Temporal Metrics for Software Vulnerabilities

Ju An Wang, Fengwei Zhang, &
Min Xia
Southern Polytechnic State University
1100 South Marietta Parkway
Marietta, GA 30060
(01) 678-915-3718
jwang@spsu.edu

ABSTRACT
It is widely recognized that metrics are important to information security. Metrics can be an effective tool for companies and information security professionals to measure, control, and improve their security control and mechanisms. However, common security metrics are often qualitative, subjective, and informal in the sense that they are lacking formal models and automated support. This paper discussed our work on temporal metrics for software vulnerabilities based on the Common Vulnerability Scoring System 2.0. A mathematical model is provided to calculate the severity and risk of a vulnerability, which is time dependent including exploitability, remediation level, and report confidence attributes of an information asset in a computing environment. A prototype of an automated tool, CVSSWizzard, is illustrated with examples.

Categories and Subject Descriptors
C.2.0 [**Computer-Communication Networks**]: General [Security and protection]; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection;

General Terms
Measurement, Security, Verification.

Keywords
Information Security, Threats and vulnerabilities, Metrics and measurement, Common Vulnerability Scoring System

1. INTRODUCTION
Vulnerability evaluation plays a central role for security posture and risk management. Vulnerability refers to flaws or weakness in a system's design, implementation, or operation and management that could be exploited to violate the system's security policy. Any flaw or weakness in an information system could be exploited to gain unauthorized access to, damage or compromise the information system. In order to evaluate vulnerability, we need well-defined security metrics to measure the severity level of a vulnerability based on scientific, systematic, and quantitative approaches. Without well-defined security metrics, companies find themselves difficult to compare and select different security options accurately. Cost-benefit analysis and ROI (return on investment) calculations are becoming standard pre-requisites for any information security product sale or purchase.

The CVSS (Common Vulnerability Scoring System) [1] 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.

If a vulnerability has no impact on confidentiality, integrity, or availability, the BaseScore of the vulnerability will be zero. However, as [2] pointed out, 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 confidentiality, integrity, and availability impact plays an important role in CVSS calculation, [2] proposed to define *f(impact)* as a multiple tiered function, such that the base score reflects the impact on confidentiality, integrity, and availability.

As shown in [2], our new formula for the BaseScore generates more reasonable vulnerability scores for common vulnerabilities. Since the TempScore depends on the BaseScore, our new formula discussed in [2] produces more reasonable temporal metrics as well. However, we believe that a new formula for temporal metrics is in need in addition to the new formula for the BaseScore in [2]. The rest of the paper is organized in the following way: Section 2 proposes our new Temporal Metrics formula. Section 3 presents a possible change to the Environmental formula with a brief introduction to our automated tool. The last section discusses further research topics, followed by references.

2. TEMPORAL METRICS
Temporal metrics represent the time dependent features of the vulnerabilities. Temporal metrics were defined in [1] as the production of the following four factors:

*TempScore = BaseScore * Exploitability * RemediationLevel * ReportConfidence*

BaseScore is calculated by the base metrics. The *Exploitability* measures the current state of exploit techniques or code

CSIIRW'08, May 12–14, 2008, Oak Ridge, TN, USA.

availability. For a vulnerability, if the easy-to-use exploit code is available, it increases the number of potential attackers as script kiddies can launch an attack with the exploit code. Thus high exploitability increases the severity of the vulnerability. The more easily a vulnerability can be exploited, the higher the vulnerability score. The possible values for *Exploitability* are: Unproven (0.85), Proof-of-Concept (0.90), Functional (0.95), High (1.00), and Not-Defined (1.00).

The *RemediationLevel* measures the degree of severity of a vulnerability in terms of its remediation like temporary fix or official fix. If a vulnerability has less official fix or less permanent fix, it should have higher vulnerability score. The possible values for *RemediationLevel* are: Official Fix (0.87), Temporary Fix (0.90), Workaround (0.95), Unavailable (1.00), and Not Defined (1.00).

The *ReportConfidence* measures the degree of confidence in the existence of the vulnerability and the credibility of the known technical details. If a vulnerability is validated by the vendor or other reputable sources, it has higher score. The possible values for *ReportConfidence* are: Unconfirmed (0.90), Uncorroborated (0.95), Confirmed (1.00), and Undefined (1.00).

According to the discussion above, the *TempScore* will have a value fall into the following range:

$$0.66555 * BaseScore \leq TempScore \leq BaseScore$$

As temporal score measures the time dependent features of the vulnerability, it should not be confined in such a narrow range. On one hand, the *Expoitability* increases as more advanced exploit technology and tools become available. On the other hand, as the vulnerability is known to more and more people, it increases its temporal metrics value as it is subject to more exploits. Of course, the temporal score of a vulnerability will decrease when there is an official fix. For example, as CISCO published its security advisory revisions from Revision 1.0 and 1.1 in 2007 December to its Revision 1.2 in 2008 January, the *RemediationLevel* of the vulnerability score decreased [5]. The original CVSS formula seems to account for the official fixing vulnerability from the vendor side more than the exploitability increase from the user perspective.

In order to reflect the time-dependency of temporal metrics, we proposed to adjust the Exploitability values as described in the table below:

Table 2. New Exploitability values

Metric Name	*Original Value*	*Adjusted Value*
Unproven	0.85	1.00
Proof-of-Concept	0.90	1.05
Functional	0.95	1.10
High	1.00	1.15
Not Defined	1.00	1.00

Comparing with the original temporal metrics formula, this modification allows the temporal metrics to be greater than the basic metrics as shown in the following inequation:

$$0.783 * BaseScore \leq TempScore \leq 1.150 * BaseScore$$

Given that the temporal score may be over 10, we could pick up the minimum value between 10 and the new temporal score value to make the temporal metrics values consistent with basic scores and environmental scores with a maximum of 10. Below is the new temporal formula:

$$TempScore = min\{10, OldTemp\}, \text{ where}$$
$$OldTemp = BaseScore * Exploitability * RemediationLevel * ReportConfidence.$$

Example 1: A vulnerability exists in the CISCO Firewall Services Module [9] with a Common Vulnerabilities and Exposures (CVE) identifier CVE-2007-5584. CISCO provided its base score (7.8) and temporal score (7) using CVSS 2.0 with the following parameters:

Access Vector: Network; Access Complexity: Low; Authentication: None; CI:None; II:None; AI:Complete.

Exploitability:Functional; Remediation Level:Workaround; Report Confidence:Confirmed.

With the same parameter values, our formulas deliver its base score as 5.19 and temporal score as 5.42, which shows that the temporal score is actually higher than its base score. This is because the Exploitability keeps "Functional" while there is no official fix ("Workaround" was used for Remediation Level value) for this vulnerability at the time of calculation. In other words, the vulnerability has been around for some time but the vendor has not released any official solution for it yet. In this case the temporal score reflects the time dependent measurement for the vulnerability.

Example 2: A vulnerability in the CISCO implementation of Multicast Virtual Private Network (MVPN) is subject to exploitation that can allow a malicious user to create extra multicast states on the core routers or receive multicast traffic from other Multiprotocol Label Switching (MPLS) based VPN by sending specially crafted messages [10]. This vulnerability is documented as CISCO Bug ID CSCsi01470 and has been assigned CVE ID CVE 2008-1156. CISCO provided its base score (7.5) and temporal score (6.2) using CVSS 2.0 with the following parameters:

Access Vector: Network; Access Complexity: Low; Authentication: None; CI:Partial; II: Partial; AI: Partial.

Exploitability:Functional; Remediation Level:Official-Fix; Report Confidence:Confirmed.

With the same parameter values, our formulas deliver its base score as 4.49 and temporal score as 4.29, which shows that the temporal score is smaller than its base score. The main reason for this is due to the fact that CISCO has released free software updates that address this vulnerability officially. Note that in both examples, our base scores are smaller than the corresponding CVSS base scores. This is because our f(impact) is defined completely different from that in the original CVSS formula [2].

3. ENVIRONMENTAL METRICS AND AUTOMATED TOOL

Another supporting evidence for this new temporal formula is its application in calculating the environmental score. Similar to the temporal formula, we proposed the following new formula for the environmental formula:

$$EnvScore = (AdjustedTempScore + (10*AdjustedTemScore) * CollateralDamagePotential) * TargetDistribution * F(AdjustTempScore), \text{ where}$$

$$AdjustedTempScore = min\{10, AdjustedBaseScore * Exploitability * RemediationLevel * ReportConfidence\}$$

$$AdjustedBaseScore = ((0.6 * AdjustedImpact) + (0.4 * Exploitability) – 1.5) * f(AdjustedImpact)$$

$$AdjustedImpact = min\{10, 10.41 * (1 – (1 – ConfImpact * ConfReq) * (1 – IntegImpact * IntegReq) * (1 – AvailImpact * AvailReq))\}$$

$$F(AdjustedTempScore) = \begin{cases} 0, & if\ \_\ AdjustedTempScore=0 \\ 1, & otherwise \end{cases}$$

$$F(AdjustedImpact) = \begin{cases} 0, & if\ \_\ Adjusted\ \text{Im}\ pact=0 \\ 1.176, & otherwise \end{cases}$$

From the original CVSS formula, the environmental metrics is calculated with the following formula:

$$EnvScore = (AdjustedTemporal + (10\text{-}AdjustedTemporal) * CollateralDamagePotential) * TargetDistribution$$

If both the base score and the temporal score are 0s, we may have an environmental score as high as 5 based on this formula if we set *CollateralDamagePotential* to be 0.5 and *TargetDistribution* to be 1. This is obviously un-reasonable from our common sense as well as practical understanding of environmental score of a vulnerability. Based on our new environmental formula, however, this is not possible. That is, the environmental score will be 0 if both basic and temporal metrics for a vulnerability are 0s.

We have implemented an automated tool, CVSSWizzard, to help calculate base, temporal, and environmental scores of software vulnerabilities. Unlike most CVSS calculators such as [11], our tool implemented a revised version of formulas to calculate base, temporal, and environmental scores [2]. Moreover, our tool has a step-by-step guideline for each parameter used in the calculation. For instance, when the user has to supply the value for "Report Confidence", our tool will pop-up the definition of this parameter, and offer a list of options with detailed explanation, as shown in the following figure.

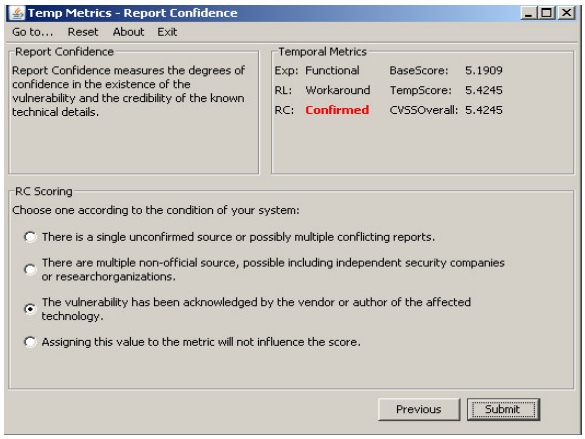

Figure 1. A screen capture for the automated tool

4. Conclusion and Discussion

In addition to provide a new formula for scoring software vulnerabilities, we developed an automated tool, CVSSWizzard, to calculate vulnerability metrics with a user-friendly interface. We will demonstrate this tool at the conference.

Software security is essential for information security or cyberspace security in general. Since the threat landscape is very dynamic, it is necessary to measure temporal metrics for software security vulnerabilities. During the years of 2002-2005, for instance, Microsoft Windows worms like Blaster, Nachi, Sasser and Zotob infected a large number of systems on the Internet. However, there have not been any new large-scale worms targeting Windows services since 2005 [7]. On the other hand, vulnerabilities found in different forms and on different platforms. A great number of client-side vulnerabilities have been identified on multiple operating systems. We believe that many aspects of temporal metrics for software vulnerabilities merit further research, which include the fundamental definition of temporal metrics, parameters, and the mathematical formula calculating the temporal score.

The approach presented in Section 4 does allow temporal scores to have values smaller or greater than basic scores. However, the current version of our temporal score formula does not include any "time" parameter in its calculation. Since the temporal metrics deliver time-dependent measurement, we believe that it is reasonable to include a time factor into the calculation of temporal metrics. On the other hand, we would like to develop a metric to measure software *trustworthiness* based on it historical data on its temporal metrics. For instance, if a monthly report is generated for software product *A* and a similar software product *B* in terms of their temporal scores, we should be able to conclude which software product is more trustworthy. We called software product *A* is *more trustworthy* than *B* if the integrated temporal scores of vulnerabilities for *A* is smaller than that of *B*. Along with the same line, we should be able to formally compare and predict the *reliability* of two similar software products based on their historical trustworthiness.

5. REFERENCES

[1] 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).

[2] J. A. Wang, M. Xia, and F. Zhang, "Metrics for Information Security Vulnerabilities, *Journal of Applied Global Research,* Volume 1, No. 1, 2008, pp. 48-58.

[3] 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.

[4] Oracle Corporation, The Critical Patch Update, http://www.oracle.com/technology/deploy/security/critical-patch-updates/cpuoct2006.html.

[5] CISCO Security Advisory: Application Inspection Vulnerability in CISCO Firewall Services Module, http://www.cisco.com/en/US/products/products_security_advisory09186a008091b11d.shtml.

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

[7] The SANS Institute, SANS Top-20 2007 Security Risks, http://www.sans.org/top20/, accessed on February 20, 2008.

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

[9] CISCO Document ID: 100389, http://www.cisco.com/warp/public/707/cisco-sa-20071219-fwsm.shtml.

[10] CISCO Document ID: 100374, http://www.cisco.com/warp/public/707/cisco-sa-20080326-mvpn.shtml.

[11] National Vulnerability Database, Common Vulnerability Scoring System Calculator, http://nvd.nist.gov/cvss.cfm?calculator.