metrics (with a result from 0 to 1) and then multiplies that result and three other metrics that each have values from 0 to 1. The rest of the equation simply alters the scale of the score by multiplying the previous result by ten and rounding that to the nearest tenth.

The equation was designed to limit the scores to the 0.0–10.0 range, but its multiplicative form is responsible for the score distribution bias. Each individual metric has more influence over higher-valued scores than lower-valued scores. For example, take two vulnerabilities with scores of 10.0 and 1.6 for which Authentication is 'Not Required'. Changing Authentication to 'Required' lowers the score for the first vulnerability by 4.0 points and the score for the second vulnerability by only 0.6 points. In many cases, two vulnerabilities with five identical metrics and one different metric have significantly separated scores if they are relatively high severity vulnerabilities and very little separation in scores if they are relatively low severity vulnerabilities. This explains much of why the score distribution is weighted or biased to the left.

5 Analysis of experimental data

In the experiment, we calculated CVSS base scores for 6831 computer vulnerabilities listed in the Common Vulnerabilities and Exposures (CVE) dictionary of computer vulnerabilities [7]. We did this by interfacing with the operational scoring capability within the National Vulnerability Database (NVD) [8], which is the United States government's public database of all CVE vulnerabilities. The authors manage the NVD program. Obtaining scores from the NVD gave us a large set of experimental data, but it did restrict us to analysing only recently published vulnerabilities because the NVD did not record CVSS metrics for vulnerabilities before November 2005. However, this is not a problem since CVSS is primarily intended as a method to score the impact of recent vulnerabilities to help prioritise mitigation actions.

The vulnerabilities analysed were the set of all vulnerabilities published in the CVE dictionary between 1 December 2005 and 1 December 2006. The vulnerabilities were analysed according to the CVSS standards guide [1]. To keep the analysis focused on recent vulnerabilities, we comprehensively covered all recent CVE vulnerabilities instead of randomly sampling the CVE dictionary of 20 000 vulnerabilities (the number available in December 2006). The analysis covers 34% of all CVE vulnerabilities that were available at the time of analysis.

Table 3: Percentage of vulnerabilities that impact confidentiality, integrity or availability

	None, %	Partial, %	Complete, %
ConfImpact	31.3	64.2	4.5
IntegImpact	20.5	75.6	3.9
AvailImpact	35.1	57.8	7.1

In this section, we present the results of the experiment by looking at the CVSS base score distribution and the diversity within specific CVSS metric values that make up the base score.

5.1 Vulnerability characteristics

This section presents results on the diversity of CVSS metric inputs based on real-world vulnerability data. Each CVSS base-score metric is shown below and the percentage of vulnerabilities that fell within each category is provided.

- ImpactBias was scored as normal 99.3% of the time. The other scores were 0.1% for confidentiality, 0.1% for integrity and 0.5% for availability.
- AccessComplexity was scored as low 86.5% of the time and high 13.5% of the time.
- The authentication metric was scored as not required 96.0% of the time and required 4.0% of the time.
- The Access Vector metric was scored as remote 88.5% of the time and local 11.5% of the time.

Table 3 shows the percentage of vulnerabilities assigned to the ConfImpact, IntegImpact and AvailImpact metrics. 'Complete' means that the entire host is affected, while 'Partial' means that only a portion of the host is affected, such as user-level access.

The real-world vulnerability data indicates that for all the metrics, with the exception of ImpactBias, there was substantial variety in the values assigned to each metric. Section 7 further discusses the ImpactBias metric.

5.2 Scoring distribution

Before actually analysing the distribution of scores in the experimental data, our expectation was that certain scores would routinely occur more frequently than others. Certain types of vulnerabilities are more prevalent than others and have very similar characteristics, so each instance of one of these vulnerability types would have similar or identical metric values, and hence similar or identical scores.

The 6831 vulnerabilities in the data set produced only 35 of the 66 possible scores. Of those, just two scores (2.3 and 7.0) covered 67.5% of the vulnerabilities, and 10 scores covered 95.2% of the vulnerabilities. Fig. 3 shows the distribution of the scores in the experimental data. The experimental data has little resemblance to the theoretical distribution shown in Figs. 1 and 2 because of the strong predominance of particular vulnerability types within the experimental data set.

This empirical distribution does not necessarily represent a problem, if it is caused solely because certain types of vulnerabilities occur with much greater regularity than others in the real world. To further examine the patterns, we performed additional analysis of the experimental data. We first looked at the metric input sets in the data to see which sets occurred most frequently and which scores they mapped to. Table 4 shows the five most frequently occurring metric input sets.

As Table 4 shows, the most common set comprised 36.0% of all scored vulnerabilities, which had 7.0 scores. The second, fourth and fifth most common sets were identical except for the element of security that is impacted. The vulnerabilities in those three sets, which all have 2.3 scores, total 28.4% of all scored vulnerabilities. These numbers show that the prevalence of a few vulnerability types is the primary cause of the bimodal nature of the score data. This indicates that a bell curve or other relatively symmetric distribution of scores over the 0.0–10.0 range is probably not feasible with real vulnerability data.

To look for other causes of the score distribution, we examined the experimental data for vulnerabilities that had the most common scores, 2.3 and 7.0. We found

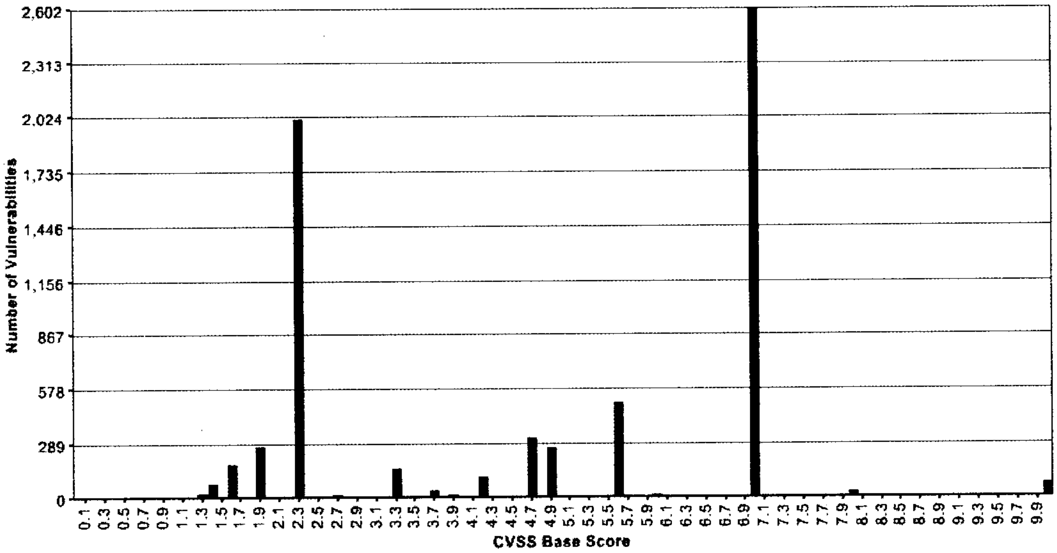


Fig. 3 *Distribution of CVSS version 1 base scores in experimental data*

Table 4: **Most frequently occurring metric input sets in experimental data**

Vulnerability count and % of all vulnerabilities	Score	AccessVector	AccessComplexity	Authentication	ConfImpact	IntegImpact	AvailImpact	ImpactBias
2461 (36.0%)	7.0	remote	low	not required	partial	partial	partial	none
1072 (15.7%)	2.3	remote	low	not required	none	partial	none	none
481 (7.0%)	5.6	remote	high	not required	partial	partial	partial	none
479 (7.0%)	2.3	remote	low	not required	partial	none	none	none
389 (5.7%)	2.3	remote	low	not required	none	none	partial	none

multiple instances where different types of vulnerabilities had the same score but appeared to cause different impacts in the real world. This is a secondary cause of the bimodal nature of the scores. Section 6.3 discusses impact comparisons in more detail.

5.3 *Sources of error*

The analysis of the vulnerability set undoubtedly contains some errors, although qualitatively the analysis appears reasonably accurate for the purposes of this study. Quantifying the true error rate is impossible since there is no authoritative source of correct CVSS scores.

One source of error is analysts deciding whether a vulnerability gives root or user level access to an operating system. This is not always clear from the advisory and it may depend on how a system administrator installs the software. This ambiguity may be partially responsible for the increased spike at score 7.0, as the analysts appear to have avoided claiming that a vulnerability gives root access unless it was clear that this was the case, which would give scores of 10.0.

6 Analysis of scoring accuracy

During the work described in Sections 4 and 5, questions were raised about the scores assigned to certain sets of metric inputs. To resolve these, we performed additional analysis of the equation and metrics to identify deficiencies and possible solutions.

6.1 *Impact Metrics*

The base-score equation gives equal weighting to the confidentiality, integrity and availability impact metrics (assuming the bias setting is 'normal'). However, in the real world, they do not necessarily affect a vulnerability's impact equally.

Integrity should be given more weight than availability because violations of integrity nearly always allow violations of availability. If an attacker can make arbitrary changes to a system, then changes could be made to negatively impact availability. For example, if one can modify or delete data (an integrity violation), then the data can be made unavailable (an availability violation). Additionally, exploitation of integrity has greater impact because it is often difficult to notice the violation, determine what was changed and restore the target to a clean state. This is not true with exploitation of availability.

Confidentiality should be given more weight than availability because many violations of confidentiality are nonrecoverable (for example, the theft of sensitive personnel records), while violations of availability are recoverable. Also, any exploitation of availability is typically noticed very quickly, while violations of confidentiality are hard to detect.

We propose giving confidentiality and integrity more weight than availability in the base-score equation. This could be done in a few ways, such as altering the equation and metric values so that the confidentiality and integrity metrics have higher values, or requiring that the availability metric be set at least as high as the higher of the confidentiality and integrity metrics. For cases where the assumed bias is not accurate, end users could modify the bias in the environmental scoring metrics.

6.2 *ImpactBias metric*

As described in Section 5.1, the ImpactBias metric affected the score for only 0.7% of the vulnerabilities. Given the metric's limited influence on the scores, it is probably not worth the time for CVSS analysts to determine how this metric should be set for each vulnerability. Also, in many cases, the proper setting for this metric is environment-dependent. For example, the integrity of a particular application might be more important to one organisation, and its availability might be more important to another. Moving the ImpactBias metrics to the environmental scoring would allow end users to override the existing bias in the base scoring, and would simplify the work of CVSS analysts.

6.3 *Incorrect scoring*

As part of the analysis, we wanted to ensure that the CVSS scores reasonably reflect the relative severity of vulnerabilities. We started doing this by looking at instances where multiple sets of inputs generate the same score. We first examined the inputs for the most commonly occurring scores, 2.3 and 7.0, and considered their relative severity. Remotely exploitable vulnerabilities that provide user level access to the operating system (OS) and locally exploitable vulnerabilities that provide complete control over the OS both produce scores of 7.0 (assume for this and future examples that all other metrics take their most probable values as specified in Section 5.1). The former should be scored higher since such a vulnerability generally has a greater impact. Another example is that remotely exploitable vulnerabilities that allow a violation of confidentiality within an application and locally exploitable vulnerabilities that allow a violation of confidentiality within the entire operating system both score 2.3. Again, the former should be scored higher because it will have a greater impact in most environments.

We examined other sets of inputs, and we found a few sets of metric inputs that produce scores that are too low relative to other types of vulnerabilities. For example, vulnerabilities providing complete control of an OS in the areas of confidentiality and integrity score lower than vulnerabilities providing user-level access to the OS. Also, remotely exploitable vulnerabilities providing complete control of the integrity of an OS score lower than remotely exploitable vulnerabilities providing user-level access to the OS. Several additional situations were also identified where different sets of inputs generate the same score but the severities of the vulnerabilities are substantially different in the real world.

In general, two trends were noted. First, the scoring difference between remotely and locally exploitable vulnerabilities is not as large as it should be. Second, vulnerabilities with a complete impact in one or two areas are being scored lower than comparable vulnerabilities with a partial impact in more areas, when the reverse should actually be the case. These problems can be corrected by altering the weights given to the metrics and/or adjusting the equations. Ideally, the metric input sets would be ordered (if not individually, then in groups) and the equation modified to approximate those orderings to within a small known degree of error.

6.4 Score diversity

All of the 6831 vulnerabilities in the experiment map to only 35 scores out of 68 possible scores, with 67.5% of the vulnerabilities having one of two scores. In practice, the CVSS base scores do not offer a greater range of impact values than other scoring systems that provide only 3 or 4 impact ratings for vulnerabilities [3–6]. This lack of score diversity is primarily due to most vulnerabilities having one of a small group of metric input sets. A secondary factor is different metric input sets mapping to the same score, even though they may have different impacts.

One way to improve score diversity is to add new metrics. However, it is not apparent that typical vulnerability advisories contain additional discriminators that could be added to the base-score equation. Another possibility is to make some of the existing metrics more granular (e.g. three or four options instead of two), and this has recently been done by the CVSS standards committee. This additional granularity will be available in the next version of the standard. Despite this, given the prevalence of certain metric input sets, it is likely that the current bimodal nature of the scores can be reduced only slightly.

7 Proposed revisions to CVSS

The problems identified within CVSS indicate that deficiencies exist with the current standard, and there exists a need to create a new version.

7.1 Revision approach

Our approach to revising the CVSS standard was to work with the CVSS standards committee to address necessary changes to the CVSS metrics themselves and to work with a group of mathematicians from NIST to create a completely new equation based on the revised metrics. Since CVSS version 1.0 is working operationally, despite its deficiencies, we followed CVSS version 1.0 in cases where we had not discovered a deficiency.

The CVSS standards committee rejected the proposal discussed in Section 6.1 of weighting confidentiality, integrity and availability differently in favour of allowing the end user the option of creating user-defined weightings. Thus, we could not address this particular shortcoming in the proposed base-score equation. The CVSS standards committee did agree to the proposal in Section 6.2 of moving the impact-bias metric into the environmental section. Thus, we were able to eliminate the impact-bias metric from the proposed equation. The CVSS standards committee also addressed the score-diversity problem, discussed in Section 6.4, by adding a third metric value to the Access Complexity, Authentication and Access Vector metrics, which each used to have only two possible settings.

We solved the scoring inaccuracies problem highlighted in Section 6.3 by dividing the CVSS equation into two subcomponents: the potential impact of exploiting a vulnerability and the difficulty of exploiting a vulnerability. We then had groups of incident response experts from the CVSS Management Team [9] to discuss how different vulnerability types should be scored with respect to 'impact' and 'difficulty'. The output of this effort was given in two tables that listed all possible 'impact' vectors and 'difficulty' vectors and the scores assigned to them by the panel of experts. Dividing the equation conceptually into 'impact' and 'difficulty' was important because each of the subequations had only 27 different possible inputs; this was a small enough set for experts to manually order and score accurately. Doing so for the entire CVSS vector set was impossible because of its size. We used the experts' ordering and scores as our 'oracle' and the mathematicians approximated the oracle with an equation that combined both 'impact' and 'difficulty'.

When a single equation was created that combined 'impact' and 'difficulty', there were a few unforeseen consequences, and the equation had to be modified iteratively to satisfy the panel of experts who score vulnerabilities on a daily basis.

7.2 Solution

In this section, we provide the CVSS base-score equation (Fig. 4), which we are proposing as the replacement to the existing CVSS version 1.0 standard. This equation was developed as described in Section 7.1. Table 5 lists the revised metric values.

7.3 Comparison of theoretical distributions

Figs. 1 and 2 in Section 4 show the CVSS version 1.0 theoretical distribution. Fig. 5 and 6 show the theoretical distribution for the proposed revised equation.

The CVSS version 1 distribution has an average of 3.6, whereas the revised equation has an average of 5.5. These averages change to 3.5 and 5.29 if the vectors that map to a score of 0 is included (see Section 4 for an explanation of why we omitted the zero value scores). This change in average theoretical score is important because security experts have complained that CVSS version 1 scores tend to be too low, and the revision puts the average theoretical score between 5 and 7, which is a reasonable average value for a vulnerability.

```
Base Score Equation =

((0.6*(10.41*(1-(1-ConfImpact)*(1-IntegImpact)*(1-AvailImpact))) +

0.4*(20*AccessComplexity*Authentication*AccessVector))-1.5)*f(x)


f(x)= 0 if ConfImpact=IntegImpact=AvailImpact=0

     1.176 otherwise
```

Fig. 4 *CVSS base-score equation*

Table 5: Base metrics for the revised equation

Metric name and description	Possible values
AccessVector – how the vulnerability could be exploited	remotely (1.0) local network access (0.646) only with local authentication or physical access (0.395)
AccessComplexity – how difficult it is to exploit the vulnerability	low (0.71) medium (0.61) high (0.35)
Authentication – if the attacker has to authenticate after gaining access to the target to exploit the vulnerability	not required (0.704) single instance required (0.56) multiple instances required (0.45)
ConfImpact – how exploitation could impact the target's confidentiality	complete (0.660) partial (0.275) none (0.0)
IntegImpact – how exploitation could impact the target's integrity	complete (0.660) partial (0.275) none (0.0)
AvailImpact – how exploitation could impact the target's availability	complete (0.660) partial (0.275) none (0.0)

7.4 Comparison of experimental distributions

We were able to use the NVD operations to compare CVSS version 1 and the revised version using all CVE vulnerabilities published between 15 December 2006 and 9 February 2007. As before, operational realities limited our ability to collect random samples of both archival and recent data, so we chose to completely cover all 1195 recent vulnerabilities in the sample. There was insufficient data to fully score 35 of the CVEs and four were rejected duplicates, so the final sample size was 1156 vulnerabilities. These vulnerabilities were scored with CVSS version 1 and the proposed revision, with one exception. NVD stopped recording the CVSS impact bias during this time period because the CVSS standards committee had voted to remove it from the next version of the standard. To resolve this deficiency, we assumed a 'normal' impact bias when doing all of the CVSS version 1 scoring with this experiment.

Fig. 7 shows the experimental distribution using CVSS version 1.0 and Fig. 8 shows the experimental distribution using the revised CVSS base score equation. The average score for the revised CVSS was 6.7, while the average for CVSS version 1.0 was 5.4. This increase reflects the desire of the security community to shift the score distribution higher since CVSS version 1.0 is largely regarded as providing scores that are too low.

The revised CVSS produced 50 distinct scores in the experimental data while CVSS version 1.0 produced only 28. (Note that Figs. 7 and 8 do not show those scores for which there were fewer than four vulnerabilities because of their low frequency compared with the peaks.) Thus the revised CVSS provides more score diversity than CVSS version 1.0.

While greater score diversity was achieved, the majority of the experimental data continue to map to only a few scores with both CVSS version 1.0 and the revised CVSS. This is primarily because the majority of vulnerabilities are the same few types. One way to overcome this hurdle is to weight the impact of confidentiality, integrity and availability differently (as we suggested in Section 6.1). Note that weighting confidentiality, integrity and availability differently would even out much of the graph, but the high frequency at 7.5 in the revised CVSS and 7.0 in CVSS version 1.0 would still not be affected.

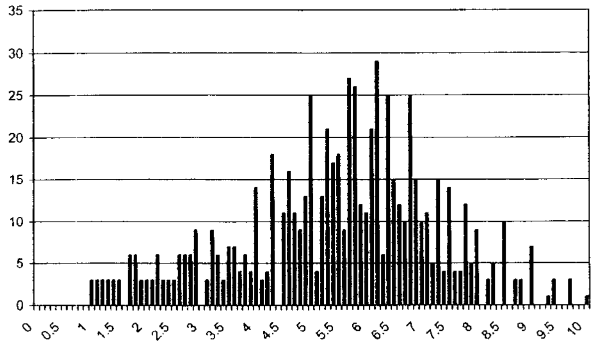


Fig. 5 *Revised CVSS equation theoretical distribution*

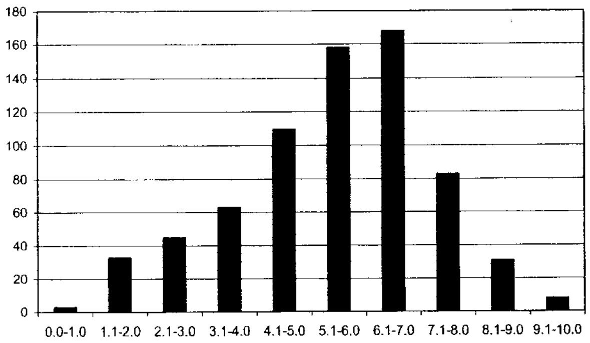


Fig. 6 *Distribution of revised CVSS base scores in theoretical data, grouped by tens*

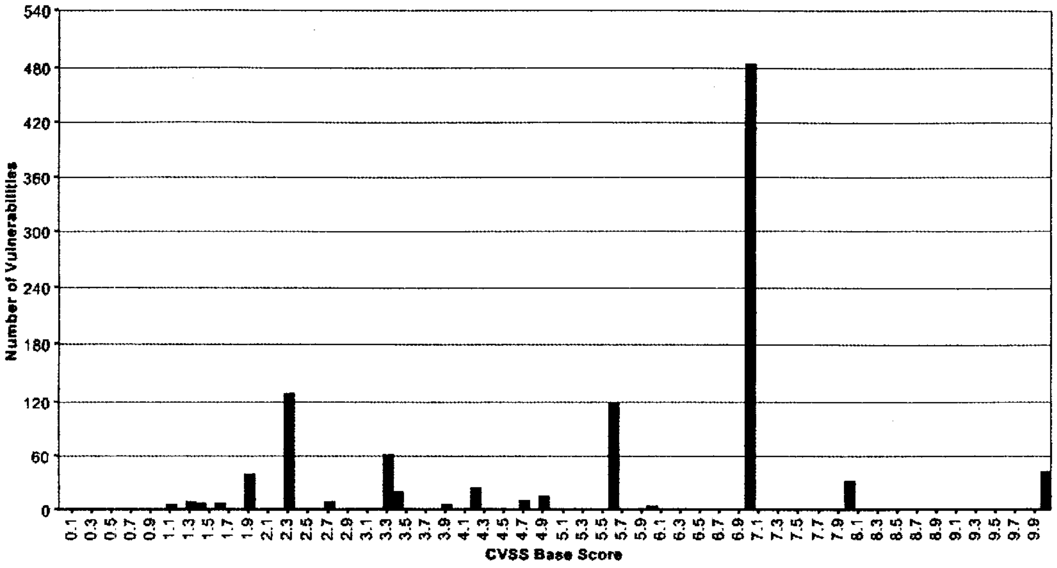


Fig. 7 *Distribution of CVSS version 1.0 base scores in second experiment*

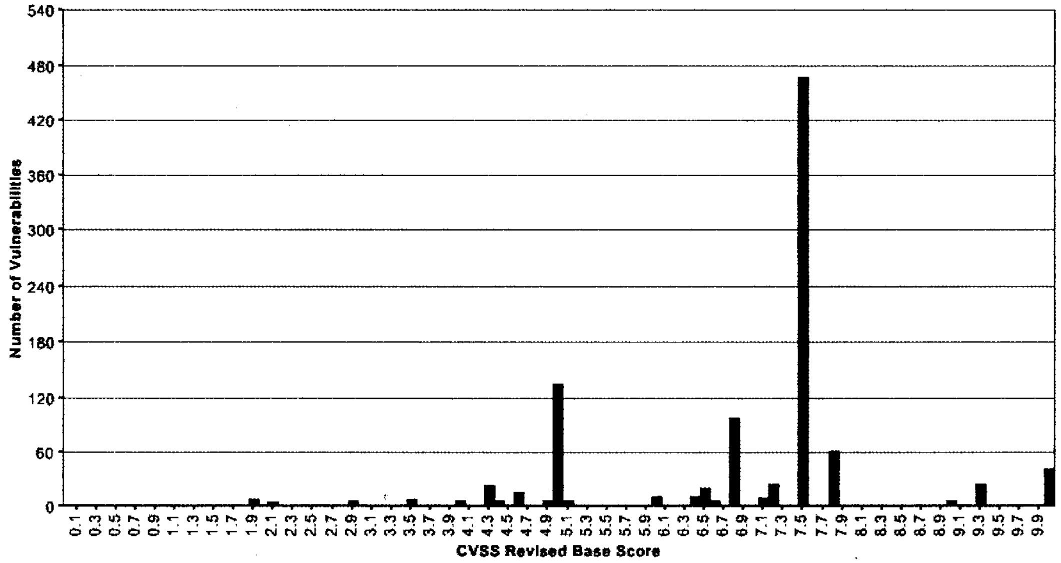


Fig. 8 *Distribution of CVSS revised version base scores in second experiment*

8 Conclusion

We have identified several deficiencies in CVSS version 1 that limit its ability to accurately estimate the potential impact of vulnerabilities. These deficiencies were identified through an analysis of experimental scoring data for 6831 vulnerabilities, as well as an inspection of the mathematical properties of the CVSS base-score metrics and equation.

To address the deficiencies in CVSS, we have made several recommendations for how the scoring system could be changed in future versions of the standard and proposed a revised scoring equation. These changes would improve the accuracy of the scores, which would help organisations and individuals better prioritise their responses to new vulnerabilities. The changes would also be backwards-compatible with scoring performed using the original version of the CVSS standard. Existing scoring metric values could simply be entered into a new equation to generate a new CVSS base score. We performed a follow-on experiment to show that the revised CVSS equation does represent an improvement over the existing equation.

In the future, we plan on using the same approach that we had taken in analysing the CVSS base-score metrics and equation to identify potential deficiencies in the CVSS temporal and environmental scores.

9 Acknowledgments

While the authors solely conducted the research critiquing CVSS version 1 and primarily conducted the research developing and evaluating the proposed revised CVSS equation, others also deserve credit for their contributions. National Institute of Standards and Technology statisticians Stefan Leigh, Andrew Rukhin and James Yen assisted in the development of the revised equation itself. Also, the CVSS special interest group (SIG) under the auspices of the

international FIRST led the revision to the CVSS metrics to address the discovered deficiencies. They also provided feedback on the deficiencies that the authors discovered within CVSS version 1. Lastly, the CVSS SIG has exhaustively tested the authors' revised equation proposal and requested that the authors update the revision repeatedly to address problems with the base scores generated for particular vulnerability types. The names of the individuals on the CVSS SIG can be found on the CVSS web site [9].

10 References

1 Schiffman, M.: 'A complete guide to the common vulnerability scoring system', 2005. Available at: http://www.first.org/cvss/cvss-guide.html, accessed 9 March 2006

2 Forum of Incident Response and Security Teams (FIRST). FIRST web site, 2006. Available at: http://www.first.org/, accessed 9 March 2006

3 United States Computer Emergency Readiness Team (US-CERT). US-CERT vulnerability note field descriptions, 2006. Available at: http://www.kb.cert.org/vuls/html/fieldhelp, accessed 9 March 2006

4 SANS Institute. SANS critical vulnerability analysis archive. Undated. Available at: http://www.sans.org/newsletters/cva/, accessed 9 March 2006

5 Microsoft Corporation. Microsoft security response center security bulletin severity rating system, 2002. Available at: http://www.microsoft.com/technet/security/bulletin/rating.mspx, accessed 9 March 2006

6 Qualys. Vulnerability severity levels defined. Undated. Available at: http://www.qualys.com/research/rnd/knowledge/severity/, accessed 9 March 2006

7 The MITRE Corporation. Common vulnerabilities and exposures, 2006. Available at: http://cve.mitre.org/, accessed 9 March 2006

8 National Vulnerability Database. Available at: http://nvd.nist.gov/

9 CVSS Management Team listing. Available at: http://www.first.org/cvss/team/