Metrics for Information Security Vulnerabilities

Andy Ju An Wang, Min Xia, and Fengwei Zhang
Southern Polytechnic State University
1100 S Marietta Parkway
Marietta, GA 30060, USA
01-678-915-3718
jwang@spsu.edu

ABSTRACT

It is widely recognized that metrics are important to information security because metrics can be an effective tool for information security professionals to measure, control, and improve their security mechanisms. However, the term "security metrics" is often ambiguous and confusing in many contexts of discussion. Common security metrics are often qualitative, subjective, without a formal model, or too naïve to be applied in real world. This paper introduces the criteria for good security metrics and how to establish quantitative and objective information security metrics with the recently released CVSS 2.0 (Common Vulnerability Scoring System), which provides a tool to quantify the severity and risk of a vulnerability to an information asset in a computing environment. We analyze some issues in CVSS 2.0 and propose our solutions. The discussion helps establish insights on security metrics and their applications in security automation and standardization.

Keywords

Information Security, Threats and Vulnerabilities, Metrics and Measurement, Common Vulnerability Scoring System

1. Introduction

It is widely recognized that metrics are important to information security because metrics can be an effective tool for information security professionals to measure the security strength and levels of their systems, products, processes, and readiness to address security issues they are facing. Metrics can also help identify system vulnerabilities, providing guidance in prioritizing corrective actions, and raising the level of security awareness within the organization. With the knowledge of security metrics, an information security professional can answer typical questions like "Are we secure?" and "How secure are we?" in a formal and persuadable manner. For federal agencies, a number of existing laws, rules, and regulations cite security metrics as a requirement. These laws include the Clinger-Cohen Act, Government Performance and Results Act (GPRA), Government Paperwork Elimination Act (GPEA), and Federal Information Security Management Act (FISMA). Moreover, metrics can be used to justify and direct future security investment. Security metrics can also improve accountability to stakeholders and improve customer confidence.

However, the term "security metrics" is often ambiguous and confusing in many contexts of discussion in information security. Some guiding standards and good experiments of security metrics exist, such as FIPS 140-1/2 [NIST 01], ITSEC [CEC 91], TCSEC [DOD 85], Common Criteria (CC) [CC1 99][CC2 99][CC3 99] and NIST Special Publication 800-55 [NIST 03], but they are either too broad without precise definitions, or too narrow to be generalized to cover a great variety of security situations. *First, security metrics are often qualitative rather than quantitative*. While TCSEC [DOD 85] provides seven levels of trust measurement called *ratings*, which are represented by six evaluation classes C1, C2, B1, B2, B3, and A1, plus an additional class D, ITSEC [CEC 91] provides six levels of trust, called *evaluation levels*, E1, E2, E3, E4, E5, and E6. The Common Criteria (CC) [CC1 99][CC2 99][CC3 99] delivers a measure of the evaluation result called a *level of trust* that indicates how trustworthy the product or system is with respect to the security functional requirements defined for it. This evaluation provides an independent assessment by experts and a measure of assurance, which can be used to compare products. Nevertheless, the *level of trust* of various methodologies is a qualitative indicator by nature. There are no mathematical formulas to be applied to obtain the level of trust as a value of such an indicator. The evaluation process is highly

qualitative as well because all the evaluation evidence, evaluator's qualification and experience, and evaluation methods are often difficult to quantify. *Second, security metrics are often subjective rather than objective.* The Delphi technique, for instance, measure a system security risk by collecting and comparing subjective opinions of individual members of a working group. Each member of the group writes down his or her opinion of a certain security risk on a piece of paper and turns it into the team that is performing the security analysis. The results are compiled and distributed to the group members who then write down their comments anonymously and return them back to the analysis group. The comments are compiled and redistributed for more comments until a consensus is formed. This method obtains a security metric mainly based on individual's subjective opinions or experience, and reaches a final conclusion by consensus.

Metrics are quantifiable measurement. Security metrics are quantitative indicators for the security attributes of an information system or technology. A quantitative measurement is the assignment of numbers to the attributes of objects or processes. For information security professionals, we are interested in measuring the fundamental security attributes of information such as confidentiality, integrity, and availability.

What is a Good Metric? First of all, *a good metric must yield quantifiable measurement*. Security metrics will measure information attributes such its size, format, confidentiality, integrity, and availability. Therefore, metrics define and reflect these attributes by numbers such as percentages, averages, or weighted sums. According to [Swanson 2003], information security metrics must be based on security performance goals and objectives. Security performance goals state the desired results of a security implementation. Security performance objectives enable accomplishment of goals by defining security policies and procedures and direct consistent implementation of security controls across the organization.

In the following, we list a few criteria for a good metric informally:

- ***Objectiveness***: Measurement must yield quantifiable information such as percentages, averages, and weighted sums. The measure should not be influenced by the measurer's beliefs or tasting. We may say that in a speed skating competition, the electronic timing system that automatically records skaters finish times provides objective measurement, while human judges in a figure skating competition may not be objective.

- ***Repeatability***: If repeated in the same context, with exactly the same conditions, the measure should return the same result. A truly scientific experiment is always repeatable. A non-repeatable measurement creates uncertain result and is thus not useful.

- ***Clarity***: A measure should be easy to interpret with a clearly defined semantics. For instance, while it can be clearly specified as how to measure the height and weight of a person, it is arguable to define a clear process and method to measure a person's attractiveness and intelligence level.

- ***Easiness***: The measurement of an attribute should be easy to perform. Sometimes one attribute may contain many different parameters. In order to make the measurement useful and effective, we may target a succinct measurement considering only the most important parameters. The measure should create or add knowledge about the entity itself, sometimes with the purpose of improving the usefulness of the entity.

The rest of the paper is organized as follows: Section 2 introduces CVSS, a measurement of vulnerabilities. Section 3 discusses the CVSS base score distribution. Section 4 presents several issues identified from CVSS base score calculation. Section 5 proposals a few solutions for the issues identified previously. The last section lists some further research issues.

2. CVSS: Common Vulnerability Scoring System

The CVSS (Common Vulnerability Scoring System) provides a tool to quantify the severity and risk of a vulnerability to an information asset in a computing environment. It was designed by NIST (National

Institute of Standard and Technology) and a team of industry partners. CVSS metrics for vulnerabilities are divided into three groups: *Base metrics* measure the intrinsic and fundamental characteristics of vulnerabilities that do not change over time or in different environments. *Temporal metrics* measure those attributes of vulnerabilities that change over time but do not change among user environments. *Environmental metrics* measure those vulnerability characteristics that are relevant and unique to a particular user's environment.

There are six base metrics that capture the most fundamental features of a vulnerability:

(1) **Access Vector (AV):** It measures how the vulnerability is exploited, for instance, locally or remotely. The more remote an attacker can be to attack an information asset, the greater the vulnerability score.
(2) **Access Complexity (AC)**: It measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system. The lower the required complexity, the higher the vulnerability score.
(3) **Authentication (Au)**: It measures the number of times an attacker must authenticate to a target in order to exploit a vulnerability. The fewer authentication instances that are required, the higher the vulnerability score.
(4) **Confidentiality Impact (CC)**: It measures the impact on confidentiality of a successfully exploited vulnerability. Increased confidentiality impact increases the vulnerability score.
(5) **Integrity Impact (IC)**: It measures the impact on integrity of a successfully exploited vulnerability. Increased integrity impact increases the vulnerability score.
(6) **Availability Impact (AC)**: It measures the impact on availability of a successfully exploited vulnerability. Increased availability impact increases the vulnerability score.

The temporal metrics in CVSS represent the time dependent features of the vulnerabilities, including exploitability in terms of their technical details, the remediation status of the vulnerability, and the availability of exploit code or techniques. The environmental metrics represent the implementation and environment specific features of the vulnerability. There are three environmental metrics as defined below, which capture the characteristics of a vulnerability that are associated with a user's IT environment.

The scoring process first calculates the base metrics according to the base equation, which delivers a score ranging from 0 to 10, and creates a vector. The vector is a text string that contains the values assigned to each metric, and it is used to communicate exactly how the score for each vulnerability is derived.

Optionally, the base score can be refined by assigning values to the temporal and environmental metrics. If the temporal score is needed, the temporal equation will combine the temporal metrics with the base score to produce a temporal score ranging from 0 to 10.

Similarly, if an environmental score is needed, the environmental equation will combine the environmental metrics with the temporal score to produce an environmental score ranging from 0 to 10. For the purpose of this paper, we give below the base metric equations only. Table 1 summarizes the parameter values for the base score system.

Base Equation:

BaseScore = round_to_1_decimal (((0.6 * Impact) + (0.4 * Exploitability) – 1.5) * f(Impact))
Impact = 10.41 * (1 - (1 - ConfImpact) * (1 – IntegImpact) * (1 – AvailImpact))
Exploitability = 20 * AccessVector * AccessComplexity * Authentication
f(Impact) = 0 if Impact = 0, 1.176 otherwise.

Metric Name	Value 1	Value 2	Value 3
Access Vector(AV)	Local(L) 0.395	Adjacent network(A) 0.646	Network(N) 1.0
Access Complexity(AC)	High(H) 0.35	Medium(M) 0.61	Low(L) 0.71
Authentication(Au)	Multiple(M) 0.45	Single(S) 0.56	None(N) 0.704
Confidential Impact(CI)	None(N)	Partial(P)	Complete(C)

	0.000	0.275	0.660
Integrity Impact(II)	None(N) 0.000	Partial(P) 0.275	Complete(C) 0.660
Availability Impact(AI)	None(N) 0.000	Partial(P) 0.275	Complete(C) 0.660

Table 1 Parameter Values for Base Scores

3. Analysis of CVSS Base Scores

3.1 Exploitability Analysis

According to the Base Equation, the Exploitability is a function of AV, AC, and Au, as shown below:

Exploitability = F (AV, AC, Au) = 20*AccessVector*AcessComplexity*Authentication

As an increased function with its parameters, Exploitability has 27 (= 3*3*3) possible values ranging from 0 to 10, and each value has an equal probability, 1/27, to be associated with a vulnerability. The minimum value of Exploitability is 1.24425, while the maximum value is 9.996799999999999.

(AC AV Au) values	Occurance	Exploitability Values
Local High Multiple	1	1.24425
Local High Single	1	1.5484000000000002
Local High None	1	1.94656
Adjacent network High Multiple	1	2.0349
Local Medium Multiple	1	2.16855
Local Low Multiple	1	2.52405
Adjacent network High Single	1	2.53232
Local Medium Single	1	2.69864
Local Low Single	1	3.1410400000000003
Network High Multiple	1	3.15
Adjacent network High None	1	3.183487999999999
Local Medium None	1	3.3925759999999996
Adjacent Medium Multiple	1	3.54654
Network High Single	1	3.9200000000000004
Local Low None	1	3.948736
Adjacent network Low Multiple	1	4.12794
Adjacent Medium Single	1	4.4134720000000005
Network High None	1	4.928
Adjacent Low Single	1	5.136992
Network High None	1	5.49
Adjacent Medium None	1	5.5483648
Network Low Multiple	1	6.39
Adjacent Low None	1	6.457932799999999
Network Medium Single	1	6.832
Network Low Single	1	7.952
Network Medium None	1	8.588799999999999
Network Low None	1	9.996799999999999

Table 2 Value Distribution of Exploitability

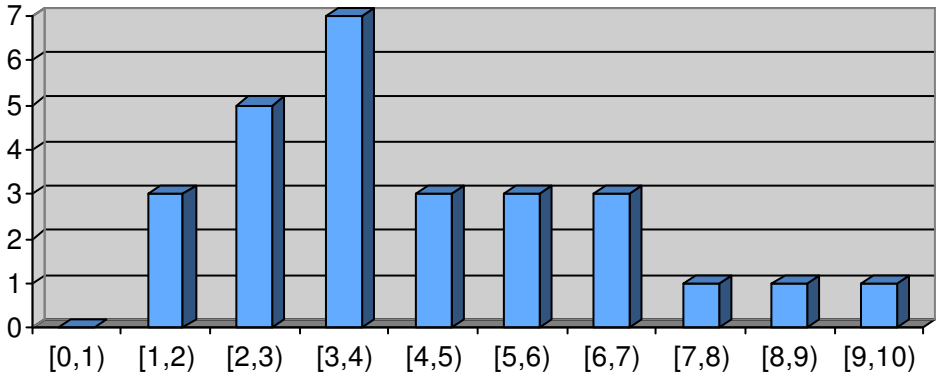

Figure 1 Distribution of Exploitability Values

3.2 Impact Analysis

For formula for Impact Analysis is given below:

Impact = G (CI, II, AI) = 10.41*(1-(1-ConfImpact)*(1-IntegImpact)*(1-AvailImpact))

This function is obvious an increasing function with respect to its three parameters: ConfImpact, IntegImpact, and AvailImpact. There are 10 kinds of result of Impact. The minimum value is 0, and the maximum value is 10.0008536. The rang is about from 0 to 10.

CI, II, AI	Occurance	Impact Values
NNN	1	0
PNN, NPN, NNP	3	2.86375
PPN, PNP, NPP	3	4.93824375
NNN	1	6.442976718750001
CNN, NCN, NNC	3	6.8706000000000005
CPN, CNP, PCN PNC, NCP, NPC	6	7.843935000000001
CPP, PCP, PPC	3	8.549602875000001
CCN, CNC, NCC	3	9.206604
CCP, CPC, PCC	3	9.5375379
CCC	1	10.00084536

Table 3 Impact Calculation

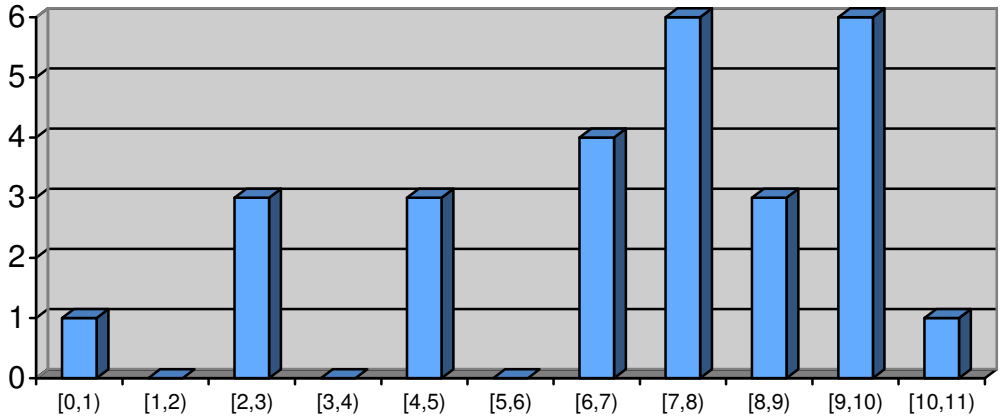

Figure 2 Distribution of Impact Values

3.3 Base Score Distribution Analysis

BaseScore = T (Exp, Imp) = (((0.6*Impact)+(0.4*Exploitability)-1.5)*f(Impact))
f(Impact) = 0 if Impact = 0, 1.176 otherwise.

This function is an increasing function with respect to its two parameters: Impact and Exploitability. There are totally 279 results for the BaseScore with 244 (= 9*27 + 1) kinds of BaseScores. The minimum of the BaseScore is 0, and the maximum is 9.995091206016. The range is about from 0 to 10.

Interval	[0,1)	[1,2)	[2,3)	[3,4)	[4,5)	[5,6)	[6,7)	[7,8)	[8,9)	[9,10)
Numbers	33	33	48	68	127	193	122	69	28	8
Total										279

Table 4 Base Scores

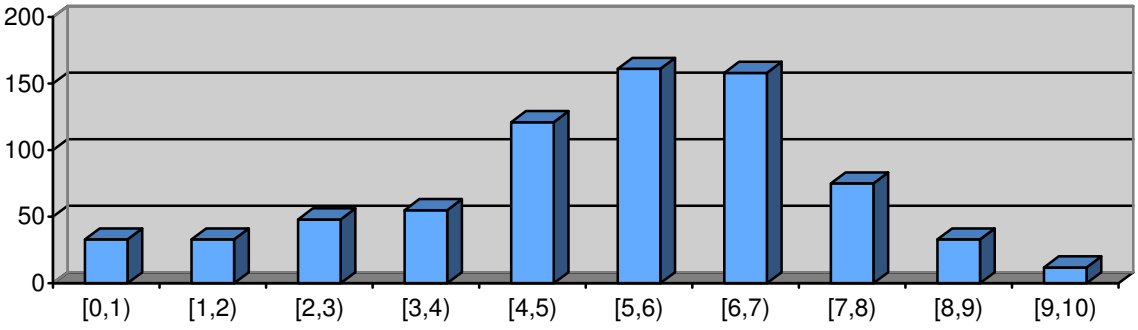

Figure 3 Base Score Distribution

4. The CIA Impacts on the Base Scores

Suppose the Exploitability has a value of 10.0 (AV:Network;AC:High;Au:None), we change the metric of CIA Impact values step by step as shown in the following table. We would like to see how the impact values affect the final base scores.

Confidentiality Impact	Integrity Impact	Availability Impact	Exploitability Subscore	Impact Subscore	Base Score
Complete	Complete	Complete	10.0	10.0	10.0
Complete	Complete	Partial	10.0	9.5	9.7
Complete	Partial	Complete	10.0	9.5	9.7
Partial	Complete	Complete	10.0	9.5	9.7
Complete	Complete	None	10.0	9.2	9.4
Complete	None	Complete	10.0	9.2	9.4
None	Complete	Complete	10.0	9.2	9.4
Complete	Partial	Partial	10.0	8.5	9.0
Partial	Complete	Partial	10.0	8.5	9.0
Partial	Partial	Complete	10.0	8.5	9.0
Partial	Partial	Partial	10.0	6.4	7.5
Complete	Partial	None	10.0	7.8	8.5
Complete	None	Partial	10.0	7.8	8.5
Partial	Complete	None	10.0	7.8	8.5
None	Complete	Partial	10.0	7.8	8.5
Partial	None	Complete	10.0	7.8	8.5
None	Partial	Complete	10.0	7.8	8.5
Complete	None	None	10.0	6.9	7.8
None	Complete	None	10.0	6.9	7.8
None	None	Complete	10.0	6.9	7.8
Partial	Partial	None	10.0	4.9	6.4
Partial	None	Partial	10.0	4.9	6.4
None	Partial	Partial	10.0	4.9	6.4
Partial	None	None	10.0	2.9	5.0
None	Partial	None	10.0	2.9	5.0
None	None	Partial	10.0	2.9	5.0
None	None	None	10.0	0	0

Table 5 Relationship between Base Score and Impact Values

Table 5 shows that we change the metric of CIA Impact step by step to make the BaseScore form 10 to 0. Let us consider the rows with the same color as one step. It turns out that the BaseScore drops slowly in the

first a few steps – every time when the impact values change, the BaseScore drops less than 0.5, but we in the last a few steps, the BaseScore drops sharply with a change of BaseScore greater than 1.3. How does this happen? It seems unreasonable to have a slower change when BaseScore is greater, while a faster change when BaseScore is smaller.

This problem is caused by the impact formula: *Impact = 10.41*(1-(1-ConfImpact)*(1-IntegImpact)*(1-AvailImpact)).* Since *ConfImpact, IntegImpact,* and *AvailImpact* are all decimal numbers less than 1, the multiplication of them produces smaller result if two of them have smaller values. When the impact is small, the vulnerability score should be small accordingly with the same rate.

We extract some rows form Table 5 to indicate the unreasonable places in CVSS BaseScore calculation.

Confidentiality Impact	**Integrity Impact**	**Availability Impact**	**Exploitability Subscore**	**Impact Subscore**	**BaseScore**
Complete	Partial	Partial	10.0	8.5	9.0
Complete	Partial	None	10.0	7.8	8.5
The value of change			0	**0.7**	**0.5**

Confidentiality Impact	**Integrity Impact**	**Availability Impact**	**Exploitability Subscore**	**Impact Subscore**	**BaseScore**
Partial	None	Partial	10.0	4.9	6.4
Partial	None	None	10.0	2.9	5.0
The value of change			0	**2.0**	**1.4**

Table 6 Extract from Table 5

In both extractions in Table 5, we keep the metrics of Confidentiality Impact and Integrity Impact unchanged, and the metric of Availability Impact from Partial to None. It is unreasonable that the BaseScore values changed significantly for the same change of availability value from Partial to None. The second case in the lower table has a value change that is almost three times of that in the first case. The change of Impact subscore is 0.7, while the change of BaseScore is 0.5 in the first case. The change of Impact subscore is 2.0, and the change of BaseScore is 1.4 in the second case.

Let us see another pair of data indicating the similar problem:

Confidentiality Impact	Integrity Impact	Availability Impact	Exploitability Subscore	Impact Subscore	BaseScore
Complete	Complete	Complete	10.0	10.0	10.0
Complete	Complete	Partial	10.0	9.5	9.7
The value of change			0	**0.5**	**0.3**

Confidentiality Impact	Integrity Impact	Availability Impact	Exploitability Subscore	Impact Subscore	BaseScore
None	None	Complete	10.0	6.9	7.8

None	None	Partial	10.0	2.9	5
The value of change			0	**4.0**	**2.8**

Table 7 The Second Extraction from Table 5

Keeping Confidentiality Impact and Integrity Impact value fixed, when we change Availability impact from Complete to Partial as shown in Table 7, the BaseScore changes are significant and looks unreasonable.

5. The Unreachable Values in (0, 5) for Base Scores

Checking the last two rows of Table 5, you will find the BaseScore values drops from 5.0 to 0 directly. This means that when AV:Network; AC:High; Au:None, the BaseScore can not reach any values between 0 and 5.0. This is unfair to some vendors because some products will a minimum CVSS score of 5.0 even their products are perfectly secure. The characteristics of these kinds of products include those with high Exploitability Subscores.

In the National Vulnerability Database (NVD), there are 34 vulnerabilities for AOL Instant Messenger with the following statistics:

Number of Vulnerabilites	Impact Subscore	Exploitability Subscore	BaseScore
1	6.4	8.6	6.8
2	6.4	8.6	6.8
3	2.9	10	5.0
4	6.4	10	7.5
5	6.9	10	7.8
6	6.9	10	7.8
7	2.9	8.6	4.3
8	2.9	8.6	4.3
9	6.4	4.9	5.1
10	2.9	10	5.0
11	2.9	10	5.0
12	2.9	10	5.0
13	6.4	10	7.5
14	10	10	10
15	2.9	10	5.0
16	2.9	10	5.0
17	2.9	4.9	2.6
18	2.9	10	5.0
19	6.4	10	7.5
20	2.9	10	5.0
21	6.4	10	7.5
22	6.4	10	7.5
23	10	10	10
24	2.9	10	5.0
25	2.9	10	5.0
26	2.9	10	5.0
27	2.9	10	5.0
28	6.4	4.9	5.1
29	6.4	10	7.5
30	6.4	10	7.5
31	2.9	10	5.0

32	2.9	10	5.0
33	2.9	10	5.0
34	2.9	10	5.0

Table 8 CVSS Scores for AIM

Most Exploitability Subscores for AOL Instant Messenger have a value of 10 because they have the following values by their product nature: AV: Network;AC:High:Au:None. Therefore, their CVSS BaseScore is always equal to or greater than 5.0. This is unfair to them as their product characteristics inherently have the high Exploitability.

With the similar approach, we found that the BaseScores could not reach the intervals (0, 0.84) and (8.4, 10) under certain exploitability and impact values.

6. Re-defining CIA Impact on Base Scores

The BaseScore equation has a multiplication factor f(Impact) as shown below:

BaseScore = round_to_1_decimal (((0.6*Impact)+(0.4*Exploitability)-1.5)*f(Impact))
f(impact) = 0 if impact = 0, otherwise f(impact) = 1.176

If a vulnerability has no impact on Confidentiality, Integrity, or Availability, the BaseScore of the vulnerability will be zero, as f(impact) = 0 if impact = 0. However, the current version of CVSS treats those minor impact situations as the same as those with significant impacts indicated by the equation f(impact) = 1.176 when the impact sub-score is not zero.

As CIA (Confidentiality, Integrity, and Availability) impact plays an important role in CVSS calculation, the authors believe that we should define f(impact) as a multiple tiered function. The BaseScore should heavily dependent on the CIA impact. In the example of AOL Instant Messenger examples given above, we found that the Impact value is around 2.9, the Exploitability value is 10, and the vulnerability BaseScore is always greater than 5.0.

Impact Subscore	Exploitability Subscore	BaseScore
2.9	10	5.0

One way to resolve this issue is to re-define the CIA impact function f(impact) as a multi-tiered function as shown in Table 9 below.

Confidentiality Impact	Integrity Impact	Availability Impact	f(impact)
None	None	None	0
None	None	Partial	0.4
None	Partial	Partial	0.7
Complete	Complete	None	0.9
Other condition			1.176

Table 9 New Definition of f(impact)

7. Conclusion and Discussion

CVSS provides a simple tool to define information system vulnerabilities reflecting the overall severity and risk presented by those vulnerabilities. Security professionals, executives, and end-users have a common language now to discuss security threats, vulnerability severity, and risk analysis with CVSS. Other metric and scoring systems for vulnerability do exist. But they are either tending to be subjective or qualitative. CVSS differs by offering an open framework for comparing and ranking vulnerabilities in a consistent fashion. This paper identifies some issues in the current version of CVSS 2.0, but we believe that CVSS has great potential to become a foundation for the automation of security tools. As CVSS matures, its metrics will become more accurate, more flexible, and more representative for common security vulnerabilities and their risks.

For the limitation of space, this paper does not cover the issues of temporal metrics and environmental metrics, which are two integral parts of CVSS 2.0. The temporal metrics represent the time dependent features of the vulnerabilities, including exploitability in terms of their technical details, the remediation status of the vulnerability, and the availability of exploit code or techniques. The environmental metrics represent the implementation and environment specific features of the vulnerability. There are three environmental metrics defined in CVSS 2.0, which capture the characteristics of a vulnerability that are associated with a user's IT environment. The scoring process first calculates the base metrics according to the base equation, which delivers a score ranging from 0 to 10, and creates a vector. The vector is a text string that contains the values assigned to each metric, and it is used to communicate exactly how the score for each vulnerability is derived. Optionally, the base score can be refined by assigning values to the temporal and environmental metrics. If the temporal score is needed, the temporal equation will combine the temporal metrics with the base score to produce a temporal score ranging from 0 to 10. Similarly, if an environmental score is needed, the environmental equation will combine the environmental metrics with the temporal score to produce an environmental score ranging from 0 to 10.

Temporal metrics measures the severity of a vulnerability that may change over time. It is very interesting to see how temporal metrics could be extended to use time-based tools such as temporal logic or interval logic. Environmental metrics are directly related to a user's IT environment. It merits further study in how temporal and environmental metrics affect IT product improvement from vendors and how end users take advantage of the tools like CVSS in their IT practice and administration.

References

[Atzeni 2005] Andrea Atzeni and Antonio Lioy, Why to adopt a security metric? A brief survey, *POSITIP Report*, 2005.

[Swanson 2003] Marianne Swanson et.al., Security Metrics Guide for Information Technology Systems, *NIST Special Publication 800-55*, July 2003.

[Chew 2006] Elizabeth Chew et.al., Guide for Developing Performance Metrics for Information Security, *NIST Special Publication 800-80*, May 2006.

[Patriciu 2006] Victor-Valeriu Patriciu, Iustin Priescu, and Sebastian Nicolaescu, Security Metrics for Enterprise Information Systems, *Journal of Applied Quantitative Methods*, V.1, N.2, Winter 2006. pp. 151 – 159.

[Mell 2007] Peter Mell, Karen Scarfone, and Sasha Romanosky, A Complete Guide to the Common Vulnerability Scoring System (CVSS), Version 2.0, Forum of Incident Response and Security Teams, http://www.first.org/cvss/cvss-guide.html (July 2007).

[CERT 2007] CERT, Vulnerability Remediation Statistics, http://www.cert.org/stats/fullstats.html, (July 2007).

[US-CERT 2007] US-CERT, United States Computer Emergency Readiness Team, http://www.us-cert.gov/, (July 2007).

[BIS 03] Matt Bishop, "*Computer Security: Art and Science*", Addison Wesley, 2003. ISBN: 0-201-44099-7.

[CC1 99] National Institute of Standards and Technology, "*Common Criteria for Information Technology Security Evaluation, Part I: Introduction and General Model*", Version 2.1, CCIMB-99-031, August 1999.

[CC2 99] National Institute of Standards and Technology, "*Common Criteria for Information Technology Security Evaluation, Part II: Security Function Requirements*", Version 2.1, CCIMB-99-031, August 1999.

[CC3 99] National Institute of Standards and Technology, "*Common Criteria for Information Technology Security Evaluation, Part III: Security Assurance Requirements*", Version 2.1, CCIMB-99-031, August 1999.

[CEC 91] Commission of the European Communities, "*Information Technology Security Evaluation Criteria*", Version 1.2, 1991.

[DOD 85] Department of Defense, "*Trusted Computer System Evaluation Criteria*", DOD 5200.28-STD, December 1985.

[NIST 01] National Institute of Standards and Technology, "*Security Requirements for Cryptographic Modules*", PIPS PUB 140-2, May 2001.

[NIST 03] Marianne Swanson, Nadya Bartol, John Sabato, Joan Hash, and Laurie Graffo, "*Security Metrics Guide for Information Technology Systems*", NIST Special Publication 800-55, National Institute of Standards and Technology, http://csrc.nist.gov/publications/nistpubs/800-55/sp800-55.pdf. July 2003.

[POT 00] Ronald W. Potter, "*The Art of Measurement, Theory and Practice*", Printice Hall PTR, Upper Saddle River, New Jersey, 2000. ISBN 0-13-026174-2.

[Wang 02] J. A. Wang, "Algebra for Components", in *Proceedings of The 6th World Multiconference on Systemics, Cybernetics and Informatics*, V. 5, Computer Science I, eds. Nagib Callaos, Tau Leng, and Belkis Sanchez. ISBN: 980-07-8150-1, July 2002, pp. 213 - 218.

[Wang 04] J.A.Wang, Security Testing in Software Engineering Courses, *Proceedings of Frontiers in Education Conference*, Session F1C, IEEE Catalog Number 04CH37579C, ISBN: 0-7803-8553-5. October 2004, Savannah, Georgia.

[Wang 05] J. A. Wang, "Information Security Models and Metrics", in *Proceedings of 43rd ACM Southeast Conference*, Volume 2, pp. 178 – 184. ISBN: 1-59593-059-0. March 2005, Kennesaw, GA.