IVSE: AN IMPROVED CVSS BASE SCORE MECHANISM WITH ENVIRONMENT REPRESENTATIVE

[1]Gagandeep Chawla, [2]Dr. Neeraj Sharma, [3] Dr. Narender Kumar Rawal
[1]Research Scholar, [2]Dean & Professor [3]Assistant Professor
[1]Computer Science,
[1]I.K. Gujral Punjab Technical University, Kapurthala, India.

Abstract: Over the past several years, number of IT vendors and computer security organizations are trying to improve the vulnerability scoring systems. Due to the parallel growth of vulnerabilities along with the increase in software's and applications, the improvement in scoring system is a never ending process. The main cause for occurrence of vulnerabilities is the complex nature of programming and human negligence while designing and coding. Vulnerabilities can be found in software, operating systems, routers, mobile applications and other hardware devices. CVSS (Common Vulnerability Scoring System) is a widely accepted framework, playing a vital role in rating and rolling back these vulnerabilities. CVSS uses three equations (two optional) to ascertain three severity rating scores. These scores are numerical in nature and ranges from 0.0 to 10.0. Where score value 10.0 is the most severe. This paper analyses the effect of introducing "Environment Representative" in CVSS-V2 base score equation. This is achieved after deducing metric classification, values and weights for the 'Environment Representative' factor.

Index Terms: CVSS, Environment representative, Vulnerability score, Operating Environment.

I. INTRODUCTION

Massive use of technology and automated equipment almost in every field is inviting IT companies to develop software and applications to operate them. To accomplish these tasks on time, sometimes it leads to the poor quality software development and hence becomes the root cause of growing vulnerabilities and security issues. These unattended flaws or vulnerabilities in software can lead to exploitation in the form of unauthorized access. The detection of such flaws and vulnerabilities is crucial and need to be resolved promptly [1][10]. Moving ahead in this concern, software and security organizations have developed many proprietary schemes to sanitize software from vulnerabilities. The common issue with such kind of schemes is that they are more Internet-centric; and the quantification of evaluation factors that are subjective in nature becomes difficult [2][11]. A computer system or equipment could have different types of operating environments and different resistant powers against vulnerabilities. Vulnerabilities have different potential of exploitation on different operating environments. So while designing vulnerability solutions, environment factor of a device should not be neglected [13]. To overcome this issue, a new improved scoring method IVSE (Improved Vulnerability Scoring with Environment representative) is proposed by introducing a factor i.e. "Environment Representative" into CVSS-V2 base score equation. In IVSE, we propose environment factor as an important factor to be included in CVSS base score. This environment factor refers to the operating system used on effected system. Considering these factors, metric references are adjusted to 0.5/0.4/0.7 for Linux/Windows/Mac and other operating systems respectively. This score reflects the user environment. The results show that after implementing "Environment Representative" in base score equation, IVSE gives a better clarity on severity of vulnerabilities.

II. CVSS (COMMON VULNERABILITY SCORING SYSTEM)

CVSS is an open industry standard for measuring the severity of vulnerabilities which is developed and maintained by SIG (Special Interest Group) [4]. It is available for free and is an open industry standard for measuring the severity of vulnerabilities [7][12]. It does scoring of IT vulnerabilities and communicates their characteristics and impact on security. This is currently maintained by a global forum called FIRST

(Forum of Incident Response and Security Teams). For the calculation of the vulnerability score, CVSS uses three metric groups namely Base metric, Temporal metric and Environmental metric as shown in Fig.1. Out of these three metrics, normally the vulnerability score is given using base metric, Temporal and Environmental are optional [5][8]. Therefore in normal scoring these two are not always included. Each metric generates a numeric score in the range 0 to 10. Figure-1 shows the metric categories of CVSS-V2.

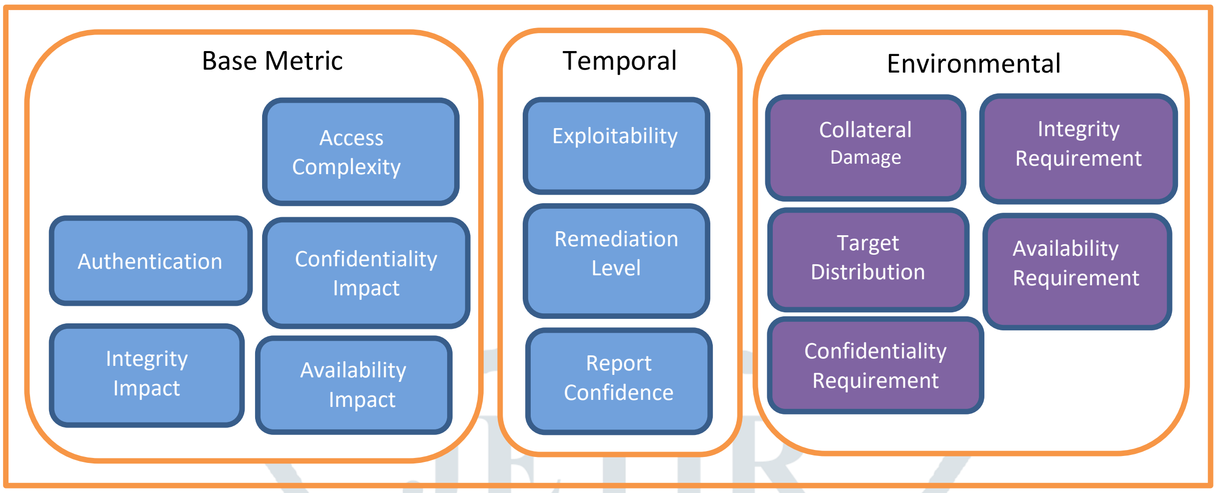


Fig. 1: Metric categories of CVSS Version 2

The first base group of CVSS represents the basic and fundamental qualities of vulnerability that do not change over time. The base group further comprises of AV (Access Vector), AC (Access Complexity), Au (Authentication), CI (Confidentiality Impact), II (Integrity Impact) and AI (Availability Impact). Of these the first three i.e. AV, AC and Au respectively represent how the vulnerability is accessed, its complexity and the type of authentication required to exploit it.

The features of vulnerabilities that changes over time are mainly comprised in the temporal group. The features of vulnerabilities that are relevant and unique to the specific user environment are comprised in the environmental group.

Of the three scores obtained base score is compulsory, and both temporal and environmental scores are not compulsory for inclusion in final vulnerability score calculation.

Fig. 2 shows the input of the three scores in the final score. It dotted box shows that both temporal and environmental scores are optional for calculation of the final score. .

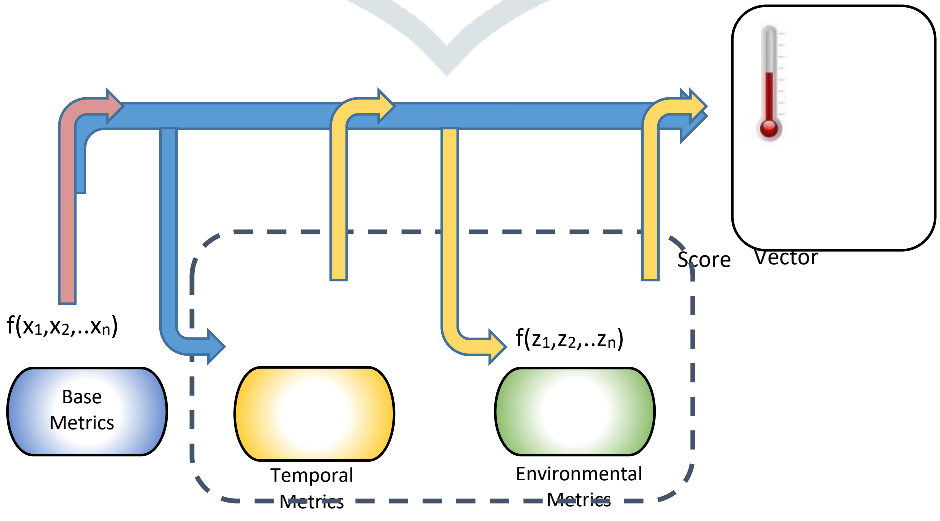


Fig. 2 Final vulnerability score calculation flow in CVSS

In CVSS the score equations for calculation of base, temporal and environmental scores is as below.

i) ***Base score equation:***

The equation 1 below gives the formula for calculation of base score in CVSS, where the method for calculation of the sub components i.e. *Impact, Exploitability and f(impact)* is shown in equation 1.1, equation 1.2 and equation 1.3 respectively.

$$\textit{Base Score} = \textit{round to 1 decimal} (((0.6*\textit{Impact}) + (0.4*\textit{Exploitability}) - 1.5)*f(\textit{Impact}))\ldots(1.0)$$
$$\textit{Impact} = 10.41*(1-(1-CI)*(1-II)*(1-AI))\ldots\ldots(1.1)$$
$$\textit{Exploitability} = 20*AV*AC*Au\ldots\ldots(1.2)$$
$$f(\textit{impact}) = 0 \textit{ if Impact}=0, 1.176 \textit{ otherwise}\ldots\ldots(1.3)$$

In the equations above, CI refers to Confidentiality Impact, II refers to Integrity Impact, AI refers to Availability Impact, AV refers to Access Vector, AC refers to Access Complexity and Au refers to Authentication.

ii) Temporal score equation

$$\text{Temporal Score} = \text{round to 1 decimal (Base Score* Exploitability} \; * \text{RL} * \text{RC})\ldots\ldots (2.0)$$

Equation 2 shows the formula for calculating temporal score in CVSS. In this, RL refers to Remediation Level and RC refers to Report Confidence.

iii) The Environmental Score Equation

$$\text{Environmental Score} = \text{round to 1 decimal ((AT + (10-AT)* CDP) *TD)}\ldots\ldots (3.0)$$

Equation 3 shows the formula for calculating environment score in CVSS. In this AT refers to Adjusted Temporal where it is the Temporal Score recomputed with the Base Score's Impact sub equation replaced with the Adjusted Impact equation, CDP refers to Collateral Damage Potential and TD refers to the Target Distribution.

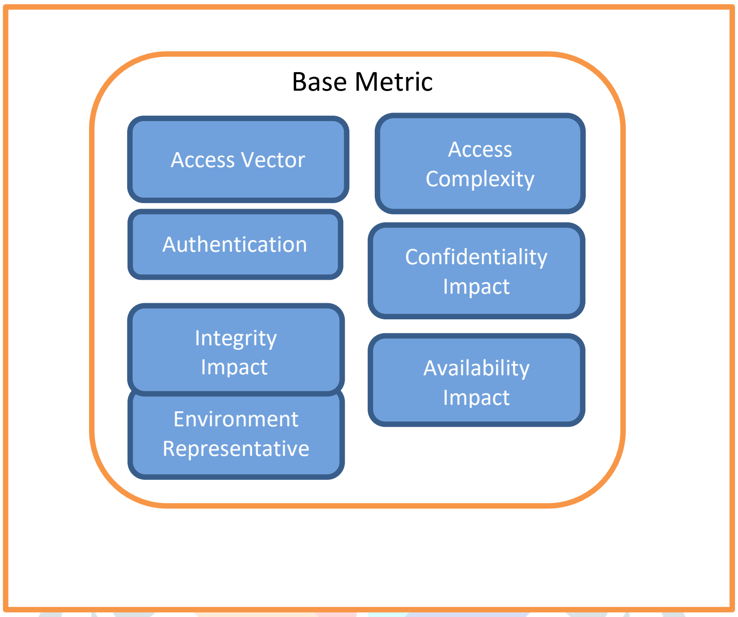
III. PROPOSED IVSE: A NEW IMPROVED VULNERABILITY BASE SCORING MECHANISM BASED ON CVSS

After analyzing the above scoring mechanism, it has been seen that environment representative should be considered in the calculation of base score itself; [3] as the vendors mostly just refer the base score only. This leads to proposing a new improved vulnerability scoring mechanism IVSE (Improved Vulnerability Scoring with Environment representative). Proposed IVSE improves the scoring system by discarding subjective factors and introducing "environmental representative" into CVSS base score equation after deducing metric classification, values and weights. This is done by adding a metric factor namely OE (Operating Environment) type that reflects the host environment.

Metric type	Explanation	Metric Value	Metric reference
OE Type	Operating Environment of system	Linux/Windows/Mac and Others	0.5/0.4/0.7

TABLE 1: METRIC OF THE INTRODUCED ENVIRONMENT REPRESENTATIVE IN IVSE

(i) OE Type: OE type represents the Operating Environment of a system for e.g. Linux Kernel, Windows, and Macintosh etc. Impact of vulnerabilities is different on different operating systems. Windows used to be the most vulnerable Operating system. The reasons could be more number of attacks probably because of its popularity and existence of vulnerable spots. However as per the bnstechreport, in the recent years windows have comparatively less vulnerability attacks as compared to other operating systems [15]. In the recent years, Android is the most vulnerable operating system. Even Linux had vulnerability attacks more than Windows [14][16]. In CVSS the base scores does not check the effect of the host environment which can impact the vulnerability scores. In CVSS the environment is considered in Environmental scores; which however is optional. Focus of adding OE is to find out possible ways to improve available scoring schemes [6][9]. Better Judgment of vulnerabilities scores offers secure development of software's and applications. This improved scoring scheme is easier to quantify and more objective, which discards some subjective factors.



IV. BASE SCORE EQUATION OF PROPOSED IVSE

Base Score = Round to one decimal (((x*Impact) + (y*Exploitability) + (z*Environment representative)-1.5)* f (Impact))…………………….……………………………….. (4.0)
Impact = 10.41*(1-(1-CI)*(1-II)*(1-AI))…………………………………….…………. (4.1)
*Exploitability = 20*AV*AC*Au*………………………..……………………….. (4.2)
Environment Representative = 10*OE………………………..………….......... (4.3)
f(impact)= 0 if Impact=0, 1.176 otherwise……………………………….............(4.4)

Equation (4) shows the base score calculation with environment representative added in the proposed IVSE. In equation (4), *x* represents the weight value associated with Impact factor as it's proportion in the base metrics. *y* represents the weight value associated with exploitability factors as it's proportion in the base metrics. *z* represents the weight value associated with environment representative as it's proportion in the base metrics. The calculation of the subcomponents of base metric in the proposed IVSE namely impact, exploitability, environment representative and f(impact) subsequently are shown in equation 4.1, equation 4.2, equation 4.3 and equation 4.4 respectively.

As per analysis and experience, values of *x*, *y* and *z* are adjusted to 0.5, 0.3 and 0.2. 10.41 is the influence weight of impact factor account for the base metric. CI is the Confidentiality Impact which has three metric reference values none, partial and complete. II is the Integrity Impact which has none, partial and complete values and AI is Availability Impact which again has three metric reference values none, partial and complete. The environment representative is calculated as shown in equation 4.4. Where OE is the operating environment and the values for this are taken as shown in table 2.

Metric	Description	Metric Value	Reference Value
OE	OS of host	Linux/ Windows/Others	0.6/0.3/0.8

TABLE 2: METRIC FACTORS IN HOST ENVIRONMENT

V. EXPERIMENTS AND ANALYSIS

We applied the proposed scoring system on a sample set of vulnerabilities discovered in 2017 and 2018 published in the National Vulnerability Database and compared the improved base score with original CVSS scores. Comparison shows that the proposed system has a significant impact on vulnerabilities base score which led to the different scores on different operating environments of the client like Linux, Windows and Macintosh etc. The computed base scores on IVSE are shown in table 3. The vulnerabilities range between (0 - 3.9) are considered as low, (4 - 7.9) medium and (8 - 1.0) are considered as high vulnerability range. The computed scores show that the host environment as per its vulnerability has different impact on the scores. We found that somewhere the computed scores are lower than the CVSS base score in some of the operating systems environment.

Vulnerability ID	Original Base Score CVSS 2.0	Scores resulted with environment representatives		
		Linux	Windows	Mac & Others
CVE-2017-18018	1.9	2.5	1.8	3.0
CVE-2017-1672	6.8	6.5	5.8	6.9
CVE-2018-6550	3.5	3.7	3.0	4.2
CVE-2018-0744	4.4	4.6	3.9	5.1
CVE-2018-6390	4.3	4.4	3.7	4.8
CVE-2017-6374	6.4	6.1	5.4	6.5
CVE-2018-6480	6.8	6.5	5.8	6.9
CVE-2018-1092	7.1	6.7	6.0	7.2
CVE-2017-18008	4.3	4.4	3.7	4.8
CVE-2018-1000488	4.3	4.4	3.7	4.8

TABLE 3: COMPUTED BASE SCORES OF IVSE

It has been observed that there is a slight decrease in the windows base score. This is due to the more resistant characteristics of windows for vulnerabilities. More or less the new computed scores are around the CVSS base scores; however results in table 3 shows that the addition of environment representative affects the vulnerability scores on different operating environments. Moreover by adding environment representative overall CVSS score will vary if we consider temporal and environment metrics also. This new proposed formula however shows that over dependence on CVSS base score only does not give the clear picture of vulnerabilities.

VI. CONCLUSION

The accurate assessment of vulnerability scores has an important role and the host environment as well does affect the vulnerability score. In this paper some deficiencies of available scoring systems are pointed out to propose a better scoring system. A new base score equation is formulated by adding environment representative in CVSS base score equation. This equation is tested on some sample set of vulnerabilities extracted form NVD (National Vulnerability Database).The experiment results show the host environment do make a difference in computing the vulnerability scores. The reduction in score is of particular importance for security management and vulnerability prioritization quality as well. It also improves the selection of most critical vulnerabilities and helps security team to tune down their alertness. Further, it helps to balance the investment on cost saving and security related activities.

REFERENCES

[1] Julien AUSSIBAL and Laurent GALLON LIUPPA. 2008. CSySEC, IUT de Mont de Marsan "A new distributed IDS based on CVSS framework1&2" IEEE

[2] Ri Wang and Ling Gao/Qian Sun/Deheng Sun. An Improved CVSS-based vulnerability scoring mechanism", 2011 Third International Conference on Multimedia Information Networking and Security.

[3] Nada H. Sherief, Ayman A. Abdel-Hamid, Khaled M. Mahar. Threat-Driven Modeling Framework for Secure Software Using Aspect-Oriented Stochastic Petri Nets.

[4] Karen Scarfone and Peter Mell. An Analysis of CVSS Version 2 Vulnerability Scoring1. Third International Symposium on Empirical Software Engineering and Measurement

[5] Christian Frühwirth and Tomi Männistö "Improving CVSS-based vulnerability prioritization and response with context information" Third International Symposium on Empirical Software Engineering and Measurement

[6] Ayodele Oluwasen Ibidap, Pavol Zavarsky, Dale Lindskog, Ron Ruhl. An Analysis of CVSS v2 Vulnerability Scoring.

[7] Laurent GALLON and LIUPPA. On the impact of environmental metrics on CVSS scores. International Conference on Social Computing / IEEE International Conference on Pricacy, Security, Risk and Trust. IEEE.

[8] Sami Petäjäsoja, Heikki Kortti, Ari Takanen, Juha-Matti Tirilä. 2011. IMS Threat and Attack Surface Analysis using Common Vulnerability Scoring Syste. IEEE

[9] Georgios Spanos, Angeliki Sioziou, Lefteris Angelis " WIVSS: A New Methodology for Scoring Information Systems Vulnerabilities"

[10] Peter Mell, Karen Scarfone, Sasha Romanosky "A Complete Guide to the Common Vulnerability Scoring System Version 2.0".

[11] Heqing Huang, Feng Zhao, Min Ye. 2010. Estimate the Influential level of Vulnerability instance based on hybrid ranking for dynamic network attacking scenarios. ISSPA

[12] Laurent Gallon LIUPPA "On the Impact of Environmental metrics on CVSS Scores" IEEE International Conference on Social Computing.

[13] HyunChul Joh1, and Yashwant K. Malaiya1Okamoto "Defining and Assessing Quantitative Security Risk Measures Using Vulnerability Lifecycle and CVSS Metrics" Int'l Conf. Security and Management | SAM'11

[14] Available online at, https://techtalk.gfi.com/2015s-mvps-the-most-vulnerable-players.

[15] 2016. Available online at, https://bnstechreport.wordpress.com/2017/01/22/top-16-most-vulnerable-operating-systems-in.

[16] 2017. Available online at, http://www.cybrnow.com/10-most-vulnerable-os-of.