Journal of Korea Multimedia Society Vol. 18, No. 2, February 2015(pp. 199-206)
http://dx.doi.org/10.9717/kmms.2015.18.2.199

Assessing Web Browser Security Vulnerabilities with respect to CVSS

HyunChul Joh†

ABSTRACT

Since security vulnerabilities newly discovered in a popular Web browser immediately put a number of users at risk, urgent attention from developers is required to address those vulnerabilities. Analysis of characteristics in the Web browser vulnerabilities can be used to assess security risks and to determine the resources needed to develop patches quickly to handle vulnerabilities discovered. So far, being a new research area, the quantitative aspects of the Web browser vulnerabilities and risk assessments have not been fully investigated. However, due to the importance of Web browser software systems, further detailed studies are required related to the Web browser risk assessment, using rigorous analysis of actual data which can assist decision makers to maximize the returns on their security related efforts. In this paper, quantitative software vulnerability analysis has been presented for major Web browsers with respect to the Common Vulnerability Scoring System. Further, vulnerability discovery trends in the Web browsers are also investigated. The results show that, almost all the time, vulnerabilities are compromised from remote networks with no authentication required systems. It is also found that a vulnerability discovery model which was originally introduced for operating systems is also applicable to the Web browsers.

Key words: Software Security Vulnerability, Web Browser, CVSS, VDM, AML

1. INTRODUCTION

These days, a web browser is the most important application, which provides the connectivity to the Web servers on the Internet. Nevertheless, from the early age, numerous security holes have been discovered in the Web browsers. In fact, more than two-third of attacks to the Internet users exploit vulnerabilities of browsers or their plug-ins [1]. Many of the security bugs provide attackers or malicious users opportunities to bypass the security barrier, and the Web browser vulnerabilities represent one of the main avenues for the spread for the viruses and worms. However, in spite of the risks involved, the convenience and dynamic technical functionality offered by the Web browsers make them indispensable.

Web browsers are used for variety of purposes, such as personal entertainments, eLearning, online banking, or even highly confidential governmental tasks. Consequently, new vulnerabilities discovered in the Web browser put millions of the Web users at risk, requiring urgent attention from developers to address there vulnerabilities. Naturally, there have been significant concerns about possible exploitation of security holes in the systems because of their vulnerabilities which are now subject to increasing everybody's attention. As a result, there are considerable discussions of Web browser security in recent years. However, in many cases, those studies are focused on detection and prevention of individual vulnerabilities. Al-

※ Corresponding Author : HyunChul Joh, Address: (712-701) Gamasilgil 50, Hayangup, Gyeongsan, Gyeongbuk, Korea, TEL : +82-53-600-5563, FAX : +82-53-600-5579 , E-mail : joh@kiu.ac.kr
Receipt date : Nov. 29, 2014, Revision date : Dec. 30, 2014
Approval date : Jan. 12, 2015
† Dept. of Computer Eng., College of IT Convergence, Kyungil University
※ This research was supported by the intramural research program in Kyungil university.

though quantitative data is sometimes cited, often there is no significant critical investigation.

Software vulnerabilities can be defined as software defects or weaknesses in the security system which might be exploited by malicious users causing loss or harm [2]. Those vulnerabilities are great concern since they provide attackers the ability to gain full control of the system or leakage of highly sensitive information. In this paper, first we quantitatively examine the vulnerability discovery process on the four Web browsers (Internet Explorer (IE), Firefox (FX), Chrome(CR) and Safari (SF)) by applying the AML [3] vulnerability discovery model. And then, we investigate the secureness of the four Web browsers with respect to the CVSS.

Table 1 shows the number of vulnerabilities for each Web browser with initial release dates and market share information. Higher market share means more efficient for malicious users because they would find it more profitable and satisfying to devote their time on software. As a result, a smaller number of known vulnerabilities does not necessarily means a more secure software system. In Table 1, we cannot say that Safari is more secure than others, only because of the small number of vulnerabilities.

A quantitative analysis allows both developers and end-users to assess the potential exposure to exploitation risks. The developers can plan testing and allocation of resource more efficiently for software updates and patches and the end-users can use vulnerability discovery models to choose their web browsers and determine what safety measures to use.

Table 1. The four Web browsers

	IE	FX	CR	SF
# of Vul.†	1044	1096	979	517
Released date	Aug. 1995	Nov. 2004	Sep. 2008	Jan. 2003
M.Share‡	57.32%	18.09%	16.21%	5.67%

† number of vulnerabilities http://nvd.nist.org/ (Sep. 27th2014)

‡ market share http://www.netmarketshare.com/ (Sep. 27th2014)

A two-page long preliminary version of this work was presented as a conference paper [4]. Here, we newly introduced vulnerability discovery process section and added more insights and analysis. The rest of the paper is organized as follows. Section 2 presents some of the related works and section 3 reviews CVSS which requires readers to understand later section. Section 4 investigates vulnerability discovery process in the Web browsers, and Section 5 analyzes each element from the CVSS base metric. Section 6 concludes this work.

2. RELATED WORKS

Frei et al. [5] have shown that a significant number of the Internet users are exposed at risk because many of them tend to delay updating the Web browsers and plug-ins when new patches are available. In the paper, they quantified the risk posed by delayed patching. Duebendorfer and Frei [6] have further investigated the web browser updates for four different browsers and concluded that silent updates are the most effective mechanism for users. Acer and Jackson [7] question the view that browsers with infrequent security patches are safer. They propose methods for evaluating browser security that take into account new industry practices such as silent patch deployment. Grosskurth and Godfrey [8] have used a semi-automated analysis method to investigate the architecture and evolution of web browsers. They have examined different strategies for code reuse, emergent domain boundaries, convergent evolutions, and debate between open and closed source development approaches.

Schryen [9] has empirically examined the vulnerability discovery processes in several software systems and found that many of the systems

shows a significant linear or piecewise linear relationship between time and the cumulative number of published vulnerabilities, but did not investigate the underlying causes of the linear growth.

There are important factors that impact the vulnerability discovery rate for a product. The most significant among them are code size, software age, popularity and software evolution. Several studies [10-13] have examined the relationship between the code size and the number of defects. The studies suggest that the number of defects increases with code. The first order approximation assumes a linear relationship between the code size and the number of defects, which allows definition of the concept of defect density. Since the vulnerabilities are a class of defects, a similar measure called vulnerability density [14] can be defined.

Vulnerability Discovery Models (VDMs) describe the discovery of vulnerabilities with the passage of time. A few vulnerability discovery models have recently been proposed. One of the most well-known VDMs is the Alhazmi-Malaiya Logistic (AML) model [3] which was originally proposed and validated for operating systems. Joh and Malaiya [15] compares AML with other types of S-shaped VDMs based on the skewness in examined datasets. It shows that AML and Gamma distribution based model perform better than other S-shaped models with skewed left and right datasets respectively.

Compared with other software systems such as operating systems and office software products, newer versions of the Web browsers tend to be released faster. A new version of a software system adds new functions and implements some defect fixes. However, a new version does not necessarily imply a reduced number of vulnerabilities since the new codes can inject new vulnerabilities.

3. CVSS METRICS

The Common Vulnerability Scoring System (CVSS) [16] has been adopted by many IT vendors to measure security vulnerabilities since its first launch in 2004. The CVSS scores for known vulnerabilities are readily available on the majority of public vulnerability databases on the Web, such as NVD (http://nvd.nist.gov). The CVSS score system provides vendor independent framework for communicating the characteristics and impacts of the known vulnerabilities. Security analysts do not need to think about qualitative evaluation of vulnerability severity when they estimate it because CVSS designed to be quantitative method in the final scores in each vulnerability.

The scoring system is now on its second version which is finalized its design in June 2007, and currently its third version is ready to be released (http://www.first.org/cvss). The CVSS is composed of three metric groups: base, temporal and environmental. It attempts to evaluate the degree of risks posed by vulnerabilities, so mitigation efforts can be prioritized. The score ranges [0.0, 10.0]; scores close to 0.0 indicates more stable whereas scores close to 10.0 means more vulnerable to exploitation and causes more serious outcome.

The base metric group, ranges of [0.0, 10.0], represents the intrinsic and fundamental characteristics of a vulnerability, so the score is not changed over time. The base metric has two sub-scores of exploitability and impact sub-scores. The two sub-scores are also ranges of [0.0, 10.0]. The exploitability sub-score captures how a vulnerability is accessed and whether or not extra conditions are required to exploit it while the impact sub-score measures how a vulnerability will directly affect an IT asset as the degree of losses in confidentiality, integrity, and availability.

The exploitability sub-score is composed by three elements of access vector (AV), access complexity (AC), and authentication (Au). The access vector reflects how the vulnerability is exploited in terms of local (L), adjacent network (A), or net-

work (N). The access complexity measures the complexity of the attack required to exploit the vulnerability once an attacker has gained access to the target system in terms of High (H), Medium (M), or Low (L). The authentication counts the number of times an attacker must authenticate to a target in order to exploit a vulnerability in terms of Multiple (M), Single (S), or None (N).

On the other hand, the impact sub-score is composed by the three key aspects in information security components: confidentiality, integrity and availability. The impact attributes are all assessed in terms of None (N), Partial (P), or Complete (C).

The temporal and environmental metrics are used to augment the base metrics and depend on the target system and changing circumstances. In this paper, these two metrics are not utilized, so they are not explained here.

4. VULNERABILITY DISCOVERY TRENDS IN WEB BROWSERS

Fig. 1 shows the AML model [3] representing the relationship among software age, cumulative number of vulnerabilities and the discovery rate. The AML assumes that during the initial learning phase, very few vulnerabilities are found. During the next phase, termed the linear phase, a steady stream of vulnerabilities is discovered. In the final saturation phase, the vulnerability discovery rate declines. The durations implicitly depend on factors such as market share or undetected number of vulnerabilities remaining. In the figure, the bell-shaped dashed line shows the instantaneous discovery rate for the vulnerabilities whereas the S-shaped solid line represents the cumulative number of vulnerabilities. Market share is significant factors impacting the effort expended in exploring potential vulnerabilities. A higher market share provides more incentive to explore and exploit vulnerabilities. The effect of the market share rise and fall is implicit in the AML model [17].

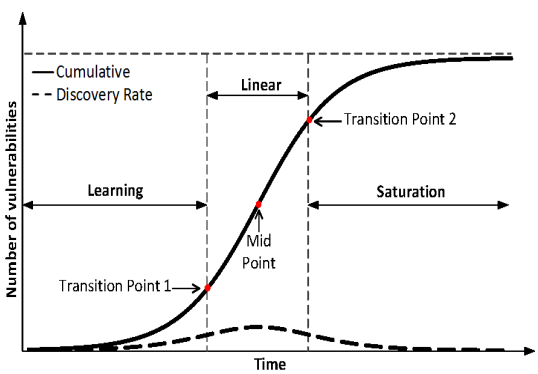

Fig. 1. Relationship between software age and vulnerabilities represented by AML.

Equation (1) gives us the three-parameter AML model where A, B and C are empirical parameters and Ω(t) represents the total number of vulnerabilities discovered at time t.

$$\Omega(t) = \frac{B}{BCe^{-ABt} + 1} \tag{1}$$

Notice that when time t goes to the infinity, B becomes the eventual number of vulnerabilities discovered in a software system. Fig. 2 shows the AML model fitting on the vulnerability discovery trends for the four Web browsers, and Table 2 displays the model parameters. There are four model fittings in each Web browser, total number of vulnerabilities, high (CVSS score 7~10), medium (CVSS score 4~6.9), and low (CVSS score 0~3.9) severity vulnerabilities. All the corresponding red solid lines represent the AML fitting results. The numbers in the figure represent Pearson's correlation coefficients (R2 values in Table 2). All the numbers are close to 1, which indicates that the model fittings are significant.

Table 3 shows the calculated transition point 1 and 2, which are located between the learning and linear phases, and the linear and saturation phases respectively in Fig. 1. The equation of how to calculate the two points is well described in [18]. Except Internet Explorer, the other three browsers already passed the transition point 2 at this point of November 2014. This signifies that the three Web browsers are now in the saturation phase. In

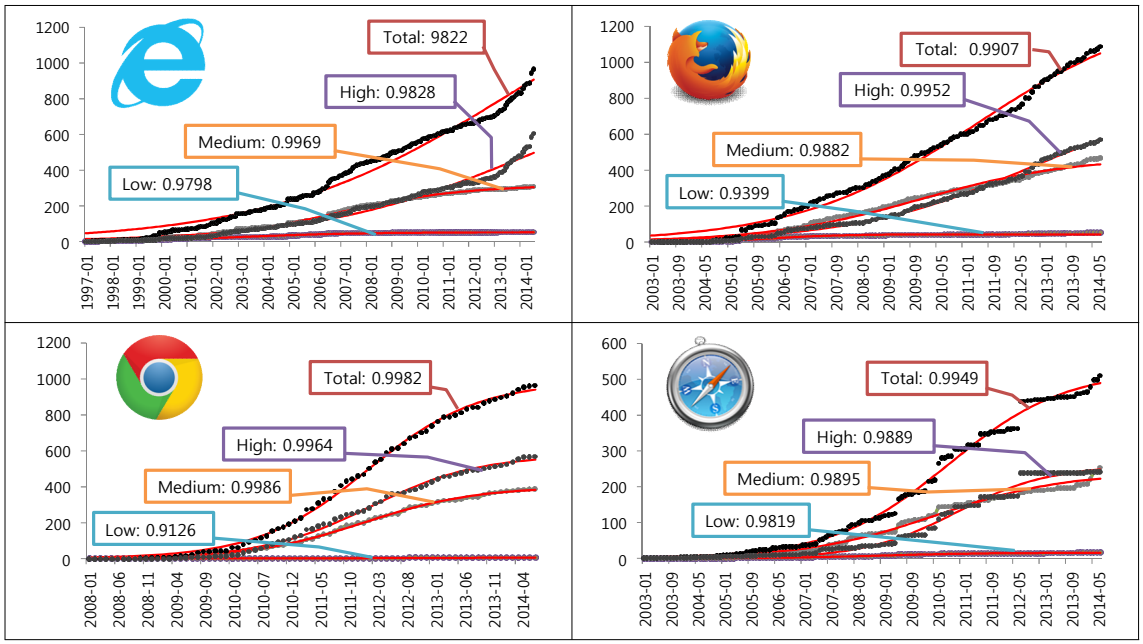

Fig. 2. AML model fitting according to CVSS Severity levels.

Table 2. AML model fitting parameters on Fig. 2 and R2 values

Browser	Severity	A	B	C	R2
IE	Total	1.09E-05	1638.400	0.021	0.982
	High	2.19E-05	940.368	0.073	0.982
	Medium	1.06E-04	317.543	0.167	0.996
	Low	6.87E-04	51.908	0.274	0.979
FX	Total	2.70E-05	1323.499	0.028	0.990
	High	4.34E-05	834.715	0.085	0.995
	Medium	9.03E-05	472.705	0.074	0.988
	Low	2.14E-03	42.909	0.865	0.939
CR	Total	1.05E-04	985.489	0.171	0.998
	High	1.82E-04	577.468	0.337	0.996
	Medium	2.53E-04	401.190	0.360	0.998
	Low	4.79E-04	56.907	1.144	0.912
SF	Total	1.08E-04	526.181	0.372	0.994
	High	2.77E-04	259.162	3.713	0.988
	Medium	2.14E-04	240.112	0.405	0.989
	Low	4.01E-03	15.766	6.293	0.981

Table 3. Transition Points defined by AML model

	IE	FX	CR	SF
Transition Point 1	May. 2007	Apr. 2008	Jan. 2011	Oct. 2008
Transition Point 2	Sep. 2019	Jun. 2014	Feb. 2013	Aug. 2012
Status in Nov.2014	Linear	Saturation	Saturation	Saturation

other words, Firefox, Chrome and Safari are in stable state, at least in terms of vulnerability discovery rate point of view. However, the AML model's assumption is that a significant chunk of codes are

not going to be introduced into a system anymore. Hence, if newer versions, having new codes, are released in the future, then the transition points could be changed.

5. ANALYSIS OF EXPLOITABILITY AND IMPACT SUB-SCORES

Fig. 3 shows the number of each value in exploitability and impact sub-score groups from CVSS. For AV and Au, in all the Web browsers, N (Network) and N (None) have the most of numbers. This indicates that exploitations are from remote networks, and if we have at least one authentication process in our systems, it is a lot safer than systems having zero authentication required. In the AC category, the majority of them are M and L. There are verysmall incidents of Hs. It implies that more complex systems are a lot less chances to be targeted. For the Impact sub-score, C (complete) takes place the highest numbers for the all three categories (Confidentiality, Integrity and Availability) in IE, FX and SF whereas, interestingly, P (partial) is the most shown letter in CR.

Table 4 emphasize the findings from Fig. 3. The table shows the top three individual combinations having the biggest number of vulnerabilities. For the Exploitability sub-score, vulnerabilities are occurred most of the time at the combination of <AV:N, AC:M/L, Au:N>, which means majority of systems are compromised <remotely with middle/low complexity and no authentication>. For the Impact sub-score, frequently, a compromised vulnerability let attackers completely gain IT assets in terms of Confidentiality, Integrity and Availability: the triple Cs in the table.

6. CONCLUSION

This paper applies the AML vulnerability discovery model on the four Web browsers and analyzes CVSS base scores quantitatively. The results indicate that the AML model which originally proposed and validated for operating systems are also

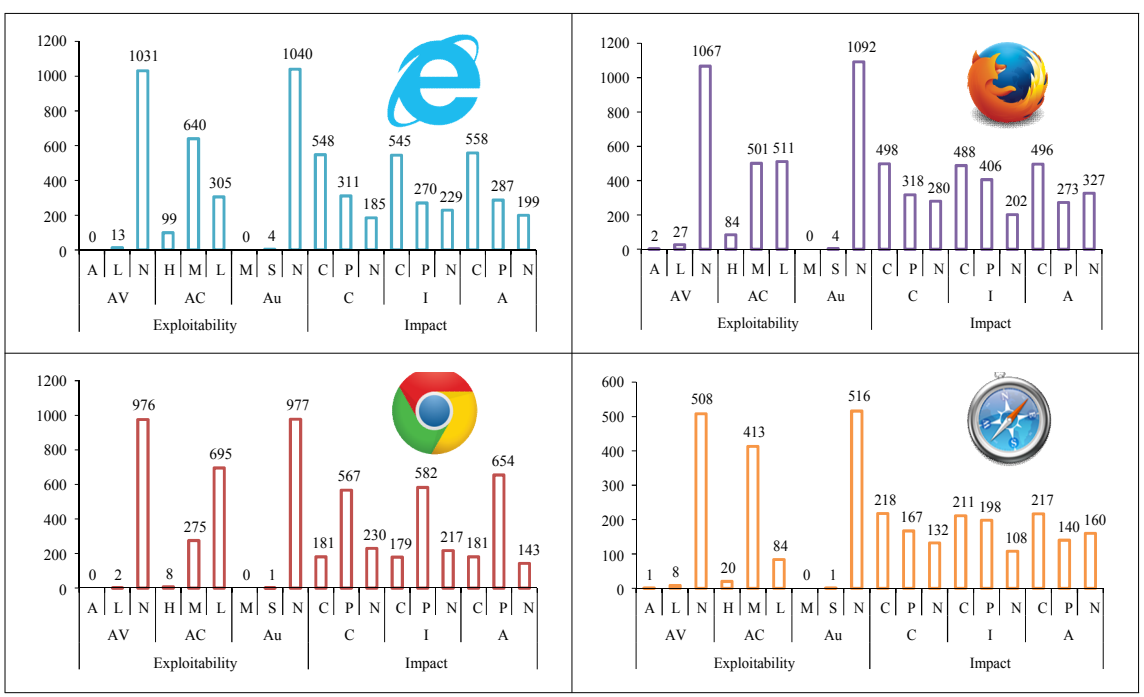

Fig. 3. Number of each value in exploitability and impact sub-score groups.

Table 4. Top three high frequency combinations

	Freq. (%)	Exploitability			Impact		
		AV	AC	Au	C	I	A
IE	498(47)	N	M	N	C	C	C
	120(11)	N	L	N	P	P	P
	72(6)	N	L	N	N	N	P
FX	260(23)	N	L	N	C	C	C
	201(18)	N	M	N	C	C	C
	109(9)	N	M	N	N	P	N
CR	395(40)	N	L	N	P	P	P
	122(12)	N	L	N	N	N	P
	102(10)	N	L	N	C	C	C
SF	193(37)	N	M	N	C	C	C
	82(15)	N	M	N	P	P	P
	67(12)	N	M	N	N	P	N

applicable to the Web browsers. Also, the results show that, almost all the time, vulnerabilities are compromised from remote networks at no authentication required systems. This suggests for organizations to enhance their network security-related facilities, and also to add authentication process in their systems. The result also reveals that exploitation aftermath is getting worse. An analogous study had been conducted by Scarfone and Mell [19] in 2009 based on 11,012 CVEs. They examined CVSS version 2 scoring system in depth without software categorizations. Also, there is a paper [20] dealing with deriving vulnerabilities and implementing tools that can analyze the derived weaknesses.

The methods in this study do not make use of detailed information on software evolutions that may be available. Therefore, further research is needed to evaluate the impact of evolution of software products that go through many versions by explicitly considering the shared code, vulnerabilities inserted and removed in the process and the impact on resource allocation for testing and patch development.

REFERENCE

[1] M. Rajab, L. Ballard, N. Jagpal, P. Mavrommatis, D. Nojiri, N. Provos, et al., *Trends in Circumventing Web-malware Detection*, Technical Report, 2011.

[2] C.P. Pfleeger and S.L. Pfleeger, *Security in Computing*, Prentice Hall PTR, New Jersey, 2003.

[3] O.H. Alhazmi and Y.K. Malaiya, "Application of Vulnerability Discovery Models to Major Operating Systems," *IEEE Transactions on Reliability*, Vol. 57, No. 1, pp. 14-22, 2008.

[4] H. Joh, "Web Browser Secureness with Respect to CVSS," *Proceeding of the 2014 Fall Conference of the Korea Information Processing Society*, Vol. 21, No. 2, pp. 464-465, 2014.

[5] S. Frei, T. Duebendorfer, G. Ollmann, and M. May, *Understanding the Web Browser Threat: Examination of Vulnerable Online Web Browser Populations and the "Insecurity Iceberg", ETH Zurich Tech Report Nr. 288*, 2008.

[6] T. Duebendorfer and S. Frei, "Web Browser Security Update Effectiveness," *Proceeding of the 4th International Conference on Critical Information Infrastructures Security*, pp. 124-137, 2010.

[7] M. Acer and C. Jackson, "Critical Vulnerability in Browser Security Metrics," *Proceeding of Web 2.0 Security and Privacy, IEEE Symposium on Security and Privacy, Oakland,*

CA, USA, May 2010.

[8] A. Grosskurth and M. Godfrey, "A Reference Architecture for Web Browsers," *Proceeding of the 2005 International Conference on Software Maintenance*, Budapest, Hungary, pp. 661-664, Sep. 2005.

[9] G. Schryen, "Is Open Source Security a Myth? What do Vulnerability and Patch Data Say?," *Communications of the Association for Computing Machinery*, Vol. 54, No. 5, pp. 130-140, 2011.

[10] F. Akiyama, "An Example of Software System Debugging," *Proceeding of International Federation for Information Processing Congress*, pp. 353-379, 1971.

[11] B.T. Compton and C. Withrow, "Prediction and Control of ADA Software Defects," *Journal of Systems and Software*, Vol. 12, No. 3, pp. 199-207, 1990.

[12] L. Hatton, "Reexamining the Fault Density Component Size Connection," *IEEE Software*, Vol. 14, No. 2, pp. 89-97, 1997.

[13] J. Rosenberg, "Some Misconceptions About Lines of Code," *Proceeding of the 4th IEEE International Software Metrics Symposium*, pp. 137-142, 1997.

[14] O.H. Alhazmi, Y.K. Malaiya, and I. Ray, "Security Vulnerabilities in Software Systems: A Quantitative Perspective," *Proceeding of IFIP WG11.3 Working Conference on Data and Information Security*, pp. 281-294, 2005.

[15] H. Joh and Y.K. Malaiya, "Modeling Skewness in Vulnerability Discovery," *Quality and Reliability Engineering International*, Vol. 30, No. 8, pp. 1445-1459, 2014.

[16] P. Mell, K. Scarfone, and S. Romanosky, *CVSS: A complete Guide to the Common Vulnerability Scoring System Version 2.0*, Forum of Incident Response and Security Teams, 2007.

[17] S.G. Eick, T.L. Graves, A.F. Karr, J. Marron, and A. Mockus, "Does Code Decay? Assessing the Evidence from Change Management Data," *IEEE Transactions on Software Engineering*, Vol. 27, No. 1, pp. 1-12, 2001.

[18] O.H. Alhazmi and Y.K. Malaiya, "Prediction Capabilities of Vulnerability Discovery Models," *Proceeding of Reliability and Maintainability Symposium*, pp. 86-91, 2006.

[19] K. Scarfone and P. Mell, "An Analysis of CVSS Version 2 Vulnerability Scoring," *Proceeding of 3rd International Symposium on Empirical Software Engineering and Measurement*, pp. 516-525, 2009.

[20] I. Mun and S. Oh, "Design and Implementation of A Weakness Analyzer for Mobile Applications," *Journal of Korea Multimedia Society*, Vol. 14, No. 10, pp. 1335-1347, 2011.

HyunChul Joh

is an assistant professor in department of computer engineering at Kyungil University. From 2012 to 2014, he was a GIST college laboratory instructor in division of liberal arts and sciences at Gwangju Institute of Science and Technology. His research focuses on modeling the discovery process for security vulnerabilities and risk metrics. He received his Ph.D. and M.S. in computer science from Colorado State University in 2011 and 2007 respectively. He also received a B.E. in Information and Communications Engineering from Hankuk University of Foreign Studies in 2005.