Jaewon Lee, CISA, CGEIT, CRISC, CIA, CRMA, is an IT security, IT risk and IT audit expert. He has been working in the finance industry for more than 15 years and is currently writing a bank-wide principle-based IT risk policy at ING Bank in The Netherlands. He can be reached at *jae-won.lee@ing.nl.*

An Enhanced Risk Formula for Software Security Vulnerabilities

As enterprises increasingly rely on IT to succeed, effective IT risk management has become an essential component of IT governance.[1] In conjunction with this, there are various studies to address risk through the software development life cycle,[2] while others are interested in risk in the production environment.[3] There are also studies to calculate risk as a whole[4] and others to address specific parts of risk components, such as a study to estimate the likelihood driven by the attack-tree approach.[5]

However, no study has explicitly enhanced the current risk formula (Risk = Likelihood × Impact) to embrace the IT natures and characteristics, such as the IT software architectural aspects (i.e., complexity), various security requirements (i.e., confidentiality, integrity and availability) and availability of solutions to respond to risk. Hence, the Common Vulnerability Scoring System 2.0 (CVSS) is used here to provide an enhanced risk formula.[6]

LIMITATIONS OF THE CURRENT RISK FORMULA

First, the current risk formula offers no clear distinction in the usage of criticality and risk rating. The differences between criticality and risk rating are as follows:

- Risk is the combination of the probability of an event and its consequence. In general, this can be explained as: Risk = Likelihood × Impact.[7] In particular, IT risk is the business risk associated with the use, ownership, operation, involvement, influence and adoption of IT within an enterprise.[8]
- Criticality analysis is an analysis to evaluate resources or business functions to identify their importance to the enterprise.[9] This can be explained as: Criticality = Probability × Severity.[10]

Probability is a statistical way of measuring likelihood (Probability = Likelihood). However, severity is not necessarily equal to impact (Severity ≠ Impact). Additionally, criticality as used here relates to vulnerabilities, while risk relates to threats. Hence, criticality and risk rating do not have the same definitions. For instance, once a buffer overflow attack succeeds, an attacker can take the administrator's privilege, depending on the configurations, which, in itself, can be considered a severe vulnerability (criticality), while the potential impact can be considered the next step (risk).

Second, neither IT characteristics nor software architectural aspects (i.e., the number of times an attacker must authenticate to exploit the issues) are explicitly embedded (subjective).

Third, confidentiality, integrity and availability are not explicitly embedded (subjective).

Fourth, software security vulnerabilities often require technical solutions (i.e., implementation of some tools to address the vulnerabilities), which may not always be available. However, the degree of availability of the solutions is not considered.

Last, the current risk formula may not be effective to evaluate the risk for any software not in the production environment yet, as no direct impact is expected while criticality still can be estimated based on the nature of the vulnerabilities. There are several studies to address this particular limitation from various aspects, such as a source-code-based software risk assessing model.[11] However, as in other studies, the current risk formula in itself is not challenged.

SOLUTION—ENHANCED RISK FORMULA

An enhanced risk formula, Risk = Criticality (Likelihood × Vulnerability Scores [CVSS]) × Impact, is proposed to derive more effective and accurate criticality as well as a risk rating for software security vulnerabilities. There are similar studies already published;[12] however, they did not address software security vulnerabilities.

COVERAGE OF SOLUTION

The enhanced risk formula is limited to software security vulnerabilities. Other vulnerable factors, such as lack of IT general controls (i.e., IT capacity management), are not considered.

There are studies to derive risk from the architectural perspective;[13] however, the architectural perspective is not further studied for the following reasons:

- The intrinsic characteristics of vulnerabilities that are constant over time and user environments (i.e., access complexity) are already included in CVSS.
- The architectural aspects are not always related to confidentiality and/or integrity. For instance, once data in a database system are stolen, the issue relies mainly on the confidentiality of the data regardless of its architecture.

The current approaches to estimate likelihood and impact are not challenged for the following reasons:

- The objective of this article is to provide an enhanced risk formula, not to challenge the current ways to estimate likelihood and impact individually.
- These are dependent on individuals and enterprises (i.e., impact is largely based on the business products).

PROOF OF CONCEPT

The first step is calculating criticality by the criticality formula: criticality = probability × severity. There are many studies that use various theories to quantify probability. For instance, the attack tree is used to calculate the likelihood[14] and the likelihood from the attack tree is also explained with the fuzzy techniques.[15] In addition, there are many sources available for an overview of the issues, such as the National Vulnerability Database (NVD)[16] and the Computer Emergency Response Team (CERT) Coordination Center.[17]

For the severity of a vulnerability, CVSS is a unique and well-recognized standard to calculate vulnerability scores, which indicate the severity of the vulnerabilities (= severity). CVSS is considered an emerging industrial standard.[18] The benefits from CVSS are:

- CVSS does not belong to any specific software products or vendors.
- CVSS reflects the IT software architectural aspects (i.e., the complexity of the attack).
- CVSS covers the security requirements (i.e., confidentiality, integrity and availability).
- CVSS includes the remediation level of the issues.
- CVSS makes the distinction between the proportion of vulnerable systems and the importance of the affected IT asset.

The likelihood driven by the attack-tree approach and CVSS demonstrate how to derive more efficient and accurate criticality of software security vulnerabilities. Using CVSS is essential as some of the limitations mentioned earlier are addressed by the CVSS calculation logic, while the ways to determine likelihood vary.

The second step is calculating risk by the enhanced risk formula, Risk = Criticality (Likelihood × Vulnerability Scoring [CVSS]) × Impact, to explain how impact can be integrated with the criticality from the first step to calculate the risk rating.

CRITICALITY CALCULATION WITH CVSS

To demonstrate the likelihood using the attack-tree approach, a scenario called insecure web servers is created in the production banking environment. The vulnerabilities and their likelihood explained in the attack tree are Address Resolution Protocol (ARP) Poisoning (Likelihood 0.02), MySQL Encoding Flaw (Likelihood 0.09) and Internet Information Server (IIS) Privilege Escalation to Root (Likelihood 0.63). The following conditions are added to calculate the CVSS scores:

- Technical conditions:
  1. Web servers should be available at all times (24/7).
  2. Windows OS and IIS are installed.
  3. A firewall/intrusion detection system (IDS) is installed in the demilitarized zone (DMZ). No internal threat is considered.
  4. Antivirus software is installed for all the servers.
- Industrial conditions:
  5. The scenario is based on the production banking environment, so the security requirements are high in general.

As the consequence of the previous conditions, the CVSS scores are calculated (**figure 1**).

CRITICALITY DISPOSITIONS

Commonly used ratings, such as high, high/medium, medium, medium/low and low are used here. The likelihood and CVSS scores are simplified and equally divided for demonstration purposes. The issues are plotted according to the likelihood and CVSS scores in **figure 2**. The different business natures and characteristics of individuals or enterprises in real-life cases are explained as well.

Figure 1—CVSS Score Calculations

CVSS Calculation	Software Security Vulnerabilities ARP Poisoning	 MySQL Encoding Flaw	 IIS Privilege Escalation to Root
Access vector	Network	Network	Network
Access complexity	Low	Low	Low
Authentication	None	None	Single Instance
Confidentiality/impact	Partial	Partial	Complete
Integrity/impact	Partial	Partial	Complete
Availability/impact	Partial	Partial	Complete
Collateral damage potential	Low (light loss)	Medium/high	High (catastrophic loss)
Target distribution	High (76-100%)	High (76-100%)	High (76-100%)
Confidentiality requirement	High	High	High
Integrity requirement	High	High	High
Availability requirement	High	High	High
Exploitability	High	Functional exploit exists	Functional exploit exists
Remediation level	Official fix	Official fix	Official fix
Report confidence	Confirmed	Confirmed	Confirmed
	= 7.9	= 8.4	= 8.7

The criticality for ARP Poisoning is rated as high/medium, and MySQL Encoding Flaw and IIS Privilege Escalation to Root are rated as high. These can be used to calculate the risk ratings for the next step.

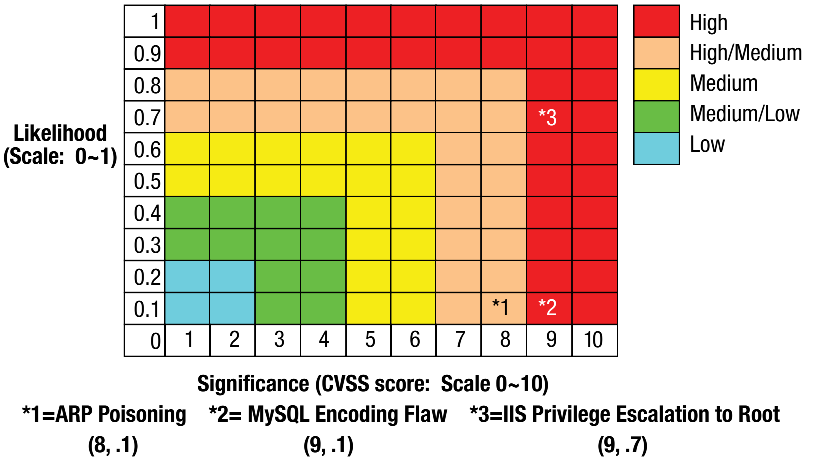
However, as the table is equally divided, various business natures and industrial characteristics are not reflected. For instance, the banking and airline businesses are often governed by more strict regulatory requirements in general. Hence, the range of the high criticality area can be intensely increased and all three issues can be rated as high.

On the other hand, businesses selling nonsensitive products, such as accessories in an offline market, may have fewer security requirements in comparison with the banking and airline industries. In this case, the range of the high-criticality area can be greatly decreased. As a consequence, all three issues can be rated as medium or even lower, although there is no change in the likelihood and CVSS scores at all. These are illustrated in **figure 3**.

As shown, the method in the first step to derive more effective and accurate criticality can be used regardless of the different natures and characteristics of individuals or enterprises. That is, the terms to define the ranges can be adjusted according to their business and industry natures without making any changes on the criticality formula, Criticality = Probability × Severity.

The figures shown here have been simplified to illustrate the concept, but practically, these need to be carefully fine-tuned to properly reflect the risk appetite and tolerance of enterprises.[19]

Figure 2—Likelihood and CVSS Scores

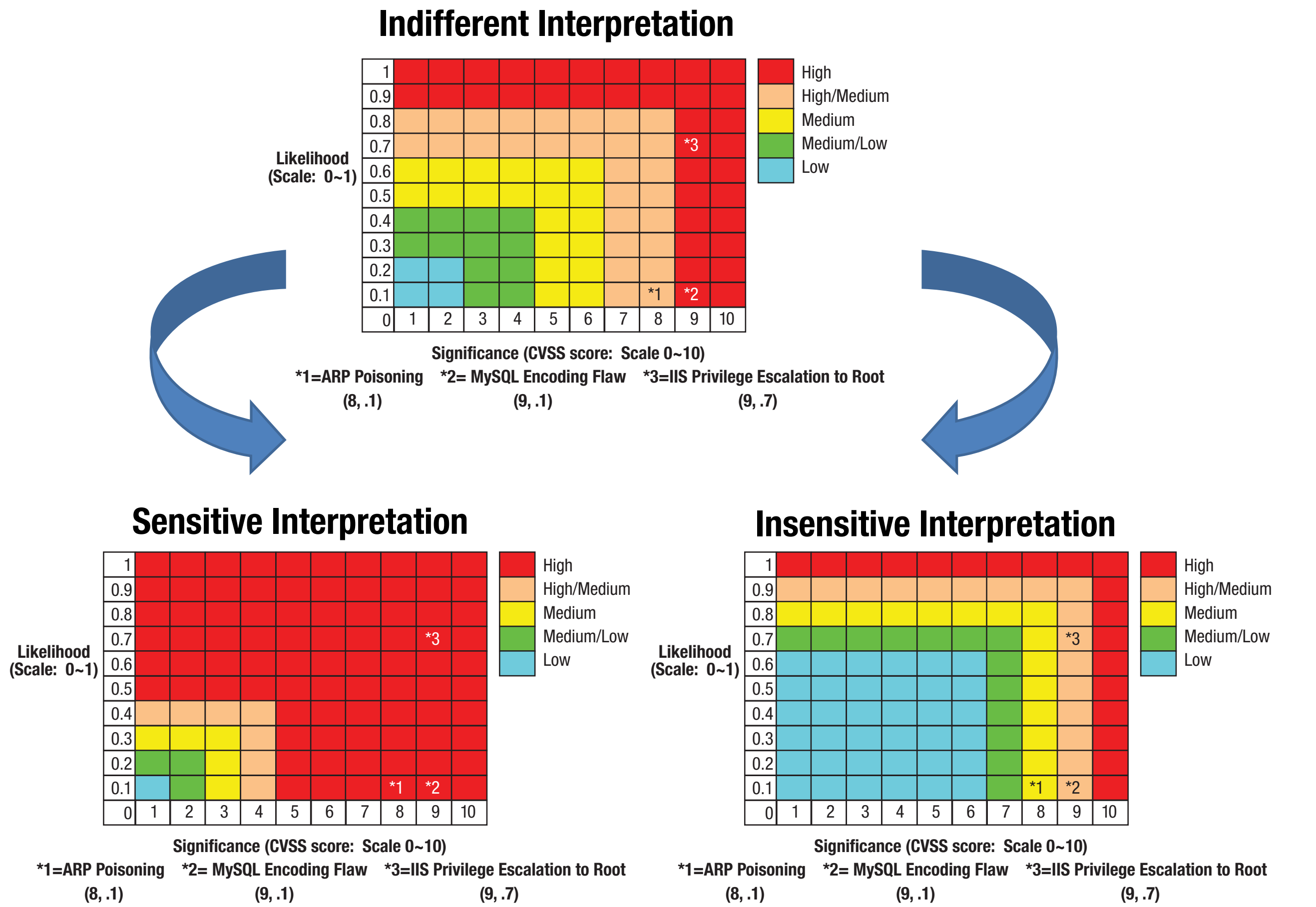


RISK DISPOSITIONS WITH ENHANCED RISK FORMULA

In the second step, the criticality from the first step should be combined with impact: Risk = Criticality (Likelihood × CVSS score) × Impact. As mentioned previously, how best to estimate the impact is not studied in this article; instead, the volume of midsized businesses is used for demonstration purposes.[20] The same rating terms as the criticality dispositions are used as well.

Similar to the first step, the criticality and the impact are equally divided in **figure 4**. The criticality ratings are

Figure 3—Criticality Interpretations



plotted according to the first step, while the amount of the impact is increased incrementally to be more practical and is randomly chosen for demonstration purposes. The different business natures and characteristics of individuals or enterprises in real-life cases are also explained.

The risk rating for ARP Poisoning is rated as high/medium, and MySQL Encoding Flaw and IIS Privilege Escalation to Root are rated as high. These can be considered the final results.

However, similar to the criticality dispositions, **figure 4** is equally divided and various business natures and industrial characteristics are not reflected. For instance, impact, about €50 million, can be perceived as high risk for small-scale businesses, while the same amount of impact can be perceived as medium or even low risk by large-scale businesses. These are illustrated in **figure 5**.

Figure 4—Criticality and Impact

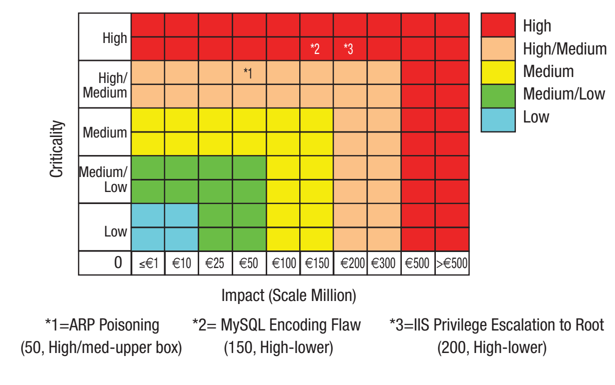
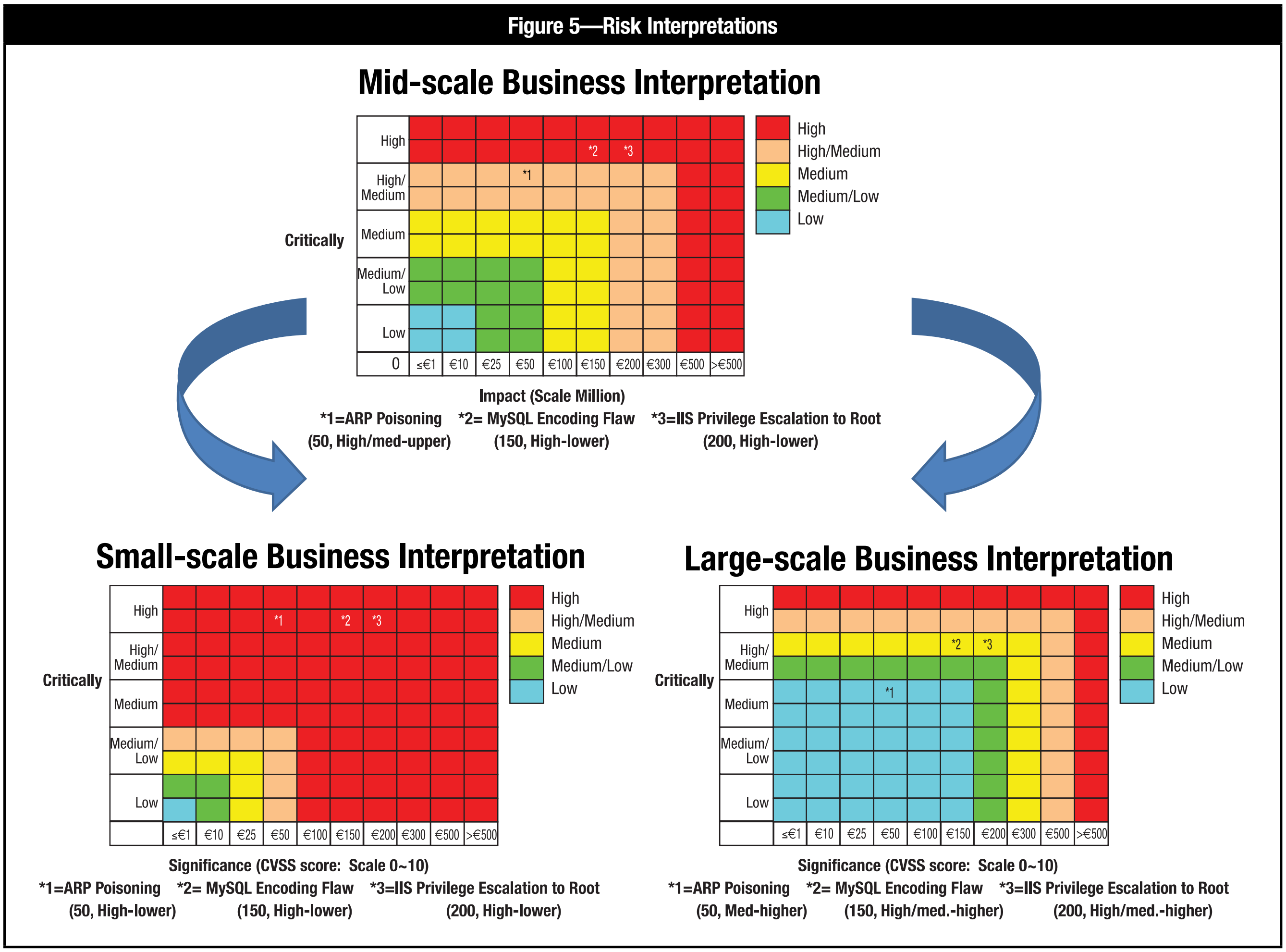


Figure 5—Risk Interpretations



Mid-scale Business Interpretation

*1=ARP Poisoning (50, High/med-upper) *2= MySQL Encoding Flaw (150, High-lower) *3=IIS Privilege Escalation to Root (200, High-lower)

Small-scale Business Interpretation

*1=ARP Poisoning (50, High-lower) *2= MySQL Encoding Flaw (150, High-lower) *3=IIS Privilege Escalation to Root (200, High-lower)

Large-scale Business Interpretation

*1=ARP Poisoning (50, Med-higher) *2= MySQL Encoding Flaw (150, High/med.-higher) *3=IIS Privilege Escalation to Root (200, High/med.-higher)

This shows that the method in the second step to derive a more effective and accurate risk rating can be used regardless of the different natures and characteristics of individuals or enterprises without making any changes to the enhanced risk formula, Risk = Criticality (Likelihood × CVSS score) × Impact.

Like the criticality figures earlier, these are also simplified to illustrate the concept. Hence, the figures need to be fine-tuned to properly reflect real-life cases.

CONCLUSION

More effective and accurate criticality for software security vulnerabilities is demonstrated by using CVSS. The enhanced risk formula, Risk = Criticality (Likelihood × Vulnerability Scoring [CVSS]) × Impact, is demonstrated to result in more effective and accurate risk ratings, which are derived from the three dimensions (likelihood, vulnerability scores and impact).

All things considered, the enhanced risk formula can partially or completely replace the current empirical judgments around IT security management, IT risk management and IT audit activities. In addition, it can also address the following:

- Criticality and risk ratings for software security vulnerabilities can be calculated separately, but the relationship is explained.
- Both IT characteristics and software architectural aspects are more clearly included.
- The method to estimate both criticality and risk rating is consistent and repeatable; all key factors are explicitly embedded into the enhanced formula.
- The enhanced risk formula still contains a certain level of subjectivity. However, as the predefined categories are already given in the CVSS calculation, this formula is more objective.
- The availability of the solution to address software security vulnerabilities is considered.

- This helps to estimate the criticality of software security vulnerabilities in the development environment as the criticality is assessed before potential impact is calculated.

The current CVSS metrics do not explicitly address the security configuration settings. However, there are studies underway to make the CVSS scoring even more accurate, flexible and representative of risk by the organization itself (Forum of Incident Response and Security Teams [FIRST][21]). Besides, the adjusted metrics for the insecure software configurations are partially provided by various studies already. For instance, the CVSS metric definitions are expanded to include settings that prevented/authorized actions and permit multiple scores per configuration issue to reflect the possible combinations of desired and actual settings without making any changes to the CVSS calculation login itself.[22] This may offer wider coverage of the enhanced risk formula going forward.

ACKNOWLEDGMENT

The author would like to express special thanks to professors Hans van Vliet and Ronald Paans at Vrije University (Amsterdam, The Netherlands) and Joris Hulstijn at Delft University of Technology, The Netherlands, for their coaching.

ENDNOTES

[1] Fischer, U.; "New Framework for Enterprise Risk Management in IT," *ISACA Journal*, vol. 4, 2008

[2] Khalid, S.; E. N. Abdeslam; H. L. Abdelwahab; *Catalog of Metrics for Assessing Security Risks of Software Throughout the Software Development Life Cycle,* International Conference on Information Security Assurance, 2008

[3] Romero, B.; M. Villegas; M. Mezal'; "Simon's Intelligence Phase for Security Risk Assessment in Web Applications," Fifth International Conference on Information Technology: New Generations, IEEE, 2008, p. 622-627

[4] Xiao, L.; Y. Qi; Q. Li; "Information Security Risk Assessment Based On Analytic Hierarchy Process and Fuzzy Comprehensive," The 2008 International Conference on Risk Management & Engineering Management, IEEE, 2008, p. 404-409

[5] Clark, K.; E. Singleton; S. Tyree; J. Hale; "Strata-Gem: Risk Assessment Through Mission Modeling," Quality of Protection Workshop, Association for Computing Machinery (ACM), October 2008, p. 51-57

[6] Mell, P.; K. Scarfone; S. Romanosky; "A Complete Guide to the Common Vulnerability Scoring System Version 2.0," Forum of Incident Response and Security Teams (FIRST), June 2007, *www.first.org/cvss/cvss-guide.html*

[7] Open Web Application Security Project (OWASP), "The Free and Open Application Security Community," Conference on Information Security and Assurance, IEEE, p. 461-465

[8] Fischer, U.; "Identify, Govern and Manage IT Risk (Part 1): Risk IT Based on COBIT Objectives and Principles," *ISACA Journal*, vol. 4, 2009, p. 29-31

[9] ISACA, *www.isaca.org/glossary*

[10] Reliability Information Analysis Center (RIAC), "Failure Mode, Effects and Criticality Analysis (FMECA)," 1993, p. 5

[11] Wong, W. E.;Y. Qi; K. Cooper; "Source Code-Based Software Risk Assessing," SAC'05, 13-17 March 2005, USA, p. 1485-1490

[12] Babut, G. B.; R. I. Moraru; L. I. Cioca; "Kinney-type Methods: Useful or Harmful Tools in the Risk Assessment and Management Process?," International Conference on Manufacturing Science and Education, 2011

[13] Bass, L.; R. Nord; W. Wood; D. Zubrow: "Risk Themes Discovered Through Architecture Evaluations," Proceedings of the Working IEEE/IFIP Conference on Software Architecture (WICSA), January 2007, p. 44-53

[14] *Op cit*, Clark

[15] Halkidis, S. T.; N. Tsantalis; A. Chatzigeorgiou; G. Stephanides; "Architectural Risk Analysis of Software Systems Based on Security Patterns," *IEEE Transactions On Dependable and Secure Computing*, IEEE, vol. 5, no. 3, July-September 2008, p. 129-142

[16] National Institute of Standards and Technology, National Vulnerability Database (NVD), *http://nvd.nist.gov/*

[17] Carnegie Mellon University, Computer Emergency Response Team Coordination Center, Software Engineering Institute, *http://www.cert.org/*

[18] Chandramouli, R.; T. Grance; R. Kuhn; S. Landau; "Common Vulnerability Scoring System," *IEEE Security & Privacy*, IEEE, November/December 2006, p. 85-89

[19] Further explanation of how to define the ratings in the figures is out of the scope of this article. It is well explained in ISACA, *COBIT® 5 for Risk*, 2013 and ISACA, *The Risk IT Practitioner Guide*, 2009.

[20] North, D.; R. Baldock; I. Vickers; "Research Into Mid-Size Business Growth," Middlesex University, UK, p. 1

[21] Forum of Incident Response and Security Teams (FIRST), *www.first.org/cvss*

[22] Scarfone, K.; P. Mell; "Vulnerability Scoring for Security Configuration Settings," Quality of Protection, Association for Computing Machinery (ACM), October 2008, p. 3-7

The *ISACA Journal* is published by ISACA. Membership in the association, a voluntary organization serving IT governance professionals, entitles one to receive an annual subscription to the *ISACA Journal.*

Opinions expressed in the *ISACA Journal* represent the views of the authors and advertisers. They may differ from policies and official statements of ISACA and/or the IT Governance Institute® and their committees, and from opinions endorsed by authors' employers, or the editors of this *Journal*. *ISACA Journal* does not attest to the originality of authors' content.

www.isaca.org