Journal of Korea Multimedia Society Vol. 17, No. 12, December 2014(pp. 1494-1502)
http://dx.doi.org/10.9717/kmms.2014.17.12.1494

Quantitative Risk Assessment in Major Smartphone Operating Systems in Asian Countries

HyunChul Joh[†]

ABSTRACT

Since smartphones are utilized in the ranges from personal usages to governmental data exchanges, known but not patched vulnerabilities in smartphone operating systems are considered as major threats to the public. To minimize potential security breaches on smartphones, it is necessary to estimate possible security threats. So far, there have been numerous studies conducted to evaluate the security risks caused by mobile devices qualitatively, but there are few quantitative manners. For a large scale risk evaluation, a qualitative assessment is a never ending task. In this paper, we try to calculate relative risk levels triggered by software vulnerabilities from unsecured smartphone operating systems (Android and iOS) among 51 Asian countries. The proposed method combines widely accepted risk representation in both theory and industrial fields. When policy makers need to make a strategic decision on mobile security related agendas, they might find the presented approach useful.

Key words: Smartphone Operating System, Software Security Vulnerability, Risk Assessment, Quantitative Analysis, Asia

1. INTRODUCTION

Today, smartphones are considered as must-have items for modern citizens. We are using them for many purposes such as web surfing, online banking, remote working and entertaining, not to mention texting and phone calls. Moreover, as the smartphones are getting improved in both hardware and software computational capabilities, it is expected that they will replace significant portions of what the PCs have been done before [1,2]. Smartphone shipments will reach 1,095 million by 2016 [3]. Consequently, smartphone markets are attracting malicious hackers more and more. As a result, growth rate of smartphone malware has been also sharply increased. According to a security threat report [4], already, in some countries like Australia and the U.S., Android threat exposure rates exceed those of PCs. A survey conducted by Portio research center [3] says the subscriber base of cellphone is forecasted to increase at a compounded annual growth rate of 7.3% between 2011 and 2016, to reach nearly 8.5 billion by end of 2016. This growth will be led by markets in Asia Pacific and Africa. Already, Asia Pacific region occupies more than 50% of the market in the subscriber base. Without any doubt, if Asian countries do not pay attention to the possible risks caused by smartphones with great caution, they will definitely suffer more tremendously when they deal with prosperous but hostile smartphone ecosystem in the near future.

Although secure coding habit has been getting improved, computer security experts believe that a complete secure software system sounds not realistic [5]. Therefore, people need to measure how

※ Corresponding Author : HyunChul Joh, Address: (712-701) Gamasilgil 50, Hayangup, Gyeongsan, Gyeongbuk, South Korea, TEL : +82-53-600-5563, FAX : +82-53-600-5579, E-mail : joh@kiu.kr

Receipt date : Oct. 20, 2014, Revision date : Nov. 23, 2014
Approval date : Nov. 24, 2014

[†] Dept. of Computer Eng., Kyungil University

※ This research was supported by the intramural research program in Kyungil University.

secure their smart devices or even organizations are, so that they could take proper actions before it is too late. Just like Lord Calvin (1824~1907) said that, "if you cannot measure it, you cannot improve it," if we want to have a more secure smartphone ecosystem, first we need to measure the risk level in the system somehow.

There have been many studies conducted to improve the security methodologies and defensive mechanisms for mobile devices qualitatively, but there are not many quantitative analyses which utilize the actual datasets to measure risk levels objectively posed by smartphone operating systems (OSes). In many industrial areas, measurements based on quantitative analysis are conducted, especially for the fields having mature datasets [6]. However, risk evaluation in cyber security has not been actively developed quantitatively due to the lack of data for validation. Fortunately, publicly available datasets have been begun to be big enough to be analyzed in smartphone security area recently. A vulnerability is a software defect or weakness in the security system which might be exploited by a malicious user causing loss or harm [7]. In this paper, we attempt to measure smartphone ecosystem risk levels in Asia countries based on software security vulnerabilities in end-user smartphone OSes. The conducted quantitative approach will allow comparison of alternative systems and optimization of risk mitigation strategies for both organizations and individuals.

The paper is organized as follows. Related works are given in Section 2. In Section 3, we will present a quantitative risk evaluation method. Section 4 explains about the datasets we are using in this paper. Demonstration for the described quantitative risk assessment is shown in Section 5. Finally, conclusions are given in Section 6.

2. RELATED WORKS

Theoharidou et al. [8] came up with a method for risk assessment that is specifically tailored for smartphones. The proposed method does not treat a smartphone as a single entity. Instead, it identifies smartphone assets and provides a detailed list of specific applicable threats. They assessed risk as a combination of asset impact and threat likelihood. Similarly, Mylonas et al. [9] examine specific evaluation criteria assessing the security level of the five well-known smartphone platforms. Their analysis examines the feasibility of attacks implemented by average application developers. Specifically, the research is based on the definition of qualitative evaluation criteria and a proof of concept malware implementation study. The authors claim that the iOS has the most defensive mechanism whereas Android and Windows Mobile were found to provide the least protection, according to their study. Meanwhile, a survey paper was conducted by La Polla et al. [10]. This paper provided a comprehensive overview of mobile malware and some predictions on future threats by listing actual examples while the authors examined literatures over the period of 2004-2011. Furthermore, they compared mobile and desktop environment securities, and it shows that differences between the two are often caused by the amount of consumed resources and power.

Jeon et al. [11] analyzed security of smartphone and suggested countermeasures. In their paper, vulnerability means that it can risk security objects potentially, and threat means that it can risk security objects directly. They presented the results of a qualitative risk analysis in order to identify and prioritize the smartphone threats. They suggested new security mechanisms in addition to antivirus and intrusion detection systems, such as system modification, system add-on or add-on applications. Cho et al. [12] analyzed a status of permissions used by 30 banking Android applications worldwide. In Android programming, a permission is a restriction limiting access to a part of code or to data on a smart device. The authors pointed out that many banking applications, which handle extremely sensitive information, unnecessarily overuse dangerous permission group. This coding habit might have a serious impact on the system or other applications, and it implies potential risks. The paper suggested that functions on the banking applications should be limited and developers need to be accordingly guided.

Meanwhile, a quantitative risk analysis has been conducted by Shahzad et al. [13] on popular software systems. In their study, various software vulnerability aspects were examined based on vendors and type of vulnerabilities. A similar large scale analysis was done previously by Frei et al. [14], and this paper is helpful to get background to understand Shahzad et al. [13] in more detail. Joh and Malaiya [15] tried to measure security risk caused by software vulnerabilities. In their paper, they formally define risk measures and examine possible approaches for assessing risk quantitatively. Further, Joh and Malaiya explore the use of CVSS vulnerability metrics to apply them to a general stochastic software risk evaluation.

3. QUANTITATIVE SMARTPHONE RISK ASSESSMENT

Common Vulnerability Scoring System (CVSS) [16] has now become *de facto* industrial standard for assessing the software security vulnerabilities. To prioritize mitigation efforts, CVSS, whose score is a decimal number in the range of [0.0, 10.0], evaluates the degree of risks posed by software vulnerabilities. CVSS score information for the publicly announced vulnerabilities are easily found at most of the major public online vulnerability databases such as NVD (http://nvd.nist.gov) or OSVDB (http://osvdb.org). The score is composed of three major metric groups of Base, Temporal and Environmental. The Base metric represents the intrinsic characteristics of a vulnerability, and is the only mandatory metric. The optional Environmental and Temporal metrics are used to enhance the Base metric score.

The Base metric includes two sub-scores called exploitability and impact sub-scores. The impact sub-score measures how a vulnerability will impact on IT asset in terms of the degree of losses in confidentiality, integrity, and availability. The exploitability sub-score attempts to measure how easy it is to exploit a vulnerability.

Informally, risk is stated as the possibility of something bad is happened [7], and formally, risk is defined to be a weighted measure depending on the consequence. For a potential negative event, the risk can be expressed as [17]:

$$\begin{aligned} Risk &= Likelihood\ of\ an\ adverse\ event \\ &\quad \times Imact\ of\ the\ adverse\ event \end{aligned} \tag{1}$$

The above equation evaluates risk due to a single security hole. If statistically independent multiple causes are considered, the individual risks can be combined together to evaluate the overall risk for a system. In software systems, the likelihood of an adverse event is sometimes represented as the product of two probabilities: probability that a vulnerable flaw is present, and the probability that such a weakness is exploited [18]. The first element is an attribute of the targeted system itself whereas the second probability depends on external factors, such as the motivation of potential attackers.

Fig. 1 shows the idea for the evaluating risk based on the known vulnerabilities. The upper plot simply represents the AML S-shaped vulnerability discovery pattern [19]. In the initial learning phase, the software is gaining market share gradually. In the linear phase, the vulnerability discovery rate reaches the maximum due to the peak popularity of the software, and finally, in the saturation phase, vulnerability discovery rate slows down. The lower plot is a conceptual diagram to illustrate the risk gap between vulnerability discoveries and patch release dates on top of the simplified S-shaped dis-

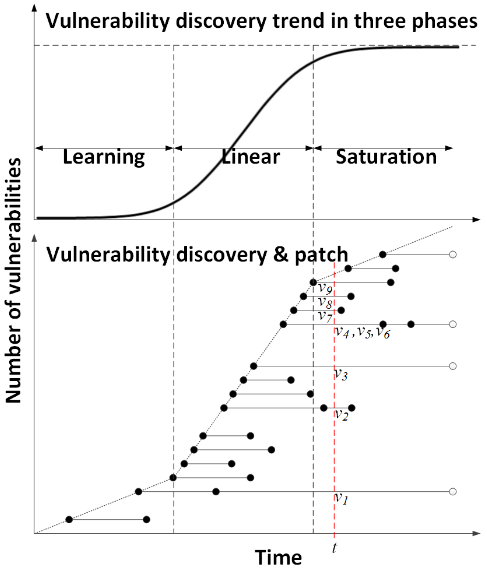


Fig. 1. Vulnerability discovery and patch behaviors on three phase lifecycle.

covery model. We could consider the horizontal line as a risk gap because the vulnerability is known but a patch has not been applied yet.

In the figure, when there are multiple dots at the right side, the solid line represents multiple vulnerabilities discovered at the same time represented by the left most located dot, but with different patch dates. A white dot is used when a patch is not hitherto available. For example, in Fig. 1 at time *t*, marked with the vertical dashed line, there are nine known vulnerabilities (v_1~ v_9) with no patches. To calculate the risk level quantitatively for a given system OS *k* at that time point, risk for every unpatched vulnerability in OS *k* needs to be calculated first, and then added them up as shown in Equation (2), where *ES* is Exploitability sub-score and *IS* is Impact sub-score for an unpatched vulnerability *i* at time *t* in OS *k*.

$$Risk_{OS_k}(t) = \sum_i ES_i(t) \times IS_i(t) \qquad (2)$$

In this paper, for a demonstration purpose, risk evaluation is conducted per each mobile OS. Thus, to calculate the entire risk level posed by smartphone OSes in a given country, we just need to add up the results. For that reason, the risk value at time *t* in country *c*,, based on OS *k*, can be calculated by Equation (3).

$$Risk_c(t) = \sum_k Risk_{OS_k}(t) \times (number\ of\ \ OS_k\ smartphone\ users\ at\ time\ t) \qquad (3)$$

4. DATASET

In this paper, to demonstrate quantitative assessment for risk level posed by smartphone OS, we simply estimate the risk due to the vulnerabilities in popular smartphone OSes, and we speculate the two most popular OSes, Google Android and Apple iOS. Android and iOS occupied more than 95% of the entire smartphone OS market for the second quarter of 2014 (http://www.idc.com/prodserv/smartphone-os-market-share.jsp).

Table 1 shows the number of vulnerabilities discovered in each OS categorized by security severity. The dataset was collected at National Vulnerability Database (NVD; http://nvd.nist.org) on October 5th 2014. NVD could be considered as a standard source because its project is sponsored by US Department of Homeland Security, and maintained by National Institute of Standard and Technology (NIST). Moreover, NVD is synchronized with Common Vulnerability Exposures (CVE; http://cve.mitre.org), so that any updates to CVE appear immediately on NVD. Notice that Android

Table 1. Num. of vulnerabilities and release date

Severity	Num. of Vuln. (2014-10-05)				Release Date	
	High	Medium	Low	Total	Initial	Latest (ver.)
Android	14	22	1	37	2008-09-23	2014-06-19(4.4.4)
iOS	133	209	62	404	2007-06-29	2014-09-25(8.0.2)

represents an open source software system while iOS is a proprietary closed source system. We examine all the vulnerabilities reported at the NVD web site regardless how the defects had arisen.

In Equation (3), is easily achieved according to Fig. 1. However, since no immediate information is available for the *number of OS$_k$ smartphone users at time t*, we reasonably construct the dataset. To approximate the number of smartphone users in each country, we utilize the number of mobile phone subscribers (http://www.itu.int on OCT 5th 2014), the two OSes' market share in each country (http://gs.statcounter.com on OCT 5th 2014) and smartphone user percentage [20]. At this point, to achieve the *number of OS$_k$ smartphone users at time t*, we just need to calculate {*the number of subscriber smartphone user percentage market share for OS$_k$*}. We assume that the number of smartphone users are remained the same during the examined period.

5. RISK FROM KNOWN UNPATCHED VULNERABILITIES

Since significant effort and time will be required to gather the actual patch release dates [21], patch dates are simulated, based on the aggregate data sources [13]. Table 2 shows patching behaviors for Google and Apple, which represents the fraction of vulnerabilities patched on average, before date (< 0), on the date (0), within 7 days (+7), within 30 days (+30), and after 30 days (>30), with respect to the vulnerability disclosure date. Disclosure time is when a vulnerability is disclosed to the public, and patch date means when a patch is available to the public. Datasets falling into >30 have high chances to be patched later, but due to the lack of information, the simulated data treats them as unpatched vulnerabilities which causes the simulated data to differ from the real patch information. Table 3 shows the simulated patch required time. This simulation is conducted to only illustrate the procedure and not for evaluating the real risk levels of the actual devices.

Fig. 2 (a) and (b) represent the risk gaps for the two OSes. They are realization of Fig. 1. In Fig. 2 (a), the first vulnerability was reported on May 26th 2009, and it is about a month later, Cupcake (Android 1.5) had been released. Since then, Google has been released 19 major releases. As a result, the vulnerabilities have been reported continually and regularly. For Android, only one vulnerability has not been patched by end of the examined period due to the high patch rate with relatively small number of known vulnerabilities. Meanwhile, in Fig. 2 (b), the vulnerability discovery trend in iOS raises steeper than its counterpart. Unlike Android, iOS has 24 simulated unpatched vulnerabilities. Based on the past and current trend, newly discovered vulnerabilities will be continually reported in both systems.

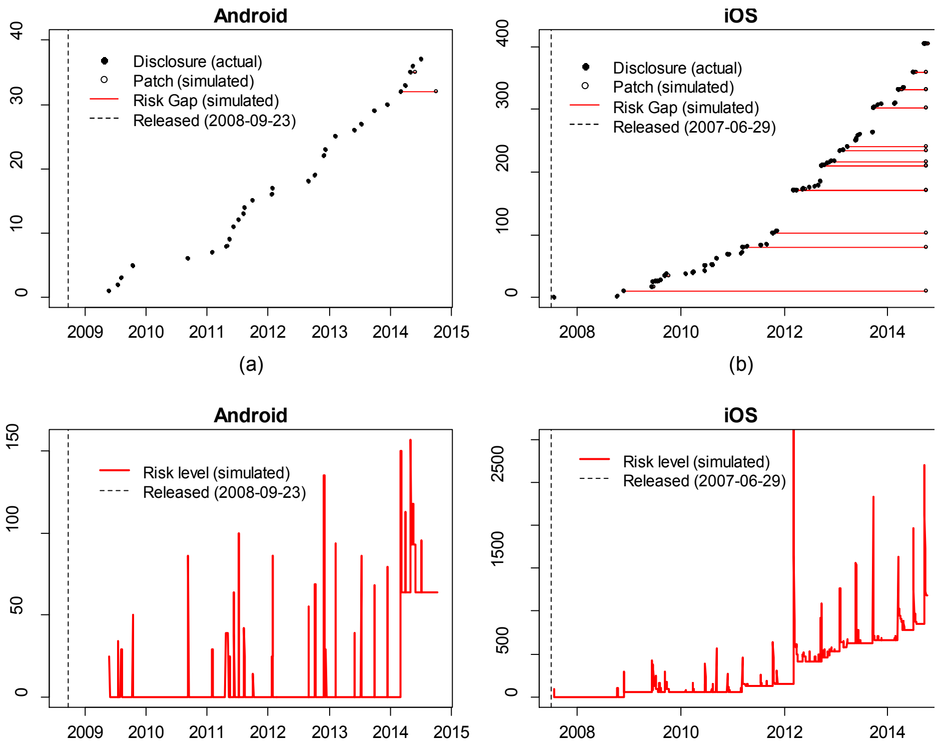
Fig. 2 (c) and (d) give the simulated risk levels. Risk level for Android is sporadically appear whenever a vulnerability is reported, and gone shortly after because most of the vulnerabilities are patched on the disclosure date as shown in Table 3. On the other hands, risk level for iOS has been steadily increased. The long term rising trend observed is caused by vulnerabilities that we have presumed to be unpatched after 30 days from the disclosure date. On March 8th 2012, the risk value hit the peak value (5074.53) due to the overlapped unpatched vulnerabilities on that time. Indeed, we can see a significant number of vulnerabilities are

Table 2. Average patch time relative to disclosure dates [13]

Vendor	< 0	0	+7	+30	>30
Google	2%	94%	1%	2%	1%
Apple	5%	78%	5%	6%	6%

Table 3. Simulated patch required time relative to disclosure dates (Num. of vul.)

OS	< 0	0	+7	+30	>30
Android	1	32	1	2	1
iOS	20	315	20	25	24



(a)

(b)

Fig. 2. Evaluated risk gaps (a, b) and risk level (c, d).

disclosed at the beginning of year 2012 in Fig. 2 (b). Since the patch dates are simulated, the results only serve as an illustration of the approach and do not represent any actual products.

Fig. 3 shows estimated risk levels in 51 Asian countries according to Equation (3). Due to the scale problem, countries are grouped accordingly: Group A has the five countries having the five highest peak values, Group B has the five countries having the next five highest peak values, etc. Notice that Group J contains six countries. In the plots, numbers in parentheses represent the peak values.

Not surprisingly, China places the first place due to the huge mobile phone subscribers with a high rate of smartphone users. Japan and Vietnam, where iOS is more popular than Android, hit the second and the third peak values respectively. With fairly large number of mobile phone subscribers and relatively high iOS usage, compared to other Asian countries, Thailand and Philippines mark the fourth and the fifth. Further, correlation coefficients are calculated between the peak risk values and the number of users discussed in Section 4. As shown in Table 4, the number of smart phone users has relatively strong positive correlations with the peak values. Naturally, the correlation coefficients for the number of Android and iOS users with the peak risk values have also strong positive relation. Due to the higher unpatched vulnerability rate, iOS has more influence on the risk than Android's one. Since the analysis depends on the simulation, the output should not represent risk level on the actual products.

Table 4. Correlation coefficients between the peak risk values (Fig. 3) and number of users

Smart Phone	Android	iOS
0.6332	0.7358	0.9998

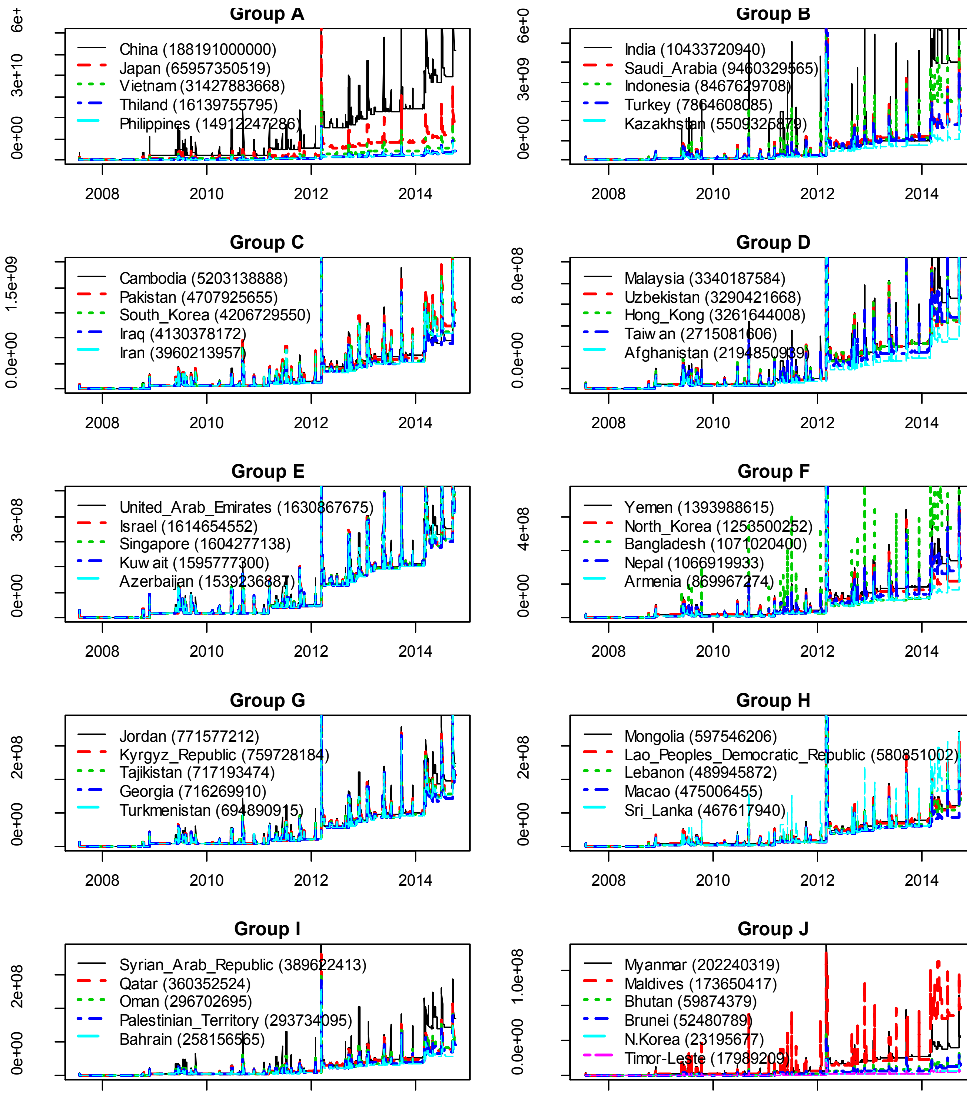


Fig. 3. Estimated risk levels in Asian countries posed by vulnerabilities in smartphone OSes.

6. CONCLUSION

We present a quantitative risk assessment due to known but unpatched vulnerabilities posed by smartphone OSes, using actual datasets. In this paper, a formal quantitative approach for risk evaluation is shown which uses a simplified vulnerability lifecycle and the CVSS metrics. The presented approach incorporates vulnerability discovery and simulated patch date information. The risk values for individual vulnerabilities can be combined to evaluate risk for an entire software system in a smart mobile device, which can be used for evaluating the risk for an entire organization, or even for a nationwide assessment.

There are several threats to validity in this study. Because we used the simulated patch date information, the results from the paper might not be fully appreciated. Also, we assume that the number of smartphone users and the market share of the two mobile OSes are remained the same during the examined time period. Furthermore, the smartphone user ratio from the white paper [20] is an estimation (Average value of Asia Pacific and Middle East). And, this paper only considers unpatched vulnerabilities to derive risk levels. Other factors which might influence on mobile security are not reflected such as mobile application malware. There is a paper [22] dealing with deriving vulnerabilities specialized in mobile applications and implementing tools that can analyze the derived weaknesses. And last but not least, we need to set up what a single unit risk is supposed to signify.

One of the main contributions of the paper is raising awareness about the security vulnerabilities in smartphone, especially in Asian countries. Also, the paper shows a possibility that risk posed by smartphone could be estimated quantitatively whose output is objective and clear than qualitative analysis, so that it is more appropriate to be used by policy makers. This study demonstrates a nationwide large scale of risk assessment which has been considered a never ending task with a qualitative risk analysis.

The demonstrated risk assessment provides a systematic approach for risk evaluation posed by any software vulnerabilities. It can be used for comparing the risk levels for alternative systems, and the approach can be incorporated into a methodology for allocating resources optimally by policy makers when information security matters.

REFERENCES

[1] T.M. Chen, “30th Anniversary of the PC and the Post-PC Era [Editor's Note],” *IEEE Network*, Vol. 25, No. 5, pp. 2-3, 2011.

[2] T. Bajarin, Why Your Smartphone Will Be Your Next PC, http://techland.time.com/2013/02/25/why-your-smartphone-will-be-your-next-pc/ (accessed Jul., 7, 2014).

[3] Portio Research Limited, Mobile Factbook, Portio Research Limited, http://www. portioresearch.com/media/3986/Portio%20Research%20Mobile%20FactbooF%202013.pdf (accessed Jul., 7, 2014).

[4] Sophos Ltd. Security Threat Report 2013, http://www.sophos.com/en-us/medialibrary/PDFs/other/sophossecuritythreatreport2013.pdf (accessed Jul., 7, 2014).

[5] S. Farrell, “Why Didn't We Spot That?,” *IEEE Internet Computing*, Vol. 14, No. 1, pp. 84-87, 2010.

[6] H. Joh, *Quantitative Analyses of Software Vulnerabilities*, Doctor's Thesis of Colorado State University, 2011.

[7] C.P. Pfleeger and S.L. Pfleeger, *Security in Computing*, 3rd ed. Prentice Hall PTR, Saddle River, New Jersey, 2003.

[8] M. Theoharidou, A. Mylonas, and D. Gritzalis. “A Risk Assessment Method for Smartphones,” *Proceedings of the 27th Information Security and Research, IFIP Advances in Information and Communication Technology,* pp. 428-440, 2012.

[9] A. Mylonas, S. Dritsas, B. Tsoumas, and D. Gritzalis, “Smartphone Security Evaluation The Malware Attack Case,” *Proceedings of the 2011 International Conference on Security and Cryptography,* pp. 25-36, 2011.

[10] M.L. Polla, F. Martinelli, and D. Sgandurra, “A Survey on Security for Mobile Devices,” *IEEE Communications Surveys & Tutorials*, Vol. 15, No. 1, pp. 446-471, 2013.

[11] W. Jeon, J. Kim, Y. Lee, and D. Won, “A Practical Analysis of Smartphone Security,” *Proceedings of the 2011 International Conference on Human Interface and the Management of Information*, pp. 311-320, 2011.

[12] T. Cho, Y. Kim, S. Han, and S. Seo, “Potential Vulnerability Analysis of Mobile Banking Applications,” *Proceedings of the 2013 International Conference on ICT Convertgence*, pp. 1114-1115, 2013.

[13] M. Shahzad, M.Z. Shafiq, and A.X. Liu, “A

Large Scale Exploratory Analysis of Software Vulnerability Life Cycles," *Proceedings of the 34th International Conference on Software Engineering*, pp. 771-781, 2012.

[14] S. Frei, M. May, U. Fiedler, and B. Plattner, "Large-Scale Vulnerability Analysis," *Proceedings of the 2006 SIGCOMM Workshop on Large-scale Attack Defense*, pp. 131-138, 2006.

[15] H. Joh and Y.K. Malaiya, "Defining and Assessing Quantitative Security Risk Measures Using Vulnerability Lifecycle and Cvss Metrics," *Proceedings of the 2011 International Conference on Security and Management*, pp. 10-16, 2011.

[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] National Institute of Standards and Technology, *Risk management guide for information technology systems*, Special Publication 800-30, 2001.

[18] L.A. Cox, "Some Limitations of "Risk = Threat x Vulnerability x Consequence" for Risk Analysis of Terrorist Attacks," *Risk Analysis*, Vol. 28, No. 6, pp. 1749-1761, 2008.

[19] 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.

[20] Cisco Systems Inc., *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2013-2018*. White Paper, 2014.

[21] A. Arora, R. Krishnan, R. Telang, and Y. Yang, "An Empirical Analysis of Software Vendors' Patch Release Behavior: Impact of Vulnerability Disclosure," *Information Systems Research,* Vol. 21, No. 1, pp. 115-132, 2010.

[22] 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.