Withdrawn Draft

Warning Notice

The attached draft document has been withdrawn, and is provided solely for historical purposes. It has been superseded by the document identified below.

Withdrawal Date November 15, 2022

Original Release Date June 8, 2022

Superseding Document

Status	Final
Series/Number	NIST Internal Report 8409
Title	Measuring the Common Vulnerability Scoring System Base Score Equation
Publication Date	November 2022
DOI	https://doi.org/10.6028/NIST.IR.8409
CSRC URL	https://csrc.nist.gov/publications/detail/nistir/8409/final
Additional Information	

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY U.S. DEPARTMENT OF COMMERCE

NIST IR 8409 ipd

Measuring the Common Vulnerability Scoring System Base Score Equation

Initial Public Draft

Peter Mell
Jonathan Spring

Domain Expert Co-authors:
Srividya Ananthakrishna
Francesco Casotto
Dave Dugal
Troy Fridley
Christopher Ganas
Arkadeep Kundu
Phillip Nordwall
Vijayamurugan Pushpanathan
Daniel Sommerfeld
Matt Tesauro
Chris Turner

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8409.ipd

National Institute of
Standards and Technology
U.S. Department of Commerce

NIST IR 8409 ipd

Measuring the Common Vulnerability Scoring System Base Score Equation

Initial Public Draft

Peter Mell, National Institute of Standards and Technology
Jonathan Spring, CERT/CC at Carnegie Mellon University

Domain Expert Co-authors:
Srividya Ananthakrishna, Huntington Ingalls Industries
Francesco Casotto, Cisco
Dave Dugal, Juniper
Troy Fridley, AcuityBrands
Christopher Ganas, Palo Alto Networks
Arkadeep Kundu, Cisco
Phillip Nordwall, Dell
Vijayamurugan Pushpanathan, Schneider Electric
Daniel Sommerfeld, Microsoft
Matt Tesauro, Open Web Application Security Project
Chris Turner, National Institute of Standards and Technology

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8409.ipd

June 2022

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

National Institute of Standards and Technology Interagency or Internal Report NIST IR 8409 ipd
Initial Public Draft
42 pages (June 2022)

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8409.ipd

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

Public comment period: June 8, 2022- July 29, 2022

Submit comments on this publication to: ir8409-comments@nist.gov

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

All comments are subject to release under the Freedom of Information Act (FOIA).

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems.

Abstract

This work evaluates the validity of the Common Vulnerability Scoring System (CVSS) Version 3 "base score" equation in capturing the expert opinion of its maintainers. CVSS is a widely used industry standard for rating the severity of information technology vulnerabilities; it is based on human expert opinion. This study is important because the equation design has been questioned since it has features that are both non-intuitive and unjustified by the CVSS specification. If one can show that the equation reflects CVSS expert opinion, then that study justifies the equation and the security community can treat the equation as an opaque box that functions as described.

This work shows that the CVSS base score equation closely though not perfectly represents the CVSS maintainers' expert opinion. The CVSS specification itself provides a measurement of error called "acceptable deviation" (with a value of 0.5 points). In this work, the distance between the CVSS base scores and the closest consistent scoring systems (ones that completely conform to the recorded expert opinion) is measured. The authors calculate that the mean scoring distance is 0.13 points and the maximum scoring distance is 0.40 points. The acceptable deviation was also measured to be 0.20 points (lower than claimed by the specification). These findings validate that the CVSS base score equation represents the CVSS maintainers' domain knowledge to the extent described by these measurements.

Keywords

computer; Common Vulnerability Scoring System; error; expert opinion; measurement; measuring; metrics; network; scoring; security.

Audience

The audience for this document includes security professionals and scientists who seek to understanding the accuracy and precision of the CVSS base score equation in representing the CVSS maintainers' human expert opinion.

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

(a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

(b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

    i. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or

    ii. without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to: ir8409-comments@nist.gov

Table of Contents

List of Appendices

List of Figures

List of Tables

Executive Summary

The Common Vulnerability Scoring System (CVSS) Version 3 maintained by the CVSS Special Interest Group (SIG) is a widely used industry standard for characterizing the properties of information technology vulnerabilities and measuring their severity. It is based on human expert opinion. Vulnerability properties are characterized through a multi-dimensional vector. The severity is defined primarily through a multi-part "base score" equation, with 8 input metrics, that is not readily amenable to human comprehension.

To develop the equation, CVSS SIG members first described a set of real vulnerabilities using CVSS vectors and assigned them one of five severity levels. This created a partial lookup table mapping vectors to severity levels. They then defined a target score range for each severity level and created an equation to attempt to map each vector to a score within the specified score range. Finally, they reviewed the equation's scoring of vectors not included in the partial lookup table to evaluate the effectiveness of the equation on the full set of possible vectors. Since the equation could not perfectly map vectors to score ranges, the CVSS Version 3.1 specification provides a measurement of error (an 'acceptable deviation' of 0.5 points). However, sufficient information is not provided to reproduce the experiment.

This work measures the degree to which the CVSS base score equation reflects the CVSS SIG expert domain knowledge while providing a reproducible justification for the measurements. It starts not from a set of real vulnerabilities, as the CVSS SIG did, but from a set of 66 vulnerability types (i.e., CVSS vectors) that represent 90 % of the vulnerabilities published by the U.S. National Vulnerability Database. CVSS SIG experts then evaluate these vulnerability types and encode their knowledge as constraint graphs; sets of graphs are then unified using a voting algorithm. These unified graphs represent sets of consistent scoring systems (mappings of vectors to scores).

The consistent scoring system closest to the CVSS Version 3.1 scores was found, and the distance between the scores and the closest consistent scoring system scores was measured. These measurements represent the degree to which the CVSS v3.1 base score equation represents the CVSS SIG expert domain knowledge.

Using this approach, the mean and maximum distance of the CVSS v3.1 scores compared to the closest consistent scoring system scores was measured and the acceptable deviation was recalculated. Unlike acceptable deviation, the new distance metrics measure the score values themselves separate from the severity levels. Using all 12 CVSS SIG inputs, the mean scoring distance is 0.13 points, the maximum scoring distance is 0.40 points, and the acceptable deviation is 0.20 points. Sets of 11 out of 12 of the inputs were used to calculate precision measurements (i.e., standard deviation).

These findings validate that the CVSS base score equation functions as described (to the extent described by these measurements); it represents the encoded CVSS SIG domain knowledge. The measurements support the equation as defined. The security community may use it as an opaque box without understanding the internal functionality.

1 Introduction

This work evaluates the validity of the Common Vulnerability Scoring System (CVSS) Version 3 (v3) "base score" equation in capturing the expert opinion of its maintainers. CVSS is managed under the auspices of the global Forum of Incident Response and Security Teams (FIRST) and is maintained by the CVSS Special Interest Group (SIG). It is a widely used industry standard for characterizing the properties of information technology vulnerabilities and measuring their severity, and it is based on human expert opinion. Vulnerability properties are characterized through a multi-dimensional vector. The severity is primarily defined through a multi-part base score equation with 8 input metrics, that is not readily amenable to human comprehension. It combines sub-equations that measure vulnerability impact with others measuring the degree of exploitability. To understand why the equation is complex and not human readable, one must understand how it was created and its specific objective. Therefore, understanding the specific objective is necessary to measure the degree to which it meets its objective.

To develop the CVSS v3 base score equation, CVSS SIG members first described a set of real vulnerabilities using CVSS vectors and assigned them one of five severity levels: Low, MedLow, MedHigh, High, and Critical. This created a partial lookup table mapping vectors to severity levels; it is partial because only a small number of the 2592 possible vectors were mapped. They then defined a target score range for each severity level and created an equation to attempt to map each vector to a score within the specified score range. Finally, they selectively reviewed the equation's scoring of vectors not included in the partial lookup table to review the effectiveness of the equation on the full set of possible vectors. The assumption behind this approach is that an equation developed to accurately map a subset of the vectors would reasonably map the rest of the vectors. The assumption was deemed to hold, as verified by CVSS SIG testing. However, the equation could not always map vectors to the specified score ranges. For this reason, the CVSS v3 specification provided a measurement of error called "acceptable deviation" (measured to be 0.5 points), which measures the maximum deviation of a vector's score from its target score range. However, the underlying data is not provided that would enable one to reproduce the experiment.

This work measures the degree to which the v3 base score equation reflects the CVSS SIG expert domain knowledge while providing a reproducible justification for the measurements. It starts not from a set of real vulnerabilities, as the CVSS SIG did, but from a set of 66 vulnerability types (i.e., CVSS vectors) that represent 90 % of the vulnerabilities published by the U.S. National Vulnerability Database. CVSS SIG experts then evaluate these vulnerability types and encode their knowledge as constraint graphs. CVSS SIG members who self-identified as vulnerability experts were used because the equation is designed to reflect their expert opinion. Twelve separate evaluations of the 66 vectors were received in the form of constraint graphs; the 12 graphs were then unified using a voting algorithm to create a single set of constraints representing CVSS SIG domain knowledge. This unified constraint graph represents a set of consistent scoring systems (mappings of vectors to scores). For each of these metrics, the consistent scoring system closest to the CVSS v3 scores was found, and the distance between the scores and the closest consistent scoring system was measured. These measurements represent the degree to which the CVSS SIG expert domain knowledge is represented by the base score equation.

Using this approach, the mean and maximum distance of the CVSS v3 scores compared to the closest consistent scoring system scores were measured and the acceptable deviation was recalculated. Unlike acceptable deviation, the new distance metrics measure the score values themselves separate from the severity levels. Using all 12 CVSS SIG inputs, the mean scoring distance is 0.13 points, and the maximum scoring distance is 0.40 points. The acceptable deviation is 0.20 points (i.e., maximum distance from a severity boundary). Sets of 11 out of 12 of the inputs were also used to calculate the precision of these measurements (i.e., standard deviation). The v3 base score equation was found to have a mean scoring distance of 0.13 points with a standard deviation of 0.02 points and maximum scoring distance of 0.52 points with a standard deviation of 0.15. If one assumes a "normal" Gaussian distribution, there is then a 95 % chance that the mean scoring distance is between 0.11 and 0.15 points and that the maximum scoring distance is within 0.32 and 0.82 points.

This study is important because the CVSS v3 base score equation design has been questioned since it has features that are both non-intuitive and not justified by the CVSS specification. By showing the degree to which the equation reflects the CVSS SIG maintainers' expert opinion, the degree to which the equation meets its objective is measured. These findings validate that the CVSS base score equation functions as described (to the extent described by the distance measurements). The measurements support the equation as defined. The security community may use it as an opaque box without understanding the internal functionality.

Note that the base score reflects the severity of a vulnerability detached from any particular deployment context. CVSS also provides "temporal" and "environmental" equations that address the changing severity of a vulnerability over time and a vulnerability's severity in the context of a deployed system. While important to CVSS, evaluations of the temporal and environmental scoring equations were not within the scope of this research.

The rest of this publication is organized as follows. Section 2 provides the background on CVSS, including details on its base score metrics and equation. Section 3 then describes the rationale for the equation, how it was developed, and the measurement of error provided within the CVSS v3 specification. Section 4 pivots to the authors' research by describing the tools, metrics, and algorithms used for this study. This includes the tool for collecting and encoding CVSS domain knowledge, an explanation of knowledge constraint graphs, and the voting algorithm for unifying multiple graphs. Section 5 focuses on data collection and processing by describing the set of analyzed CVSS vectors, the participants included in the study, the produced knowledge constraint graphs, and the unified knowledge constraint graph. Section 6 describes the measurement approach, defines "consistent scoring systems", and describes heuristics for identifying the closest consistent scoring system. These two concepts are then used to elaborate the measurement methodology to measure the distance between CVSS scores and the closest consistent scoring system. Section 7 presents the results with measurements of mean distance, maximum distance, and acceptable deviation. Section 8 interprets these results and relates them to the findings of other research. Section 9 is the conclusion.

2 Common Vulnerability Scoring System

In 2003 the United States National Infrastructure Advisory Council (NIAC) [2] commissioned a working group of industry and academia security experts to design a vulnerability scoring system. The goal was to create a single open, comprehensive, interoperable, flexible, and simple approach to promoting a common understanding of vulnerability severity. The resulting Common Vulnerability Scoring System (CVSS) was presented in a NIAC report in 2004 [3]. In 2005, CVSS was transitioned to the Global Forum of Incident Response and Security Teams (FIRST) [4] for its ongoing development and maintenance. FIRST released the CVSS Version 1.0 specification [5] in 2005, Version 2.0 [6] in 2007, and Version 3.0 [7] in 2015. The current Version 3.1 [1] was released in 2019 and is the one evaluated in this publication.

CVSS contains three metric groups: base, temporal, and environmental. The base metrics define the intrinsic severity of a vulnerability in general for the world-wide computing infrastructure. The temporal metrics evaluate the severity of a vulnerability over time. And the environmental metrics measure the severity of a vulnerability relative to a particular computing environment. The score produced by a metric group may be fed as input into another, as shown in Figure 1.

The output of the scoring is a single score (from the base metrics and, optionally, the temporal and environmental) and a vector string that lists the specific input metric values that produced the score. The vector strings use acronyms to represent the input metrics and their assigned metric values; the base score vector string acronyms are listed in Appendix A.

The scope of this research is the base metric scoring, more specifically the equation used to calculate the v3 base scores. This covers both v3.0 and v3.1 as the base score equation is identical for both. The temporal and environmental scoring are not discussed.

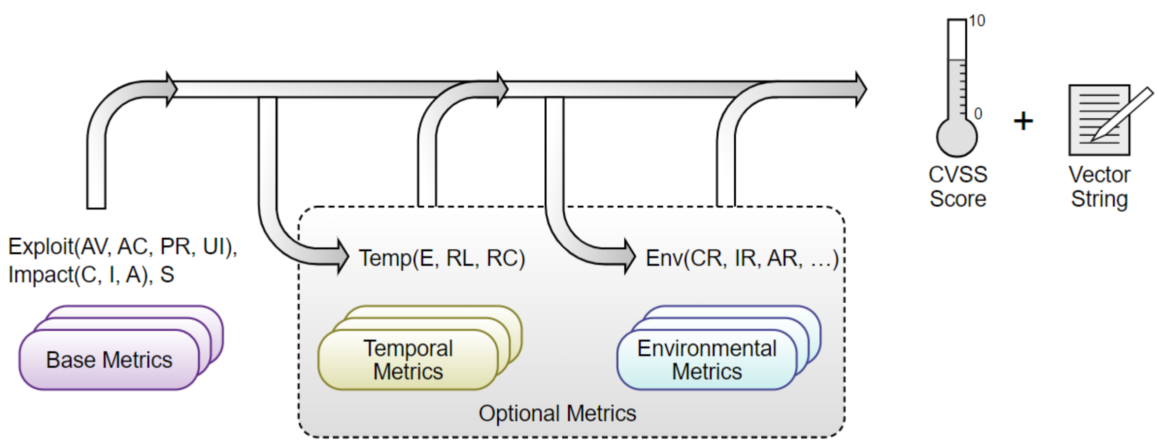


Fig. 1. Base, Temporal, and Environmental Scoring Progression (from [1])

2.1 CVSS Base Score Metrics

The CVSS base score for a vulnerability is calculated from the eight inputs shown in Figure 2. Four of them – attack vector (AV), attack complexity (AC), privileges required (PR), and user interaction (UI) – are labelled "exploitability metrics." These represent characteristics of the vulnerable object that reflect its ease of exploitability relative to the vulnerability being scored. Three of them – confidentiality (C), integrity (I), and availability (A) – are labelled "impact metrics." These represent the degree to which an impacted component may suffer due to a successful exploit of the vulnerability. The scope metric (S) evaluates whether successful exploitation of the vulnerability enables the attacker to cross a security or trust boundary when impacting components.

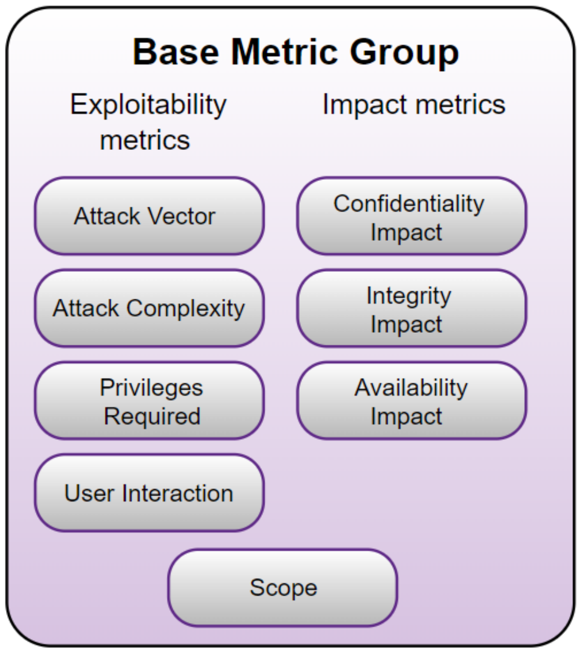


Fig. 2. CVSS Base Score Metrics (from [1])

Each of the eight metrics can be assigned one of a set of metric values. The metric values for each of the 8 metrics are shown in Table 1 along with a short description. These are more thoroughly defined in [1].

2.2 CVSS Base Score Equations

The CVSS v3 base score for a vulnerability is calculated by determining the qualitative metric value for each of the eight metrics, converting those qualitative values to numbers using the mapping in Table 2, and then inputting the eight numbers as input into the base score equation. Several online CVSS v3 calculators (e.g. [8] and [9]) are available to enable one to try out CVSS scoring.

Table 1. Metric Value Descriptions, CVSS v3

CVSS Metric	**Metric Value**	**Short Description**
Attack Vector	Network	Remotely exploitable
	Adjacent	Local network exploitable
	Local	Non-network attack on local host (e.g., through read/write/execute capabilities)
	Physical	Attack requires physical presence
Attack Complexity	Low	Attack can be launched at will
	High	Attack requires preparation and/or additional knowledge to be successful
Privileges Required	None	Attacker does not need prior privileges to launch the attack
	Low	Attacker must already have user level privileges
	High	Attacker must already have admin level privileges
User Interaction	None	No user interaction is required
	Required	User interaction is required
Scope	Unchanged	Attack can only effect resources within the security authority of the vulnerable component
	Changed	Attack can effect resources outside of the security authority of the vulnerable component
Impact Metrics (CIA)	High	Total loss
	Low	Some loss
	None	No loss

Table 2. Numerical Values for Base Score Metrics, CVSS v3

CVSS Metric	**Metric Value**	**Numerical Value**
Attack Vector	Network	0.85
	Adjacent	0.62
	Local	0.55
	Physical	0.2
Attack Complexity	Low	0.77
	High	0.44
Privileges Required	None	0.85
	Low	0.62 (or 0.68 if Scope is changed)
	High	0.27 (or 0.5 if Scope is changed)
User Interaction	None	0.85
	Required	0.62
Impact Metrics (CIA)	High	0.56
	Low	0.22
	None	0

Table 3. Qualitative Severity Rating Scale

Rating	CVSS Score
None	0.0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

The v3 base score equations are shown in Figure 3. Note that the base score is constructed from two sub-scores, impact and exploitability, that each respectively take as input the numerical values for the impact and exploitability metrics. Scope is a modifier at the base score level (it does not appear in the sub-scores).

The Base Score formula depends on sub-formulas for Impact Sub-Score (ISS), Impact, and Exploitability, all of which are defined below:

ISS =	**1 - [(1 - Confidentiality) × (1 - Integrity) × (1 - Availability)]**
Impact =	
If Scope is Unchanged	6.42 × ISS
If Scope is Changed	7.52 × (ISS - 0.029) - 3.25 × $(\text{ISS} - 0.02)^{15}$
Exploitability =	8.22 × AttackVector × AttackComplexity ×
	PrivilegesRequired × UserInteraction
BaseScore =	
If Impact \<= 0	0, *else*
If Scope is Unchanged	Roundup (Minimum [(Impact + Exploitability), 10])
If Scope is Changed	Roundup (Minimum [1.08 × (Impact + Exploitability), 10])

Fig. 3. CVSS v3 Base Score Equations (from [1])

The base score equations produce a score between 0.0 to 10.0. This range is historical, dates back to Version 1, and has been kept for consistency. The qualitative severity rating scale shown in Table 3 maps score ranges to qualitative labels and aids users in understanding the significance of a particular score. This mapping is more than just a user aid as it was used in the development of the equations (see Section 3.1).

3 Rationale for the CVSS Base Score Equations

Readers may find it challenging to understand the CVSS v3 base score equations in Figure 3 and the CVSS specification gives no explicit rationale for why they have this particular form. There is no explanation for why the constants and coefficients have those particular values, why the eight input variables have the numerical values specified in Table 2, or why there is a term raised to the 15^{th} power.

The fact that the form of the v3 equations is not explained (or may not have an explanation) does not invalidate them, but it does make validation an important task. Technology has often been engineered to work without knowing exactly why it works [10]. The equations then can be viewed as an opaque box – a machine – that produces an output given an input.

In order to test the consistency of the v3 base score equations, it is then necessary to perform experiments to determine if the opaque box (i.e., the equations) produces the desired output given a specific set of inputs. To do that, one needs to understand how the equations were developed and what the expected outputs are.

3.1 Development of the CVSS Base Score Equation

Between 2014 and 2015, the CVSS SIG leveraged human expert opinion to develop the CVSS v3 equations, as discussed in [1]. To create the equation, the SIG first identified a set of real vulnerabilities, and the properties of each vulnerability were evaluated to create an associated CVSS vector. CVSS SIG members then used expert knowledge to label each vector (representing a real vulnerability) with its severity: Low, MedLow, MedHigh, High, and Critical. The target score ranges from the previously discussed 'Qualitative Severity Rating Scale' provided in Table 3 were also leveraged. This defined a desired score range for each labeling of severity (e.g., "High" had a defined score range of 7.0 to 8.9). This labeling then defined a partial lookup table that mapped a subset of possible CVSS vectors to a target range of scores. Next, the SIG hired a contractor team to develop an equation to assign a score to each CVSS vector. Each score was to fall within the target score range within an acceptable deviation (see Section 3.2). Note that the contractors were given vectors mapped to five severity levels (i.e., Low, MedLow, MedHigh, High, and Critical) but only four non-zero target score ranges (i.e., Low, Medium, High, and Critical). To address this difference, the contractor team was given the discretion to best fit the MedLow vectors in either the Low or Medium bin and to place the MedHigh vectors in either the Medium or High bin.

The intuition behind this approach was that the produced v3 base score equation would appropriately score the rest of the vectors (having been essentially trained with the set of hand-evaluated vectors). After the equation was developed, extensive testing was performed to validate this assumption for a subset of the vectors that were not in the partial lookup table.

3.2 Acceptable Deviation

Unfortunately, the contractor was unable to formulate a v3 base score equation that strictly met the mapping requirements. Thus, it was necessary to develop a metric to measure such discrepancies, leading to the development of the metric "acceptable deviation". Acceptable deviation measures the worst case in which a hand-rated input vector deviates from its required scoring range. More precisely, it is the absolute value of the maximum difference between a hand-rated vector's score generated from the base score equation and the closest score within its required score range. Note that it does NOT mean that the scores are accurate within a range of +/- the acceptable deviation. For example, the acceptable deviation is 0 for a vector labeled as "High" with a score of 7.1. This is because 7.1 is within the score range for High of 7.0 - 8.9, per Table 3. The acceptable deviation is 0.4 for a vector labeled as "High" with a score of 9.3 because its score is 0.4 points higher than the top of the specified range for "High".

4 Metrology Tools, Metrics, and Algorithms

This section discusses the tools, metrics, and algorithms developed to support measurements of the CVSS v3 base score equation. Section 4.1 presents the NIST Knowledge Encoder tool which ingests and encodes human expert opinion as knowledge constraint graphs. Section 4.2 explains the idea of a knowledge constraint graph, and Section 5.4 discusses a metric to measure the level of inconsistency between multiple graphs encoded from different experts. Lastly, Section 4.4 presents the voting algorithm for unifying multiple graphs into a single unified graph. The tool, knowledge constraint graphs, inconsistency metrics, and voting unification algorithm will be used to collect and process the CVSS human expert domain knowledge discussed in Section 5.

4.1 Knowledge Encoder Tool

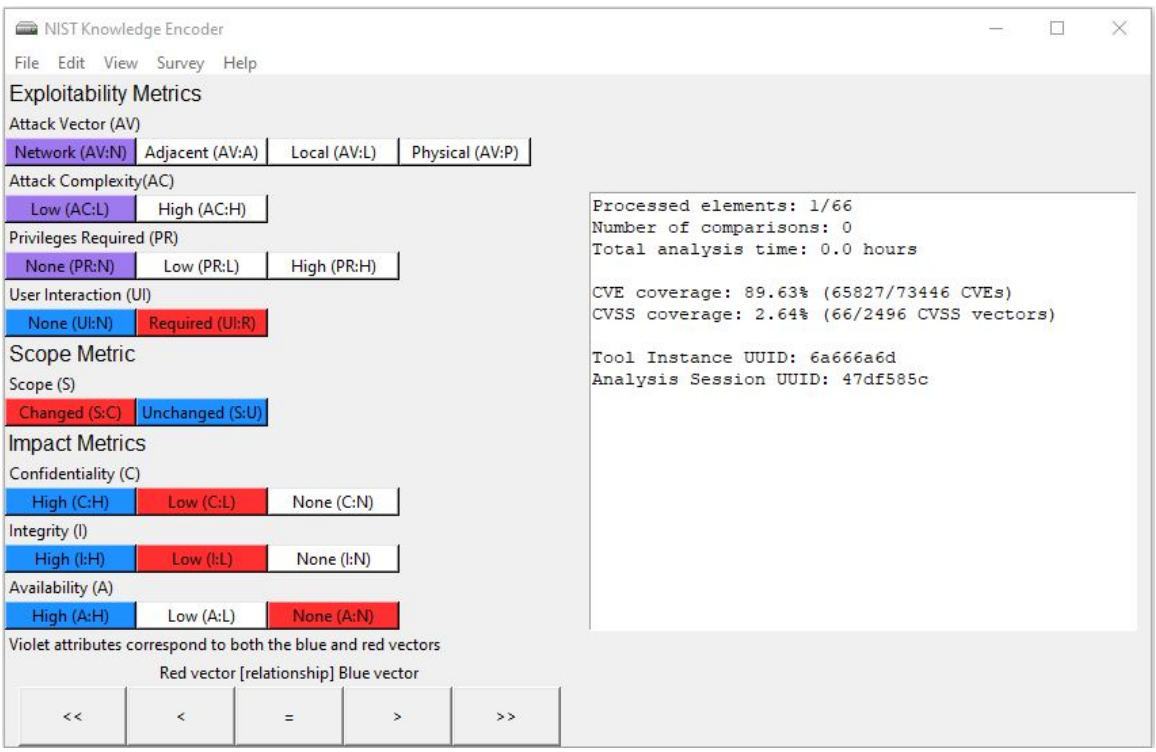


Fig. 4. CVSS Analysis Screen of the NIST Knowledge Encoder Tool

The NIST Knowledge Encoder tool was developed to encode the volunteers' domain knowledge. It is a Python program with a Tkinter graphical user interface (GUI). It uses the NetworkX Python package as a graph database in which to encode the extracted knowledge. An image of the main CVSS analysis screen is shown in Figure 4. Each participant of the study was provided with a copy of the tool source code, which they executed locally. The tool recorded their domain knowledge and then outputted the encoded knowledge as a graph.

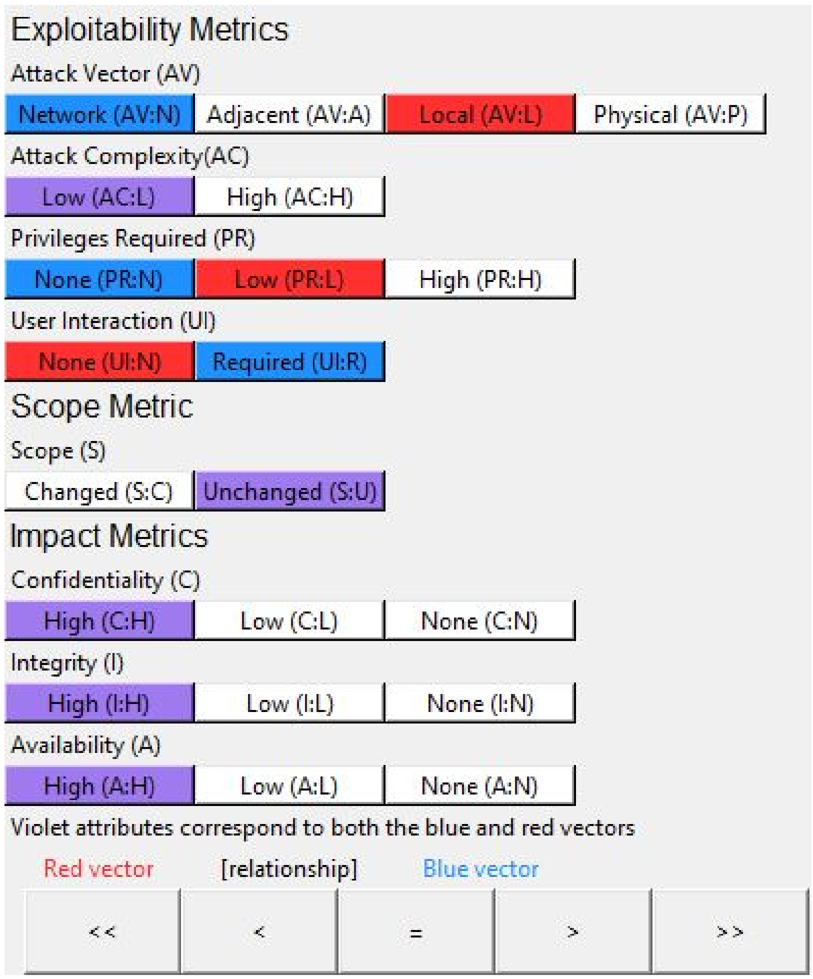


Fig. 5. CVSS Comparison Interface

The tool uses the interface shown in Figure 5 to iteratively present to the user pairs of CVSS vectors to compare – a “red” vector and a “blue” vector. The boxes in red represent the metric values for the red vector. The boxes in blue represent the metric values for the blue vector. The boxes in purple represent the metric values that apply to both the red and blue vectors. The metric value boxes for each of the eight metrics are arranged in order of decreasing severity to aid visual analysis. The user evaluates the metric values for the two vectors and then presses a button at the bottom of the interface to indicate the relationship of the red to the blue vector. They can specify ‘<<’ (much less than), ‘<’ (less than), ‘=’ (equal to), ‘>’ (greater than), and ‘>>’ (much greater than). The red vectors are drawn from a pool of not yet processed input vectors; the most frequently occurring within CVEs are chosen first (see Tables 9 and 10). Each blue vector is an already processed vector that represents 0 or more other vectors of equal severity.

Figures 4 and 5 show the four most popular CVSS vectors, per Tables 9 and 10 in Appendix B. In Figure 4, the red vector is CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N while the blue vector is CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H. Note that both share the same metric values for the first three metrics, making those metric value boxes purple in the figure. In Figure 5, the red vector is CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H while the blue vector is CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H. Unlike in the previous example, these two vectors differ in their "Attack Vector" metric value. Thus, for the red vector the box "Local (AV:L)" is highlighted red while for the blue vector the box "Network (AV:N)" is highlighted in blue. However, these two vectors also share five metric values resulting in the five boxes highlighted purple.

In the background the tool performs a modified binary insertion sort. The tool uses the traditional algorithm with the following modifications:

- The human makes the comparison decisions that are normally done by the computer
- The human can declare a vector being sorted as equal to a set of already ordered vectors
- The human defines the distance between compared vectors (e.g., greater than and much greater than).

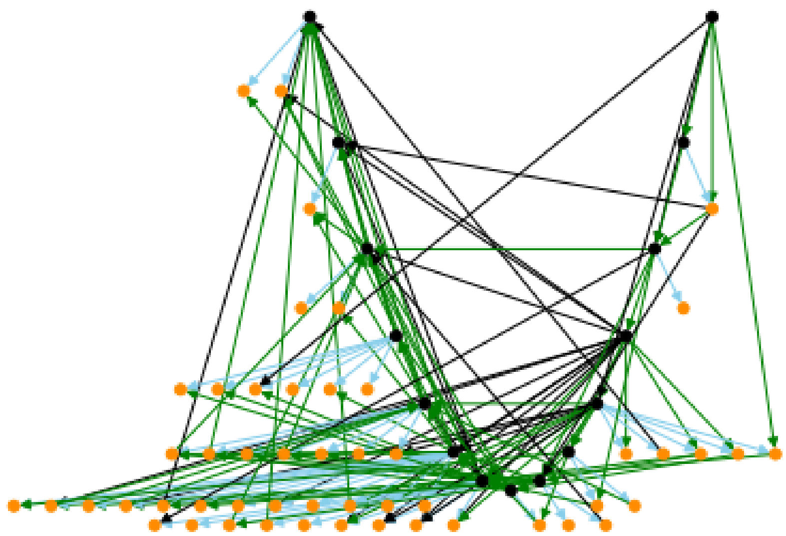
These modification result in an output that groups vectors into multiple sets where all members of a set are defined to have equal severity. It then totally orders these sets and provides distance constraints between each set. This output is recorded as a dot-and-line style graph with labelled edges, referred to as knowledge constraint graphs.

4.2 Knowledge Constraint Graphs

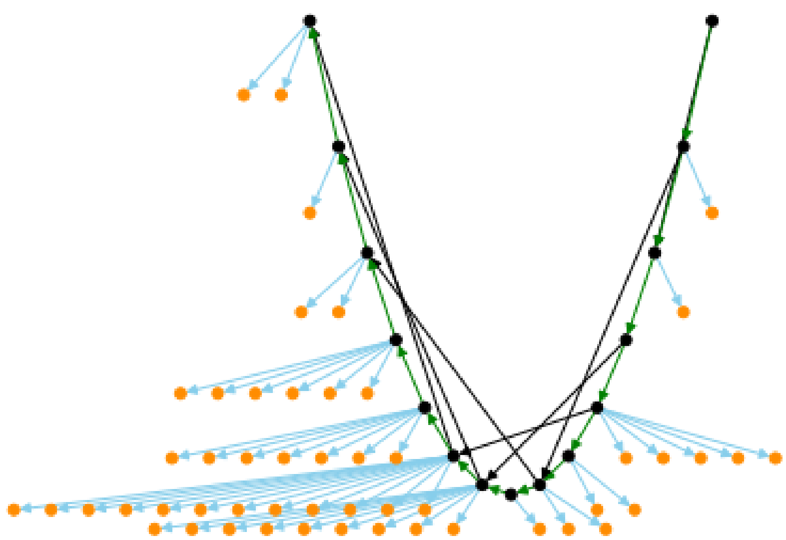
A knowledge constraint graph is a dot and line graph representation that orders a set of vectors and defines distance constraints between the vectors. Each node in the graph represents a vector and each labelled edge in the graph provides ordering and distance constraints for the connected nodes. The graphs are directed acyclic graphs (DAG).

Edges represent the distance constraints between nodes. Edges with a label of 0 represent equality (and are shown visually using light blue edges). Edges that represent greater than (or '>') have a label of 1 (and are shown visually using green edges). Edges that represent much greater than (or '>>') have a label of 2 (and are shown visually using black edges). Note that less-than and much-less-than edges are not added because they are represented by changing the direction of the edge.

Figure 6a shows an example knowledge constraint graph with 66 nodes and 166 edges that was produced from the encoding of human expert knowledge using the tool.



(a) Raw Graph 70a



(b) Simplified Graph 70a

Fig. 6. Example Knowledge Constraint Graph

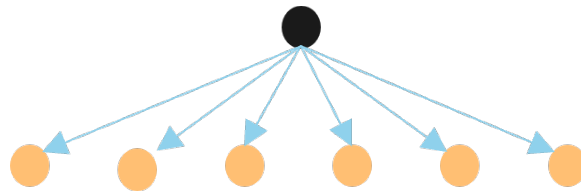


Fig. 7. Example Equivalency Set Star Sub-graph

4.2.1 Equivalency Sets

An important concept for constraint graphs is the idea of “equivalency sets”. An equivalency set is a set of nodes that are defined to have equal significance (i.e., should have the same CVSS score). They are represented as star sub-graphs; an example is shown in Figure 7. The parent node (the center of the star sub-graph) is the node in the equivalency set whose vector has the greatest frequency among a defined set of CVEs (see Tables 9 and 10). This node is called the “representative” node.

In a knowledge constraint graph, the representative nodes are displayed as black nodes. Other vectors that participate in equivalency sets are displayed as yellow nodes. Light blue edges represent equality and connect parent representative nodes to their children. Yellow nodes always have exactly one parent (through a light blue equality edge) as they can participate in only one equivalency set. Black nodes with no yellow node children represent equivalency sets of size 1.

4.2.2 Magnitude Measurements

Another important concept for constraint graphs is that of measuring the ‘magnitude’ of the distance between nodes. If two nodes are connected by an edge, the label on the edge defines the magnitude. Thus, an edge $x \rightarrow y$ with a label of 0 indicates that x is equal to y ($x = y$) in severity. An edge $x \rightarrow y$ with a label of 1 indicates that x is greater than y ($x > y$) in severity. An edge $x \rightarrow y$ with a label of 2 indicates that x is much greater than y ($x >> y$) in severity.

If two nodes x and y are not directly connected by an edge, then the magnitude is defined as the maximum magnitude of all edges on all paths between x and y. If there is no path between x and y, then the magnitude is undefined.

4.2.3 Simplified Graphs

Figure 6b is a simplified version of Figure 6a. All out-edges from the yellow nodes were changed to originate from their parent representative black node (found by traversing the one-per-node light blue edge backwards to find the parent). All in-edges coming into yellow nodes were changed to make their destination be their black node representative parent. Given that each parent black node represents an equivalency set where the black node is equal in significance to all of its child yellow nodes, this simplification does not change the logic represented by the graph. Lastly, all redundant edges are removed; if an existing path can represent the logic conveyed by a single edge, then the edge is removed.

Note how in Figure 6b there exists a single longest path that connects all of the equivalency sets by their representative black nodes. This feature is guaranteed to exist by the construction of the graphs. The first node on this path is the most significant vector (the one that should have the highest score). It is depicted in the upper right in all of the visualizations. Likewise, the least significant node is always on the upper left. Note also how each black edge is a shortcut for a longer path of green edges. This indicates that a path of '>' relationships may result in a '>>' relationship (which is intuitive).

4.3 Inconsistency Metrics for Knowledge Constraint Graphs

When multiple human experts use the tool, the produced constraint graphs can be evaluated to determine their level of inconsistency with each other. The purpose of performing such measurements is to identify possible outliers that might indicate either 1) a inexperienced participant that should not have participated in the study or 2) a valid but very divergent view on vector severity.

To measure inconsistencies, a pairwise approach was taken to compare all pairs of produced graphs. For each pair of graphs, the encoded relationships for all pairs of vectors were evaluated. In doing this, only the direction of the relationships was evaluated not their magnitudes. Thus, greater than and much greater than were treated equally. If the graphs agreed on the relationship for a pair of vectors, that pair was marked as "consistent". If the graphs disagreed on the relationship for a pair of vectors, that pair was marked as "inconsistent". If a pair of graphs disagreed on the direction of an inequality (i.e., one said greater than and the other less than), then that vector pair relationship was marked as 'opposite inconsistent' (a more severe form).

For each pair of graphs the number of "inconsistent" and 'opposite inconsistent' relationships was obtained (note that the set of opposite inconsistent pairs is a subset of the inconsistent pairs). Dividing those numbers by the total number of relationships results in ratios for each metric. This gave 'inconsistent' and 'opposite inconsistent' ratios for each pair of graphs. From this, the mean 'inconsistent' and 'opposite inconsistent' ratios for each graph could then be computed by taking the mean of the measurements in which a particular graph participated (since each measurement is for a pair of graphs).

4.4 Voting Unification Algorithm

This section discusses the algorithm for taking multiple knowledge representation graphs as input and unifying them into a single graph representing a consensus of the inputs.

4.4.1 Analysis of Votes

The voting algorithm will evaluate all ordered pairs (x,y) where the node number of x is less than y. Thus, for every pair (x,y), (y,x) is excluded because that would be redundant.

For each pair, votes will be tallied using a simple array $[a,b,c]$ to represent the number of input graphs for which $x < y$ (represented by a in the array), $x = y$ (represented by b in the array), and $x > y$ (represented by c in the array). Note that at this stage of the analysis, $>>$ is treated the same as $>$ and $<<$ is treated the same as $<$ (at this point, only the direction needs to be known, not the magnitude).

A transformation is then made to more accurately represent the $x = y$ votes. To see the need for this, consider the following example. A pair (x,y) may have a set of votes [4,2,4] (4 less than votes, 2 equal votes, and 4 greater-than votes). We want this to result in a decision for equal even though equal has the lowest number of votes. Each of the two votes that conflict (one greater than and one less than) are interpreted as really a vote for equal. Since the experts can't agree, the vectors are likely so close in significance that they should be marked as equal. To make this adjustment, anytime there exists a pair of opposing votes, one less than and one greater than, they are converted into a single vote for equal because that changes the difference between the less-than votes and equal votes by 1 and between the greater-than votes and the equal votes by 1. The transformation may be applied multiple times. In this example, [4,2,4] is transformed into [0,6,0] by applying the transformation four times. Consider another example where the transformation is applied to a vector pair with a set of votes [2,1,3]; this will also result in a decision for equal. The instances of both greater-than and less-than votes get transformed into equal votes that result in a final transformed vote tally of [0,3,1]. If in the final set of transformed votes there is a tie (e.g., [0,3,3]), the non-equal one is awarded the decision (either less than or greater than). These transformed vectors along with the ones that did not require any transformation are then fed into the prioritization stage of the algorithm.

4.4.2 Priority Ordering

The algorithm next orders all pairs of vectors by priority order (to be defined by three sorting approaches) such that the first pairs are those in which there is the most confidence in the experts' opinion and the last pairs are those in which we there is the least confidence.

The pairs are sorted first in descending order by the maximum number of votes received for the winning category (less than, equal, or greater than). For example, for a pair with votes [0,6,2] the maximum number of votes is 6 (for equal in this case). The intuition is that if a pair has a higher number of maximum votes then its decision is stronger (supported by more human experts) than a pair with a lower maximum number of votes. Thus, [6,4,0] is stronger than [0,5,5].

The authors considered applying this sort using the vector values prior to the equality transformation of conflicting votes (presented in Section 4.4.1). They decided against that approach because conflicting votes for ¿ and ¡ are not a sign of human certainty. This decision has a byproduct of increasing the certainty measurement for = votes, but this effect is limited (capped at half of the total number of possible votes) because a pair of opposing votes gets transformed into a single equal vote in the transformation.

For pairs with the same maximum value, there is a secondary sort in ascending order by the number of opposite votes in the original voting (prior to the transformation). The intuition is that pairs that have few opposite votes (votes for both less than and greater than) are considered to be supported more strongly by the experts than pairs with many opposite votes.

Finally, for pairs that have values that tie in both the first and secondary sort, there is a third sort added to guarantee a total ordering of the pairs. It gives priority to processing vector pairs that are most often seen in the wild. More specifically, each vector pair is sorted in descending order by the frequency of the vector in the pair that most frequently occurs within CVE in the NVD. Note that this third sort is rarely used and is not strictly necessary, but it conveniently removes non-determinism so that the algorithm will always produce exactly the same answer.

4.4.3 Unified Graph Construction

The unified knowledge constraint graph is constructed by iterating over the pairs in priority order and attempting to add edges based on the pair voting information. The unified graph is initially empty; nodes and edges are added as the algorithm evaluates each pair. Occasionally, the addition of an edge will violate the directed acyclic nature of the graph by creating a cycle. Those edges are not added; they represent lower priority (less certain) relationships that contradict higher priority (more certain) relationships. Cycles are not allowed because they would represent logical inconsistencies (e.g., $x > y > z > x$).

For each pair (x, y) the algorithm attempts to add an edge to the, initially empty, unified constraint graph based on the maximum vote calculation (i.e., for less than, equal, or greater than). If $x = y$, it adds an edge $x \rightarrow y$ with the label 0 (to represent equality). If $x > y$, one determines the magnitude of the relationship (see above) and adds an edge $x \rightarrow y$ with a label of 1 for greater than and 2 for much greater than. If $x < y$, one determines the magnitude of the relationship (see above) and add an edge $y \rightarrow x$ (note the reversal of the order of x and y) with a label of 1 for greater than and 2 for much greater than.

In some circumstances the graph construction algorithm may rearrange edges in order to simplify the graph but the encoded logic is always preserved. For example, if a set of vectors are all equal, the algorithm will form a star sub-graph of edges representing equality as opposed to creating a path of edges representing equality (this is for simplicity of the visualization, but it also helps in writing the graph algorithms that assume certain graph structures).

4.4.4 Description of Constructed Graph

Constructed unified graphs have the same form as simplified raw graphs; in other words, they look the same (see Figure 6b as an example). A constructed unified graph usually totally orders the input vectors but is not guaranteed to do so, especially in the presence of contradictory and/or inconsistent expert opinion. However, the unified graph will have a longest path of edges labelled with either 1 or 2 (greater than or much greater than). Each node on this longest path will represent an equivalency set – a set of nodes that were defined to be of equal significance. To represent the equivalency sets, each node on the longest path is at the center of a star sub-graph, constructed with edges labelled 0 where each child node is equal to the representative parent (the center of the star). If a node on the longest path is not equal with any other node, its star graph will be size 1 (containing just itself).

5 Data Collection and Processing

This section discusses how human expert opinion was collected and processed in order to create unified knowledge constraint graphs. Sub-section 5.1 discusses the dataset of analyzed vectors while Sub-section 5.2 describes the pool of volunteer analysts. Sub-section 5.3 presents the produced individual analyst knowledge constraint graphs. Sub-section 5.4 provides the measurements of inconsistency taken on analyst data. Sub-section 5.5 presents the unified knowledge constraint graph built from all analyst data. Sub-section 5.6 concludes the section by discussing how the number of equivalency sets identified in the unified graph does not represent the discovery of some optimal number.

While this section focuses on the unified knowledge constraint graph using all inputs, many such unified graphs will be created using differing subsets of the input data for statistical reasons (i.e., differing subsets of input knowledge constraint graphs).

5.1 Data Set of Analyzed Vectors

For this research human experts were asked to analyze 66 of the 2496 CVSS v3 vectors that had a non-zero impact (2.64 % of them). Note that there are 2592 vectors in total but only 2496 have a score other than 0.0. The vectors chosen were those that the NVD mapped the CVE vulnerabilities to most frequently, using the NVD CVSS data available on 2021-01-08. This set of 66 vectors covered 90 % of the CVEs. The 66 vectors chosen are shown in Appendix B, in Tables 9 and 10, along with their respective frequencies.

5.2 Volunteer Participants

The CVSS v3 equations were designed to represent human expert knowledge, in particular CVSS SIG knowledge. Thus, to measure how well the equations reflect current CVSS SIG domain knowledge, the domain knowledge of a group of 12 volunteers from the CVSS SIG membership of 2021 was leveraged. The 12 volunteers are the domain expert co-authors as well as the second author. The first author was the principle investigator.

To support this research, the CVSS SIG domain experts each represented their domain knowledge of computer vulnerability types as a mathematical graph structure. In doing so, the domain experts compared vulnerabilities using the CVSS philosophy of evaluating a vulnerability's severity in general to the world apart from any particular installation environment. This was an attempt to mitigate the possibility that the domain experts would be influenced by their particular security domain or specialty. Additionally, the volunteers were instructed to compare vulnerabilities based on their own personal expert opinions (not based on the existing CVSS scoring). This was an attempt to eliminate bias based on the expert's knowledge of the CVSS scores for certain vectors and/or use of CVSS calculators.

The human studies portion of this research was conducted with the approval of the NIST Research Protections office under the study entitled "Metrics Generation with the NIST Human Knowledge Encoder Toolkit" (Study #: ITL-2020-0227).

Table 4. Statistics on CVSS SIG Produced Knowledge Constraint Graphs

Graph	Nodes	Raw Graph Edges	Simplified Graph Edges	Analysis Time (hrs)
02c	66	194	67	3.8
3d6	66	242	72	6.3
5fd	66	236	69	1.9
6e5	66	256	69	5.5
70a	66	166	72	2.1
88d	66	228	70	8.1
908	66	247	72	1.4
977	66	142	67	0.7
98a	66	284	68	6.5
d3d	66	186	69	1.7
f00	66	187	70	1.5
f59	66	224	69	2.5
Overall Mean	66	216	69.5	3.5

5.3 Produced Knowledge Constraint Graphs

The 12 domain experts each produced a knowledge constraint graph that represented their CVSS domain knowledge using the NIST Knowledge Encoder tool. These graphs are provided in Appendix C. Table 11 contains the raw graphs and Table 12 contains the corresponding simplified graphs where the redundant edges have been removed.

The mean creation time for the set of graphs was 3.5 hours with a minimum of 0.7 and a maximum of 8.1. The number of nodes for all graphs is 66 because there were 66 vectors analyzed. The number of edges varies because the humans ordered the nodes differently as they made decisions for the human-directed binary search algorithm. The mean number of edges for the raw graphs is 216 with a minimum of 142 and a maximum of 284. The mean number of edges for the simplified graphs is 69.5 with a minimum of 67 and a maximum of 72. The statistics for each graph are provided in Table 4.

5.4 Knowledge Constraint Graph Inconsistency Measurements

The inconsistency and opposite inconsistency of the 12 knowledge constraint graphs were analyzed. These metrics were defined in Section 5.4. The results are shown in Table 5. The overall mean inconsistency was 22.5 % and the opposite inconsistency was 14.4 %. Thus, the human experts were in general agreement, although there were certainly differences for certain pairs of vectors.

Table 5. Mean Inconsistency and Opposite Inconsistency Results

Graph	Mean Inconsistency Percent	Mean Opposite Inconsistency Percent
02c	20.8	11.5
3d6	17.1	10.3
5fd	20.8	13.8
6e5	19.1	13.0
70a	25.1	13.5
88d	20.7	13.8
908	20.9	14.2
977	35.2	22.7
98a	21.1	14.7
d3d	25.2	16.2
f00	25.8	17.5
f59	19.7	11.2
Overall Mean	22.5	14.4

Table 6. Vectors Initially Assigned the Highest Severity in the Unmodified Graph f00

CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N
CVSS:3.1/AV:N/AC:L/PR:H/UI:R/S:C/C:L/I:L/A:N
CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H

Table 7. Vectors Initially Assigned the Lowest Severity in the Unmodified Graph f00

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

5.4.1 Graph f00

Graph f00 (Figure 11k) was an extreme outlier that was discovered to have a significant but correctable error. Its initial mean inconsistency was 82.1 % and opposite inconsistency was 73.8 %. Upon inspection, it was discovered that the analyst creating f00 with the tool did all of their ratings backwards. To fix this, the edges in their graph were simply reversed (and checked with the participant); the resulting mean inconsistency metric then dropped to 25.8 % and opposite inconsistency to 17.5 %. The opposite ratings became obvious by looking at the vectors that they rated the most severe and those that they rated as least severe (see Tables 6 and 7.

5.4.2 Graph 977

After fixing graph f00, graph 977 (Figure 11h) was the most significant outlier. Its mean inconsistency and opposite inconsistency was 35.2 % and 22.7 %; this was the greatest among the graphs (see Table 5). While these ratios were not as excessively high as the original graph f00, they – combined with the fact that the participant spent only 43 minutes on the analysis – induced concerns about data quality (the mean analysis time for all analysts was 3.5 hours). To address this, the participant offered to perform their analysis again, this time with greater care. The analyst spent 48 minutes the second time and produced graph 382 (not shown). Supporting the validity of the original graph 977, graph 382 had mean inconsistency metrics that were very similar to 977 (32.7 % and 21.2 %). Unfortunately however, graphs 977 and 382 were inconsistent between themselves (27.9 % inconsistent and 13.3 % opposite inconsistent).

Uncertain of how to proceed with this, a complete set of evaluation metrics was run three times and the final overall results were compared (using all analyst input). For the three trials, graph 977 was used first, followed by graph 382, and then a graph generated by unifying graphs 977 and 382 using the voting algorithm. Fortunately, the final results varied little for the three trials (the variation in the primary measurement statistics were at most .04); this is attributed to the voting algorithm smoothing out discrepancies since there were a total of 12 graphs voting. Since it did not matter which of the three graphs was used and to avoid any possible perception of inappropriately manipulation of the input data, the originally submitted graph 977 was used in the experiments. Graph 382 as well as the generated unified graph (that had combined graphs 977 and 382) were discarded.

5.5 Unified Knowledge Constraint Graph

The 12 CVSS SIG knowledge constraint graphs, created using with the tool from Section 4.1, were combined into a single unified constraint graph using our voting algorithm from Section 4.4. This unified graph is shown in Figure 8. It has 66 nodes, each reflecting the 66 analyzed vectors. It has 71 edges that order the equivalency sets, define members within equivalency sets, and provide distance constraints. There are 16 equivalency sets; the smallest is 1 vector and the largest is 12 vectors. The longest path is 16 which traverses the

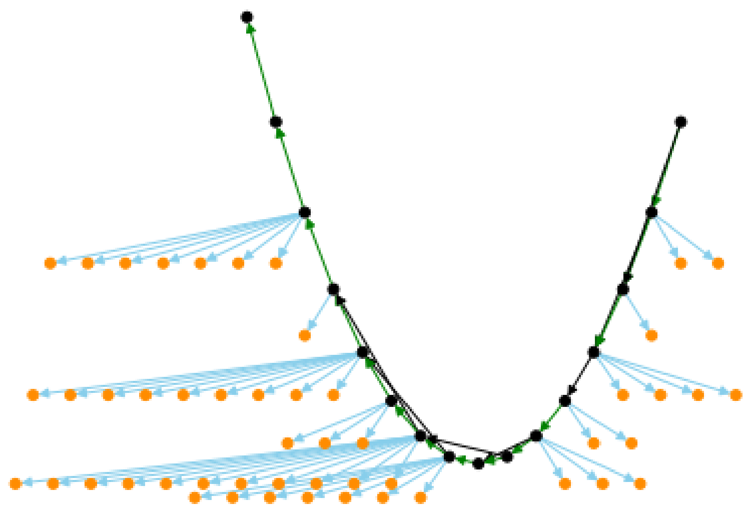


Fig. 8. Unified Knowledge Constraint Graph

representative nodes for each equivalency set. The 7 black edges represent much-greater-than relationships; the 14 green edges represent greater-than relationships, and the 50 light blue edges represent equality. While not guaranteed by the voting algorithm, this graph totally ordered the equivalency sets. In creating this graph, 130 of the 2145 proposed edges (6.1 %) were discarded due to lower confidence relationships that contradicted previously added higher confidence relationships. This is explained in Section 4.4.3.

5.6 Optimal Number of Equivalency Sets

One may ask if the 16 equivalency sets in the unified graph indicate the discovery of some optimal number of equivalency sets for CVSS, but this is not the case. The number of equivalency sets grows with the number of vectors analyzed. It might plateau at some optimal number but this research effort does not have sufficient data to evaluate that. What it can show is that for up to 66 vectors, an increasing number of vectors analyzed results in an increasing number of equivalency sets generated. This can be seen in Figure 9. The small dots of different colors represent the individual knowledge constraint graphs created from the tool from each human expert with a specific number of input vectors. The lines of small dots higher up show analysts that rarely used the equal button. The larger black dots toward the bottom represent the unified knowledge constraint graphs generated using all input graphs and an increasing number of input vectors (from 1 to 66). For comparison with CVSS v3, note that CVSS was designed using just five equivalency sets (i.e., the qualitative severity levels: None, Low, Medium, High, and Critical).

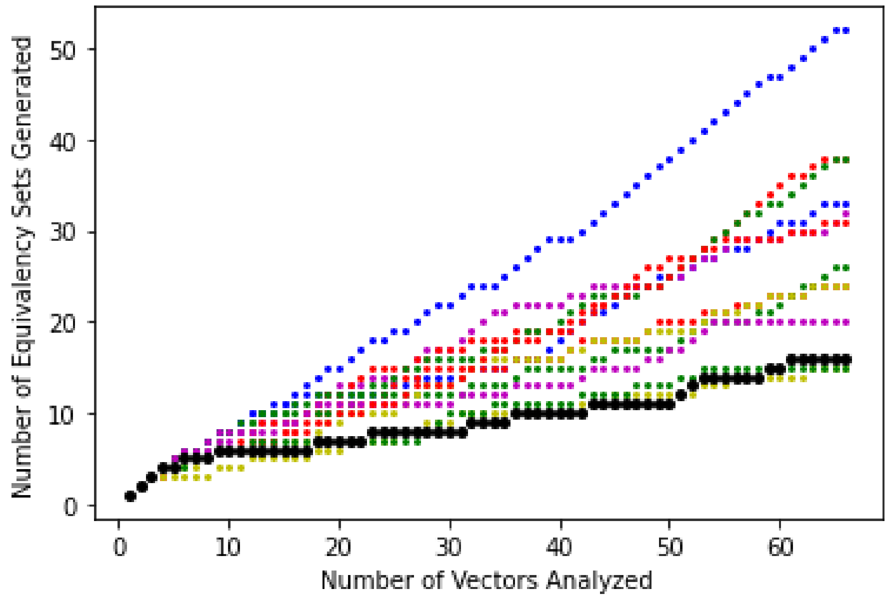


Fig. 9. Equivalency Sets Produced per Number of Vectors Analyzed (legend: large black dots are for the unified graph, and small colored dots are for individual analysts graphs)

6 Measurement Approach

This section discusses a general metric-agnostic approach to measuring the inconsistencies between the scores in CVSS v3 relative to the encoded CVSS SIG domain knowledge. This approach will be applied to three different metrics and the results provided in Section 7.

6.1 Consistent Scoring Systems

This subsection defines the terms "scoring system" and "consistent scoring system".

6.1.1 Scoring System Definition

For the purposes of this work, a "scoring system" is defined as a mapping of vectors to scores. Given any CVSS vector, a scoring system produces a score for that vector. CVSS v3 is an important example of one of many possible scoring systems.

6.1.2 Consistent Scoring System Definition

This work defines a 'consistent scoring system' as a scoring system that conforms to a particular knowledge constraint graph. Scoring systems may or may not be consistent with a constraint graph. For a scoring system to be consistent with a graph, the scores assigned to each vector must satisfy the constraints defined by the edges in the graph (both the direction and magnitude of the edges in a path between vectors). Each edge defines a direction between two vectors x and y and a relationship ($>$, $>>$, or $=$).

If an edge $x \rightarrow y$ is labelled $>$, then the scoring system must map x to a score that is greater than y. If an edge $x \rightarrow y$ is labelled with $>>$ (much greater than), then the value of x must be greater than the value of y by some constant associated with the graph. If an edge $x \rightarrow y$ is labelled with $=$, then the scoring system must map x and y to the same score. Note that the label $<$ never appears on an edge because it is not necessary; the direction of the edge represents the direction of the inequality.

If there is no direct edge between vectors x and y in a constraint graph, the relationship is the greatest from the set of relationships on the path of edges between x and y. For example, if there is a path of four edges from x to y with relationships $>$, $>$, $>>$, and $=$ then the defined relationship from x to y will be $>>$ (the greatest on the path). If there is no path from x to y then the relationship is undefined (this does not happen in this study as all graphs are totally ordered).

6.2 Generation of a Closest Consistent Scoring System

To generate a consistent scoring system for a particular graph, a greedy algorithm was developed. The algorithm takes a constraint graph and the CVSS v3 scores for the 66 analyzed vectors as input. It iteratively operates on individual equivalency sets (sets of nodes required by the constraint graph to have equal values) in order of decreasing size. Thus, for the unified constraint graph representing all 12 expert inputs (see Figure 8), it operates on the following 16 equivalency sets of varying sizes (in descending order): 12, 10, 8, 8, 5, 4, 4, 3, 3, 2, 2, 1, 1, 1, 1, 1. For each equivalency set, it calculates the mapped score for the vectors in the set to be the median of the CVSS v3 scores for those vectors. If the computed value is higher than the maximum allowed per the constraint graph given the scores already assigned for the vectors in the graph, the computed value is reduced to the nearest value that is consistent with the graph. An analogous operation is done to increase scores that are below the minimum allowed value. The output of the algorithm is a scoring system – an assignment of each vector with a score that is consistent with the input constraint graph.

Note that the greedy algorithm is designed to minimize the mean distance between the chosen score and the CVSS v3 scores for vectors within an equivalency set. Unintuitively, it uses the median (not mean) of a set of CVSS v3 scores because the median can be proven to minimize the sum of the differences (i.e., using median in the algorithm minimizes the mean of the sum of scoring differences) [11].

The code also uses another heuristic that minimizes the maximum distance between the chosen score and the CVSS v3 scores for vectors within an equivalency set. For this, instead of choosing the median value for the set of CVSS v3 scores in an equivalency set, it chooses the mean of the maximum and minimum value. This reduces the maximum distance because it minimizes the distance to the greatest outliers.

Note that in generating a closest consistent scoring system, the heuristic that will provide the best results given the metric currently being measured is used. This decision is discussed more in Section 6.3 and Section 7.

6.3 Measurement Methodology

Given some measurement metric (three are evaluated in Section 7), all 12 input constraint graphs are taken from our 12 CVSS SIG domain experts and are used to create a unified knowledge constraint graph. With this graph, a closest consistent scoring system using the algorithm described in Section 6.2 is generated. That closest consistent scoring system is then used as input to the measurement metric along with the CVSS v3 scores in order to calculate the result.

Note that the heuristic chosen will be the one that minimizes the metric being evaluated. A large number of consistent scoring systems usually exist, and we want to find the one (using whatever methodology) that is closest to CVSS v3 for the particular metric being measured. One could use any consistent scoring system, but such a measurement would be an upper bound that could be lowered by finding a closer consistent scoring system.

A source of error in performing measurements this way is the possibility that the particular unified knowledge constraint graph used just happens to allow for a scoring system close to CVSS v3. It could be possible that a slightly different set of inputs into the voting algorithm could have resulted in a worse measurement. Since it is not possible to obtain multiple sets of 12 inputs to test this for each metric, this issue is addressed by performing additional measurements using all combinations of 11 of the 12 inputs to create 12 unified knowledge constraint graphs. Each metric is then independently evaluated on all 12 unified graphs. From these 12 measurements, a mean result and standard deviation can be calculated. This gives the ability to calculate the precision of the measurements.

7 Measurement Results

This section measures the inconsistency of the CVSS v3 base score equation relative to the encoded CVSS SIG domain knowledge. The approach presented in Section 6 is used to perform three measurements: mean scoring distance, maximum scoring distance, and acceptable deviation. Table 8 contains all measurement results. These results are explained in Sections 7.1, 7.2, and 7.3. Section 8 interprets these results.

Table 8 provides the results for both heuristics presented in Section 6.2 for all three evaluated metrics. As discussed in Section 6.3, the "Mean" heuristic compares the CVSS v3 scoring system with the consistent scoring system whose scores minimize the mean differences between the scores of the two systems. The "Max" heuristic compares the CVSS v3 scoring system with the consistent scoring system whose scores minimize the maximum differences between the scores of the two systems. Both approaches provide upper bound measurements, so either could have been chosen for this work. Both are presented because the bounds for the three metrics can be slightly optimized by optimizing on the mean scoring distance for the mean scoring distance measurement and optimizing on the maximum scoring distance for the maximum scoring distance and acceptable deviation measurements. These optimized results are shown in bold in Table 8.

It is important for the reader to understand that these bolded results came from comparing CVSS v3 with two different consistent scoring systems (two that were closer to CVSS in different ways). While the authors defend this approach as being correct, this may cause discomfort with some readers due to the complexities involved; these are not simple measurements despite their surface simplicity. Readers who are uncomfortable with this measurement approach should simply use the results for the heuristic that minimizes the maximum scoring distance (labelled "Max"). Doing so compares CVSS v3 with a single consistent scoring system and provides a usable upper bound very close to what is achieved with this approach. Roughly the same results are obtained and the same conclusions are drawn using either metrology approach.

7.1 Mean Scoring Distance

Mean scoring distance measures on average how far off each CVSS v3 score is from the closest score consistent with the encoded domain knowledge. More precisely, for each vector evaluated by the CVSS SIG analysts, calculate the absolute value of the difference between the CVSS v3 score and the score assigned by the closest consistent scoring system (using the heuristic to minimize mean distance). The mean scoring distance is the mean of these values.

Using the unified knowledge constraint graph (i.e., using all 12 CVSS SIG domain knowledge graphs as input), CVSS v3 was found to have a mean distance of 0.13. Performing the calculation on a set of 12 knowledge constraint graphs, each formed from 11 of the 12 input graphs, CVSS v3 has a mean distance of 0.13 points with a standard deviation of 0.02 points. If one assumes a 'normal' Gaussian distribution,there is a 95 % chance that the actual distance is between 0.11 and 0.15 points.

Table 8. Measurement Results for Mean Scoring Distance, Maximum Scoring Distance, and Acceptable Deviation

Metric	Heuristic	# Inputs	# Trials	Result	Std Dev
Mean scoring distance	Mean	11	12	**0.13**	**0.02**
Mean scoring distance	Mean	12	1	**0.13**	**0**
Mean scoring distance	Max	11	12	0.18	0.02
Mean scoring distance	Max	12	1	0.17	0
Max scoring distance	Mean	11	12	0.70	0
Max scoring distance	Mean	12	1	0.70	0
Max scoring distance	Max	11	12	**0.52**	**0.15**
Max scoring distance	Max	12	1	**0.40**	**0**
Acceptable deviation	Mean	11	12	0.18	0.06
Acceptable deviation	Mean	12	1	0.20	0
Acceptable deviation	Max	11	12	**0.17**	**0.06**
Acceptable deviation	Max	12	1	**0.20**	**0**

7.2 Maximum Scoring Distance

Maximum scoring distance measures the maximum distance that any CVSS v3 score is from its closest score consistent with the encoded domain knowledge. More precisely, for each vector evaluated by the CVSS SIG analysts, calculate the absolute value of the difference between the CVSS v3 score and the score assigned by the closest consistent scoring system (using the heuristic to minimize maximum distance). The maximum scoring distance is the maximum of these values.

Using the unified knowledge constraint graph (i.e., using all 12 CVSS SIG domain knowledge graphs as input), CVSS v3 was found to have a maximum distance of 0.40. Performing the calculation on a set of 12 knowledge constraint graphs, each formed from 11 of the 12 input graphs, CVSS v3 has a maximum distance of 0.52 points with a standard deviation of 0.15 points. If one assumes a ‘normal’ Gaussian distribution,there is a 95 % chance that the actual distance is between 0.32 and 0.82 points.

7.3 Acceptable Deviation

The CVSS Version 3.1 specification contains a measurement of scoring error called acceptable deviation. It asserts that the acceptable deviation for the CVSS v3 scoring system is 0.5 points (maximum distance from a severity boundary).

Acceptable deviation is defined in Section 3.2. To measure it, the method in Section 6.3 was used as with the previous two measurements. It required not just a mapping of vectors to scores but also of scores to bins using the mapping from the CVSS v3.1 specification (shown in Table 3). To obtain the measurement for each vector evaluated by the CVSS SIG analysts, the deviation was calculated as the distance that a CVSS v3 score is from its vector’s specified bin. The acceptable deviation is the maximum of these deviations.

Using the unified knowledge constraint graph (i.e., using all 12 CVSS SIG domain knowledge graphs as input) and using the heuristic to minimize maximum distance (in this case, both heuristics worked equally well), CVSS v3 was found to have an acceptable deviation of 0.20 points (i.e., distance from a severity level boundary).

Note that in doing this calculation, any vector whose scores (for both the generated consistent scoring system and the CVSS v3 scoring system) map to the same bin have no deviation associated with them. Of the 66 vectors, 65 had no deviation. This means that, according to the encoded domain knowledge, they were assigned scores that mapped the vector to the correct bin. The one vector with a deviation was AV:A/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H. Its closest consistent scoring system score was 7.2 which mapped it to the "High" bin (per Table 3). The CVSS v3 score is 6.8, which is in the "Medium" bin. Since the score range for "High" is 7.0-8.9, the CVSS v3 score is a 0.2 distance from the "High" bin (resulting in a deviation of 0.2 points). Thus, the CVSS v3 scoring of vector AV:A/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H was responsible for the acceptable deviation of 0.2 points (otherwise, it would have been 0).

Next, using the 12 knowledge constraint graphs, each formed from 11 of the 12 input graphs, CVSS v3 was calculated to have an acceptable deviation of 0.17 points with a standard deviation of 0.06 points. If one assumes a 'normal' Gaussian distribution, there is a 95 % chance that the actual acceptable deviation is between 0.05 and 0.29 points.

7.4 Increasing Accuracy with More Data

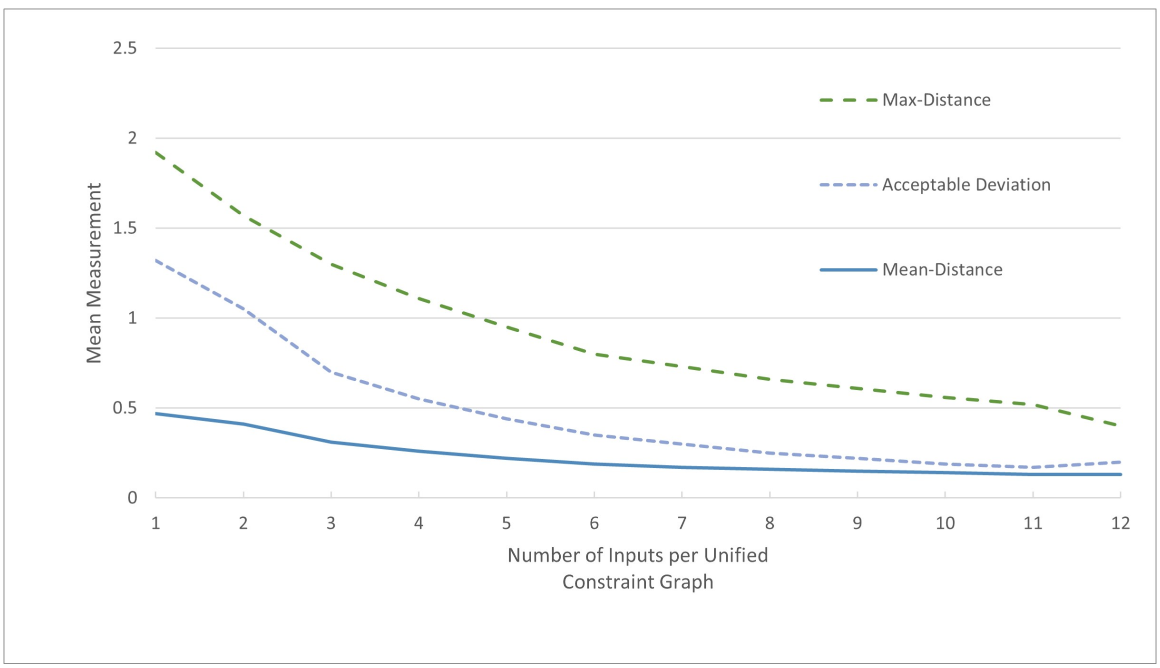


Fig. 10. Decreasing Error with an Increasing Number of Inputs

In performing these three measurements, it was empirically discovered that greater accuracy is achieved through having a greater number of expert participants inputting data into the voting algorithm. This can be seen in Figure 10. To create this figure, for each x-axis value 12 combination x experiments were performed using all combinations of the available inputs. Thus, for the x-axis value of 5, 792 experiments were performed (12 combination 5).

The measured mean metrics tend lower as the number of inputs into the unified constraint graphs used to perform the measurements increases. This follows "wisdom of the crowds" research that shows that human error in making group decisions often decreases when using a larger set of humans [12] [13]. More analysts should then produce more accurate results (enabling the voting algorithm to better eliminate rating mistakes made by particular individuals).

The curves eventually level off indicating a diminishing benefit to using additional analysts. This makes sense because even if all human error is eliminated in performing the measurement, what will remain is the actual measurement of the CVSS v3 scoring system. From the figure, it appears that the y-axis plateau value for both the mean mean-distance and mean acceptable deviation were achieved as the curves end in almost a horizontal line. For the mean max-distance, additional analysts would likely lower the measurement of distance somewhat. Unfortunately, additional qualified CVSS SIG analysts could not be obtained.

8 Interpretation of Results and Related Work

A variety of related work has explored perceived flaws in CVSS and recommended improvements. A subset of these enumerated flaws relate to the v3 base score equation itself. The results here address many of these concerns.

One of the best listings of perceived flaws in CVSS is [14], which also contains suggestions that could be used to improve and/or revise CVSS or to create alternate scoring systems. One concern is that in CVSS v3, the metric values are ordinals (ordered categories) but they are converted into ratio data (allowing numerical differences with a zero value) within the v3 base score equation. The CVSS specification provides no justification for the assigning of numerical values to these ordinal values (e.g., Attack Vector Adjacent = 0.62). It also provides no justification for how the particular numerical values were chosen. By assigning numbers, difference relationships are established not only between ordinal values of a particular CVSS metric (e.g., privileges required), but between ordinal values of different unrelated metrics (e.g., confidentiality and attack complexity). Additionally, [14] points out that it provides no justification for the equation that then takes these numerical values as input. Although not mentioned in [14], many have questioned the complexity of the equation and why, for example, it has a term raised to the 15$^{\text{th}}$ power. Combining these concerns, [14] points out that the CVSS specification makes claims like "faster + fastest = 6" for which there is no empirical or theoretical justification. In summary, [14] says that the CVSS specification provides "little transparency on the formula creation process". Other critiques of CVSS expressing concern about the equations include [15], [16], [17], [18], [19], [20], and [21].

The authors agree that such math is invalid in most cases. The formula creation process was opaque; the specific form of the v3 base score equation is not justified; and the equation is not human understandable. The improvement proposals in [14] and in the other critiques represent laudable goals. This said, the unjustified ratio math is acceptable if the use of the CVSS v3 scores is limited to creating an ordinal ranking of the vectors. This works in most cases as IT security organizations want to know how a particular vector ranks in severity compared to other vectors. The equation then becomes a black box that does not need to be justified or explainable. It simply needs to be tested to make sure that it produces the desired output ordinal rankings. This should not discourage its use as many effective computations are opaque boxes.

If one takes a step back to ask, "does the v3 base score equation do what it claims to do?", this research demonstrates that it does capture expert opinion within the "acceptable deviation" stated by the specification (measured at .2 versus the .5 advertised in the specification). However, the authors note that the acceptable deviation metric is not ideal due to its unintuitive definition and its focus on the optional binning from Table 3. For this reason, the metrics of mean and maximum scoring distance were added. The results for these two metrics enable a better understanding of the accuracy of the CVSS scores in representing the CVSS expert domain knowledge. As shown in the results from Table 8, CVSS v3 has a mean scoring distance of .13 and a maximum scoring distance of .4 using the full input dataset. The CVSS v3 scores are very close to a set of scores completely consistent with the encoded human expert opinion (at least relative to the expected differences represented by the acceptable deviation of 0.5 in the specification).

While the CVSS v3 equation represents the CVSS SIG expert domain knowledge very closely, it still does not represent it perfectly. The reason for this is the use of the generated equation. As stated previously, the goal of the equation is to approximate a partial lookup table. It achieves this goal to a measurable level for the set of 66 analyzed vectors (as seen by the measurements of mean and maximum scoring distance). One might ask why CVSS does not simply use a lookup table instead of a confusing equation. The answer is that the equation enables the scoring of all CVSS vectors, not just the ones that were human-evaluated. The equation strives to project CVSS SIG domain knowledge from a small analyzed set to the complete set. This said, the accuracy of this projection to the applicable 2430 non-analyzed vectors has not been formally evaluated either in the CVSS v3 specification nor in this work.

9 Conclusion

This work evaluated the CVSS v3 base score equation and determined that its scores conform to the acceptable deviation stated in the specification relative to the encoded CVSS SIG domain knowledge. Furthermore, the authors added the metrics of mean and maximum scoring distance to find that the scores themselves (apart from any binning) are very close to a set of scores completely consistent with the encoded human expert opinion. The base score equation effectively reflects CVSS SIG human expert opinion (to the extent shown by these measurements).

References

[1] (2019) Common vulnerability scoring system v3.1: Specification document, https://www.first.org/cvss/v3.1/specification-document [accessed 20211123].
[2] National infrastructure advisory council, https://www.cisa.gov/niac [accessed 20211123].
[3] Chambers J, Thompson J, Noonan T, Web M (2004) National infrastructure advisory council, common vulnerability scoring system, final report and recommendations by the council, https://www.cisa.gov/sites/default/files/publications/niac-common-vulnerability-scoring-final-report-10-12-04-508.pdf [accessed 20211118].
[4] Global forum of incident response and security teams, https://www.first.org/ [accessed 20211123].
[5] Shiffman M (2005) Complete cvss v1 guide, https://www.first.org/cvss/v1/guide [accessed 20211123].
[6] Mell P, Scarfone K, Romanosky S (2007) A complete guide to the common vulnerability scoring system version 2.0, https://www.first.org/cvss/v2/guide [accessed 20211123].
[7] (2015) Common vulnerability scoring system v3.0: Specification document, https://www.first.org/cvss/v3.0/specification-document [accessed 20211123].
[8] Common vulnerability scoring system version 3.1 calculator, https://www.first.org/cvss/calculator/3.1 [accessed 20211123].
[9] Common vulnerability scoring system calculator, https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator [accessed 20211123].
[10] Vincenti WG (1990) What engineers know and how they know it: Analytical studies from aeronautical history. Johns Hopkins Studies in the History of Technlogy (Johns Hopkins University Press, Baltimore and London), .
[11] Schwertman NC, Gilks A, Cameron J (1990) A simple noncalculus proof that the median minimizes the sum of the absolute deviations. The American Statistician 44(1):38–39.
[12] Galton F (1907) Vox populi (the wisdom of crowds). Nature 75(7):450–451.
[13] Surowiecki J (2005) The wisdom of crowds (Anchor), .
[14] Spring J, Hatleback E, Householder A, Manion A, Shick D (2021) Time to change the cvss? IEEE Security & Privacy 19(2):74–78.
[15] Spring J, Hatleback E, Manion A, Shic D (2018) Towards improving cvss. SEI, CMU, Tech Rep .
[16] Howland (2021) A case against cvss: Vulnerability management done wrong, https://hlchowland.medium.com/a-case-against-cvss-vulnerability-management-done-wrong-99a0f8b740a3 [accessed 20220209].
[17] Munaiah N, Meneely A (2016) Vulnerability severity scoring and bounties: Why the disconnect? Proceedings of the 2nd International Workshop on Software Analytics,

pp 8–14.

[18] Allodi L, Massacci F (2013) How cvss is dossing your patching policy (and wasting your money). blackhat USA 2013, 27 July-1 August 2013, Las Vegas, USA, .

[19] Allodi L, Massacci F (2014) Comparing vulnerability severity and exploits using case-control studies. ACM Transactions on Information and System Security (TISSEC) 17(1):1–20.

[20] Feutrill A, Ranathunga D, Yarom Y, Roughan M (2018) The effect of common vulnerability scoring system metrics on vulnerability exploit delay. 2018 Sixth International Symposium on Computing and Networking (CANDAR) (IEEE), pp 1–10.

[21] Abramson (2018) A review of the common vulnerability scoring system, https://medium.com/critical-stack/a-review-of-the-common-vulnerability-scoring-system-2c7d266eda28 [accessed 20220209].

Appendix A—Acronyms

Selected acronyms and abbreviations used in this paper are defined below.

AI	Artificial Intelligence
CERT/CC	Computer Emergency Response Team Coordination Center
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DAG	Directed Acyclic Graph
FIRST	Forum for Incident Response and Security Teams
GUI	Graphical User Interface
NIST	National Institute of Standards and Technology
IR	Interagency or Internal Report
NVD	National Vulnerability Database
SIG	Special Interest Group
US	United States

CVSS base score vector string metrics and associated metric values:
(e.g., AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H):

AV (Attack Vector)	(N: Network, A: Adjacent, L: Local, P: Physical)
AC (Attack Complexity)	(L: Low, H: High)
PR (Privileges Required)	(N: None, L: Low, H: High)
UI (User Interaction)	(N: None, R: Required)
S (Scope)	(U: Unchanged, C: Changed)
C (Confidentiality)	(H: High, L: Low, N: None)
I (Integrity)	(H: High, L: Low, N: None)
A (Availability)	(H: High, L: Low, N: None)

Appendix B- Set of Evaluated CVSS vectors

On January 8 of 2021, NVD contained 73446 CVEs scored with CVSS version 3.1. The 66 most frequent CVSS vectors for these CVEs covers 90% of them. These top 66 CVSS vectors are listed in Tables 9 and 10 using the 'CVSS Vector String' format [1] along with their respective frequency counts. Appendix A contains expansions for the vector string acronyms.

Table 9. Top 66 Most Frequent CVSS Vectors per Mappings from NVD (higher frequency vectors)

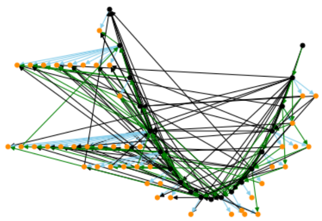
CVSS Vector	CVE Frequency
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	9979
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N	5572
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H	4434
CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	4378
CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H	3978
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H	3834
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N	3228
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H	2847
CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:N	2501
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N	1626
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	1375
CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H	1371
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:H	1243
CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N	1119
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N	1000
CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H	966
CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	895
CVSS:3.1/AV:N/AC:L/PR:H/UI:R/S:C/C:L/I:L/A:N	877
CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	770
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N	763
CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:H/A:H	748
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N	700
CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N	606
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:H/A:N	567
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H	553
CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N	549
CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H	497
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	440
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N	432
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N	407
CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:H	370
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:L/A:N	358
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:L/A:N	335

Table 10. Top 66 Most Frequent CVSS Vectors per Mappings from NVD (lower frequency vectors)

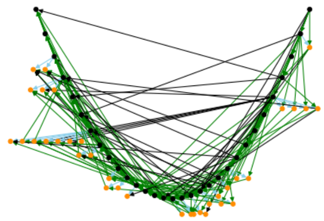
CVSS Vector	CVE Frequency
CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H	334
CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:N/I:N/A:H	307
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	295
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L	290
CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N	288
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N	286
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N	285
CVSS:3.1/AV:L/AC:H/PR:N/UI:R/S:U/C:H/I:H/A:H	268
CVSS:3.1/AV:P/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H	251
CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N	249
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:H	228
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N	215
CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:N	214
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L	205
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H	194
CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H	188
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:N	184
CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N	179
CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:H	163
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	162
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N	156
CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:N/A:H	151
CVSS:3.1/AV:L/AC:L/PR:H/UI:N/S:U/C:H/I:N/A:N	147
CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:N	143
CVSS:3.1/AV:L/AC:H/PR:L/UI:N/S:U/C:H/I:N/A:N	140
CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:H/I:H/A:H	138
CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:N/I:H/A:N	132
CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N	128
CVSS:3.1/AV:A/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H	125
CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H	124
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:H	118
CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N	112
CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N	110

Appendix C- Encoded Knowledge Constraint Graphs

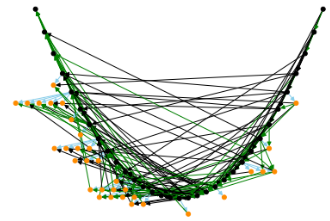
This appendix provides the graphs produced by the 12 CVSS SIG experts using the NIST Knowledge Encoding Tool. Figure 11 provides the raw graphs created by the tool. Figure 12 provides the simplified graphs where the redundant edges have been removed. Additionally, all edges have been updated to originate from and terminate to the representative nodes (the ones with the greatest frequency) for each equivalency set. This does not change the logic represented by the graph.



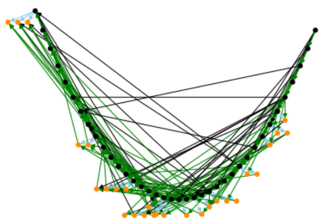
(a) 02c



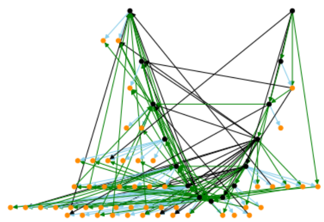
(b) 3d6



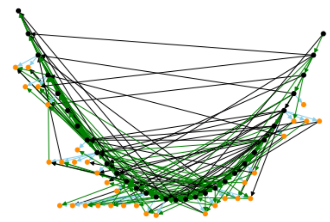
(c) 5fd



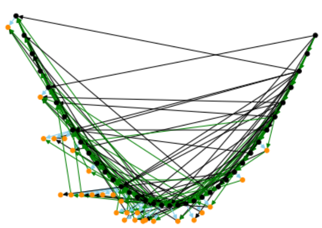
(d) 6e5



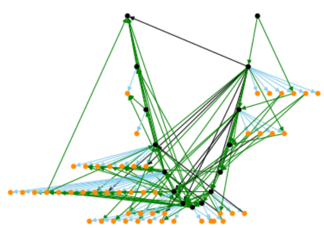
(e) 70a



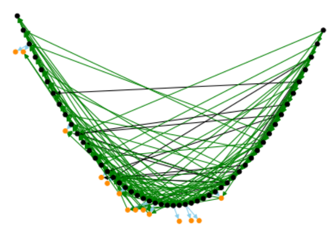
(f) 88d



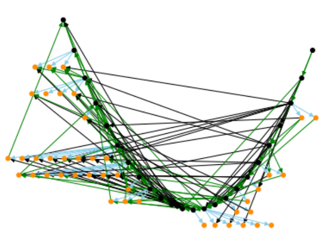
(g) 908



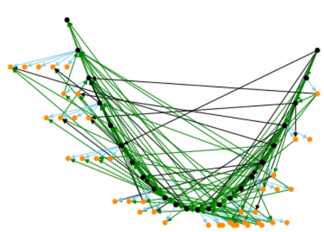
(h) 977



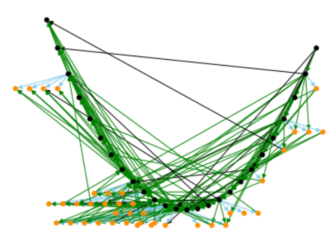
(i) 98a



(j) d3d

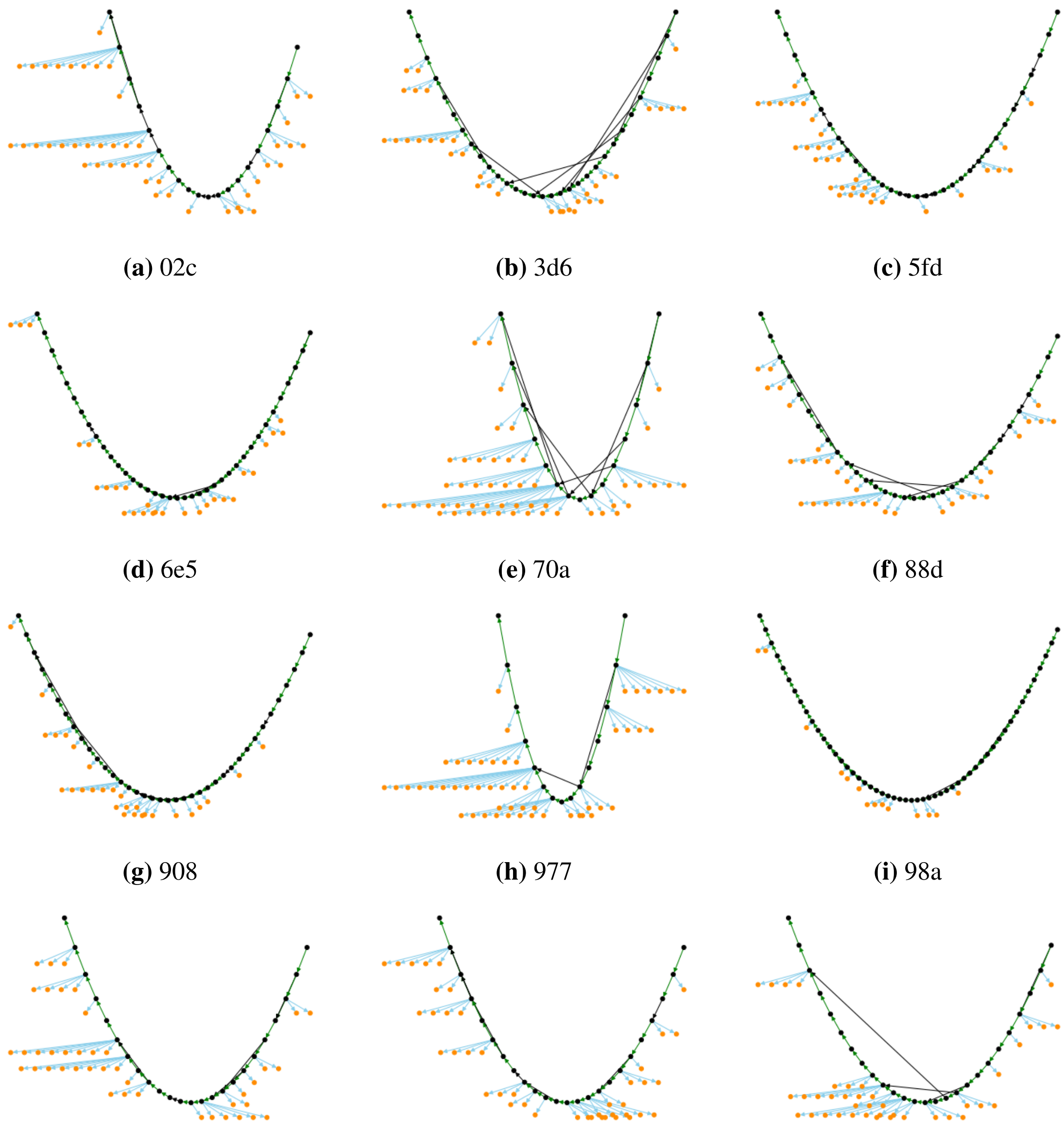


(k) f00



(l) f59

Fig. 11. Raw Graphs Produced by the Knowledge Encoding Tool for the 12 CVSS SIG Experts



(a) 02c **(b)** 3d6 **(c)** 5fd

(d) 6e5 **(e)** 70a **(f)** 88d

(g) 908 **(h)** 977 **(i)** 98a

(j) d3d **(k)** f00 **(l)** f59

Fig. 12. Simplified Graphs with Redundant Edges Removed