Towards An Analysis of Software Supply Chain Risk Management

Shixian Du, Tianbo Lu, Lingling Zhao, Bing Xu, Xiaobo Guo, Hongyu Yang

***Abstract*—Nowadays, software supply chain participants have become international distributors, which make software supply chain more and more complex. This complexity makes manager understand, acquire, monitor and manage software supply chain products and processes more difficult than ever, and then relevant security problems happen, such as software with security holes. But most of security problems are different from other supply chains. Therefore, based on system's perspective and current several analysis methods of software supply chain, the paper analyzes and summarizes software supply chain risks, software supply chain risk management methods, and puts forward some basic risk management practices to protect software supply chain's security. Finally, the paper discusses the future research direction to software supply chain.**

***Index Terms*—software supply chain, software supply chain risk, risk management, software security**

I. INTRODUCTION

ICT (Information and Communication Technology) is the lifeblood of modern civilization, and is indispensable to organizations and individuals, who rely on ICT to support the implementation of key activities and tasks. ICT supply chain mainly includes hardware supply chain and software supply chain. Now, every country has paid great attention to strengthen the security of hardware supply chain, especially critical infrastructure. Since software supply chain is as important as hardware supply chain and every supply chain can't be divorced from software, there should be given more attention to software supply chain.

Manuscript received July 2, 2013. This work is supported by the following programs: the National Natural Science Foundation of China under Grant No.61170273; 2010 Information Security Program of China National Development and Reform Commission with the title "Testing Usability and Security of Network Service Software"; Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (NO. CAAC-ITRB-201201).

Shixian Du is with the School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876 China (e-mail: shixian2011@126.com).

Tianbo Lu is with the School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876 China (e-mail: lutb@bupt.edu.cn).

Lingling Zhao is with the School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876 China.

Bing Xu is with the School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876 China (e-mail: xub@bupt.edu.cn).

Xiaobo Guo is with the School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876 China.

Hongyu Yang is with the Information Technology Research Base of Civil Aviation Administration of China, Civil Aviation University of China, Tianjin, 300300 China.

With the increase of globalization, outsource [1] and numerous sources of risk, software supply chain is relative fragility, for example, there are hundreds of viruses intruding into the supply chain. Moreover, supply chain risk management has been the topic of the latest CROSSTALK magazine (http://www.crosstalkonline.org/). Obviously, it's urgent to effectively manage the risk of software supply chain to protect its security.

For thorough understanding software supply chain's security, we should first know what is software supply chain, and what's the difference from hardware supply chain. Then we talk about the risks which software supply chain is facing and some risk management practices. At last, some concrete measures are proposed to improve software supply chain risk management.

II. DEFINITION OF SOFTWARE SUPPLY CHAIN

In general, supply chain aims at manufacturing and transmitting entity, while there are some supply chains which are relevant to software, namely, software supply chain. Currently, there are several major software supply chain's definitions, as follows:

(1) Software supply chain appears to be different from supply chain. In 2001, Lynne F. Baxter and John E. L. Simmons put forward the viewpoint about software supply chain, which only involved supplier and customer, and there was a dyadic relationship where software was purchased, between an autoteller manufacturer and its customers [2].

(2) A software-focused supply chain is a supply chain where software constitutes a significant part of the total value of the product, and "goods" may not be physically flowing through the supply chain [3].

(3) Software supply chain is composed of component vendors, application vendors, users and assemblers which are based on the method of distributed software development. Besides, its service management includes deploy, operate, optimize and retirement processes [4].

(4) Software supply chain consists of a series of related software, hardware and service, including software maintenance, release, and deployment processes [5].

(5) The Carnegie Mellon Software Engineering Institute (SEI) considers software supply chain is the network of stakeholders who develop the contents of software product or modify the contents [6].

(6) Software supply chain includes supply chains for physical components, integrated components and software. In detail, the supply chain of commercial software products includes product development organizations and their

ISBN: 978-988-19252-3-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

suppliers. The supply chain of custom software systems includes main contractors, sub-contractors, and supply chains for the commercial products' use [7].

Therefore, software supply chain can be divided into two systems: software supply chain is simply the process of developing and assembling software products. Another system is that software supply chain is the whole development, release, deployment, and maintenance processes of software from source code to the final software delivering to users. The participants in software supply chain include: the purchaser in the industry and government, information security manager supporting purchase, software suppliers, service suppliers, contractors, distributors, retailers and end-users. The later system is more precise and comprehensive than the former system.

III. COMMONNESS AND INDIVIDUALITY OF SOFTWARE SUPPLY CHAIN

Software is a special commodity which is different from hardware, and plays a vital role on system. By comparing general description of hardware and software supply chain, we can see that there are common characteristics between them, but they still have their own individuality. In other words, they both cross each other, but are different between them.

A. Commonness

Software supply chain is a kind of supply chain. As same as other supply chains, it is a business process model and a value chain made of components suppliers, manufacturers, distributors, retailers and end-users. It completes the process starting from customer demand to providing the customer's required products and services. From the view of inner organization, software supply chain includes procurement, developing, distribution and other departments. From the view of external organization, software supply chain includes suppliers, developers, vendors, and end-users.

On the other hand, software supply chain is also closely related to whether supplier is reliable, whether suppliers deliver the goods on time, and whether the goods can be conformance to specification. Take an example, some software products' transmission mode is the same as hardware supply chains', such as anti-virus software by CD burning. During the transmission, some threats may occur, such as theft, hijacking, accidental burning and so on.

B. Individuality

Software product is different from other entities. As electronic products, they can be transmitted over the network to send to customers. In this transfer process, they will encounter new risks that other hardware products don't encounter. For example, as a result of the customer's temporary leave, other people using customer's computer accept the file, so that there are the dispute between the parties, that is the sender has sent, but the recipient has not received. Moreover, there are many threats on the Internet, such as denial of service attacks and IP spoofing. Denial of service attacks can lead suppliers' system sudden failures, so they will be unable to send software to the customer in time, leading to delivery delays. And malware software can intercept software package sent by the provider during the transmission, and change the right IP address to other IP address, so that the customer don't receive the package.

As there are different from the characteristics of the traditional supply chain, it's necessary to put forward new risk management approaches to protect software supply chain's security. Next, we should know what risk software supply chain face.

IV. SOFTWARE SUPPLY CHAIN RISK

Software supply chain connects software suppliers, service suppliers, contractors, distributors, retailers and end-users. The structure is so complex that potential risks exist in every step. If there is one problem in any step, it will cause serious influence on the whole chain, and cause directly or indirectly significant economic losses to the relevant participants. As software supply chain risk has some similarities to hardware supply chain, we can analyze the software supply chain risk based on antecedent SCRM.

Software supply chain risk is mutual interdependence, because each process of supply chain isn't alone, and every organization has its risk, while its risk isn't segregated from other organization of software supply chain. So the stakeholder of software supply chain should evaluate and manage risk under a broader background of the system. As the Fig.1 shows, several sources can trigger the risk of software supply chain [6, 8-11].

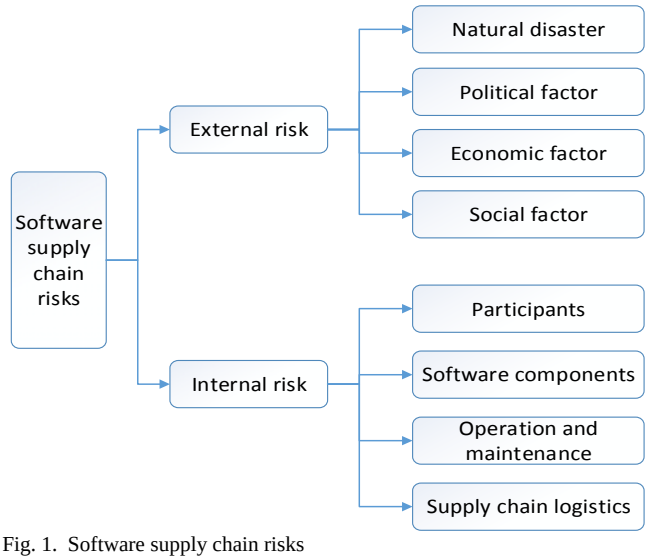

Fig. 1. Software supply chain risks

A. External risk

Software supply chain risk can divide into two large classes, external risk and internal risk. External risk can't be excluded, and it comes mainly from the unexpected emergencies of external environment, which affect one or more members of software supply chain to cause supply chain disruptions. It includes natural disaster (earthquake, fire, flood, etc.), political factor (such as, war and terrorist attack), economic factor (such as, financial risk), and social factor (such as, national laws). For external risk, participants in the software supply chain can't avoid this risk, and it's

ISBN: 978-988-19252-3-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

more difficult to predict when it will happen.

Internal risk comes from the uncertainty of various participants and operational processes. As the Fig. 1 shows, it includes participants, software components, operation and maintenance, and supply chain logistics.

B. Participants

When it comes to internal risk, we firstly should not believe all participants. As long as suppliers have access to final software products, there is software supply chain risk. These suppliers include distributors, transporters and the organizations of developing, enhancing or changing system content. If we may not reduce participants' risks, the risk inherited from each layer of supply chain can turn into security crisis.

Purchaser

After purchaser ensuring the demand of software, purchasing will relate to many suppliers, while the selection will be open to question.

For example, as usual, it's limited to get the knowledge of the safety of COTS product for purchaser [12]. Usually if the purchaser is disposed to purchase COTS with a great bug from unreliable supplier, the COTS can be identified as high risk components. In software supply chain, high risk COTS product connected with other software product and system can develop a software system or system of system. Therefore, unreliable purchaser will bring a series of system's risks to software supply chain.

Supplier

Malicious supplier can tamper the functions of software, for example, they can deliberately insert backdoor on the system so that they can freely access the system in future.

More importantly, supplier designs software which may have some software vulnerabilities [13], such as MITRE's Common Weakness Enumeration (CWE, http://cwe.mitre.org/). Unintentional available design or code can trigger some risks. Sometimes the developer writes some codes of empty functions to use it in future, but never use it. In addition, some defects can make unauthorized organization change software product or system, and then decrease security attribute. Besides, when suppliers discard the software product or system, improper processing information can potentially bring risk of future software product.

C. Software components

Software supply chain risks are all software components' risk in the system. A vulnerability of software components can bring a vulnerability of software supply chain and more. Moreover, some components' vulnerabilities tend to be ignored, and then change the software's functions so that risks are brought to software supply chain.

Reusable components

Software reuse is the process which creates software systems from predefined software components (that is, reusable components). Dependencies between software components may make defective reusable components rapid infiltrate and do great damage to software supply chain. For example, single defective reusable component can cause the problem of security vulnerabilities, data leaks, system stability and performance.

Open source software

Open source software can bring risks to software supply chain [20, 25]. As open source, software supply chain becomes more complex. Everyone can get the source code of software, while the malicious user will research the source code to capture the vulnerability of the software, and then attack the software by using the vulnerability.

Additional software component

Current software has many features, but some functions are redundant for professional software which aims to achieve particular function. As software add so many features, software developers must write multiple lines of code to achieve these corresponding functions, while these codes' security is open to question. For example, software developers use the strcpy function in his code, but the function is insecure and prone to trigger buffer overflow. Under normal circumstances, these problems will cause a vulnerability of the software which creates the conditions for the attacker of software supply chain.

Business component

A business component can be used for one year or longer. During use of business component, the defects of the component may be found and new attacks of technology may come, malicious user could attack software supply chain with business component. At the same time, the upgrade of component makes the invalid of risk assessment; besides, especially some temporary business products, it tends to have more risks. Therefore, more and more risks are introduced to the software supply chain though business component.

D. Operation and maintenance

As time goes on, the use and deployment of software and the change of service can trigger operational problems. Unsafe deployment configuration (such as lack of certification) can make software vulnerable. And in some cases, service change can make some function unavailable, so that it interrupts software supply chain.

Moreover, software usually provides more functions than user actually need. In many cases, many unused functions and services always work, while user don't know that, so that it can lead to security vulnerabilities, and increase risk of operation in the case of user's ignorant operation. It's necessary to make all unessential functions unavailable to reduce the probability of software attacks.

E. Supply chain logistics

When software product or system transfer from one organization to other organization, improper access control of product and service (such as the failure of components delivery and configuration control) can make unauthorized participant tamper with the product at each stage of supply chain.

Besides, software distribution often uses the Internet, rather than physical medium. In other words, software supply

ISBN: 978-988-19252-3-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

chain can't be divorced from Internet. An open loophole in the network may make this software supply chain have vulnerability, and then may cause unexpected consequences. The more complex modern network environment is, the more risk of software supply chain there is.

Considering there are many risks in the software supply chain, we need to increase the overall supply chain's ability of resistance and avoid all participants of software supply chain suffering huge losses, effectively managing its risk has important practical and theoretical significance to ensure software supply chain's security. In the following, the paper summarizes the methods of risk management for software supply chain.

V. SOFTWARE SUPPLY CHAIN RISK MANAGEMENT

Software supply chain risk management is the process of identifying, measuring software project risk, laying down, selecting and managing the processing schemes of software supply chain risk. The process is dynamic, periodic, systematic and integrated. Software supply chain risk management aims at strengthening the understanding of the security risks to software system and its components, mitigating software supply chain risks, minimizing the damage of affected software, and ensuring low quality or counterfeit software products do not enter software supply chain.

Moreover, software supply chain risk management is both rich and complex. It mainly includes four aspects: risk identification, risk assessment, risk treatment and risk monitoring and control [14, 15, 19].

A. Risk identification

Risk identification is the first step in risk management, and also is the foundation of risk management. Risk identification is to identify the main risks faced by the participants in the software supply chain.

Current software supply chain risk identification method mainly includes attack surface analysis [10, 16] and systemic analysis [6]. Besides, we can use the method of flow chart which learns from the antecedent SCRM. Flow chart clearly shows the entire supply chain workflow, and we can do in-depth analysis on each link to identify risks fully.

B. Risk assessment

Once risks have been identified, we must assess their severity of potential impact (generally damage or loss) and the probability of occurrence. Modern software supply chain risk assessment methods are threat modeling [11, 17-18] and systemic assessment.

In fact, it's more effective to make assessments at the purchase of software components. To make assessments remain valid, we should regularly assess the risk of software supply chain.

C. Risk treatment

Once risks have been identified and assessed, all techniques to manage software supply chain risks, that is risk treatment, fall into one or more of these major categories, namely, risk avoidance, risk sharing and risk prevention.

Risk avoidance

Risk avoidance reduces risk damage in the way of changing plans, voluntarily giving up or refusing to take risks. Although the damage can be avoided, risk avoidance means losing the benefit which the risk brings. It is mainly applied to the two situations, one situation is risk can't be prevented and controlled. When there is insurmountable risk that some step of software supply chain encounter, we should remove risk from small risk areas to avoid the risk of a head-on collision, and then enter into the area of larger risks until the ability of risk-resisting enhanced. Another situation is designing supply chain structure to avoid software supply chain risks. The second one is more powerful than the former.

Risk sharing

Software supply chain participants take a certain amount of risk together, namely, risk sharing. It aims to reduce the risk which single participant takes and help avoid relationship risks. Software supply chain achieves this goal, which asks for each participant to achieve risk sharing and benefit sharing. There are two ways of risk sharing in software supply chain, as follows.

(1) Risk allocation on the ability to afford risk. Each link in software supply chain has different risk tolerance, so it should make their developing risk not exceed its affordability, so as to maintain the relative stability of participants' cooperation.

(2) Risk allocation on the ability to control risk. The participant who has bigger risk control capability takes risk, that is, the participant should bear the risk when he deals with risk to obtain the maximum benefit.

Risk prevention

Risk prevention refers to take preventive measures to reduce the likelihood of damage, which is in all links of software supply chain. Risk prevention needs to establish a sound system of risk measurement and management structure to make organizational structure contribute to risk management. It includes defining software supply chain risk sources, making organizer's responsibilities definite, establishing effective risk control system and a series of effective internal control.

Besides, software supply chain needs to establish a close cooperating relationship between the links of software supply chain, which is prerequisite for successful risk prevention.

D. Risk monitoring and control

Risk monitoring and control aims at closely monitoring those risk factors with high risk levels. Each participant of software supply chain should pay attention to risk monitoring and control. One effective method of risk monitoring and control is establishing risk warning system covering a variety of risk indicators, which relies on the implementation of risk assessment, so that software supply chain remains in a safe operating condition. The risk warning system compares the value of risk assessment with the warning value through the establishment of cordon, understand current understanding of the risk profile of the supply chain, and then predict potential

ISBN: 978-988-19252-3-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

risks.

VI. Software Supply Chain Risk Management Best Practices

Based on the above software supply chain risk management methods, the following we put forward some best practices to improve risk management.

A. Design good software supply chain

Good software supply chain's structure increases software supply chain stability and anti-attack, and help achieve risk avoidance. In software supply chain's structure, any subtle changes of link may bring about the change of other links. During the process of improving software supply chain structure, we can take the following aspects into consideration.

Establish an efficient information delivery channels

We can use modern communication and information means to manage and optimize the entire software supply chain, and achieve supply chain participants interconnect and information sharing via the Internet.

Enhance the transparency of information

Transparency of information is conducive to enhancing trust between the link cooperative participants, effectively inhibit software supply chain's relationship risk and improve the response speed of software supply chain.

In general, if information sharing between upstream and downstream of participant is more adequate, feedback mechanism is more timely and the processing flow is more standardized, so that the risk of software supply chain is smaller, on the contrary, the greater. At the same time, most of software supply chain risk management emphasize on strengthening public-private partnerships [24]. In other words, the coordination and cooperation between links should be emphasized in the software supply chain's structure. Such coordination and cooperation relations are based on mutual common goal, mutual trust, and free exchange of information and knowledge shared innovations.

Simplify software supply chain

The complexity of software supply chain is an important source of the uncertainty, the difficulty of coordinate to the whole software supply chain becomes non-linear growth with the link increasing. Therefore, in a way, software supply chain is more simplified, the security of software supply chain is more high. We can eliminate redundant links to simply software supply chain. Simplifying software supply chain also means that the link participants should restructure the organization process according to the process of software supply chain, such as taking cross-functional balanced manage the procurement, development and so on.

Increase software supply chain flexibility

In the software supply chain, the uncertainty of demand and supply is an objective fact, flexible design is an important means to increase software supply chain flexibility and eliminate the uncertainty caused by the external environment variables. Flexible means can be divided into:

1) Maintain an appropriate number of suppliers [26]. Software supply chain risk management emphasis is to ensure high-quality and stable software supply, and a certain number of suppliers is conducive to the dynamic adjustment of the supply chain and to improve the security.
2) Production process flexibility. In software supply chain, suppliers and customers are relative concepts. That is, customers in one link of software supply chain can be regarded as suppliers of the next link [22]. Considering the complex relationship, designing supply chain structure should be flexible to quicken the relationship transition.

B. Manage open source component

As same as other supply chain, software supply chain not only focus on strengthening supply chain security from a global perspective, but also strengthen the quality to ensure security from the local perspective, that is software itself (for example, minimizing software supply chain risks based on software assurance). In other words, it's necessary to keep watch on software code and examine the origin of open software components and processes used to develop it to manage software supply chain risk.

View of most installations are open source components, the life cycle of open component can divide into four steps. We can firstly collect existing components' information, analyze the vulnerability of the application, then establish control through the development life cycle, and finally manage the component based on more in-depth understanding of the components of the software [27].

C. Secure software development methods

Secure software development methods are important to software supply chain risk management. On one hand, we can refer to the SAFECode's fundamental practices for secure software development. On the other hand, there are five secure development methods we can use in the development of software, that is, S3R (Security、Safety、Reliability 、Survivability), SDL [23], CLASP (Comprehensive, Lightweight Application Security Process) [28], Seven Touchpoints [29] and SAMM (Software Assurance Maturity Model) [21]. During the development life cycle, developers should avoid using vulnerable code. Moreover, it's also essential to regularly evaluate the software vendors' secure development methods. So it can help the supplier find where they is easy to ignore, and where developing steps should be kept.

In addition, we can effectively improve software development environment by means of high-tech management and service.

D. Software testing

Examining the software product directly by software testing is one of software supply chain risk management approaches. For example, static analysis technology examines source code and binaries for vulnerabilities. Above all, binary analysis is a highly effective weapon against

ISBN: 978-988-19252-3-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

threats. It avoids trusting the development tools and process.

As time goes on, new software testing technology may come. We can continue to use software testing to find bugs and remove security vulnerabilities, and then improve software supply chain's security.

VII. Conclusion

Software will always be central of system. The security of software supply chain will become more sensitive. The paper mainly analyzes software supply chain risks and how to manage the risks. We believe as long as we deeply research software supply chain and its risk, more effective measures will be taken to protect the security of the software supply chain.

However, there is less data of software supply chain than hardware supply chain nowadays. Considering many modern analyses are based on large statistical data, less data means it's less sufficient to analyze the risk of software supply chain. So software supply chain's security is worth further research, and we believe the future research will be concentrate on the following aspects to enhance its security. Firstly, given that software supply chain distribution channels always get through the Internet, it's worth researching on enhancing security from cyberspace to ensure its safe passage. Secondly, new software development ways will be in-depth study. Secure code is a necessary but not sufficient condition for ensuring the security of software supply chain. Scientific development ways are more important than secure code. Thirdly, based on learning from other standards, researchers can put forward new standard which is specially suitable for software supply chain. Open software supply chain standard can get the widest range of technical and service support, contribute to scientifically manage software supply chain, and then enhance the security of software supply chain.

References

[1] James Andrew Lewis, "Foreign influence on Software risks and recourse", Center for Strategic and International Studies, 2007.
[2] Lynne F. Baxter and John E. L. Simmons, The Software Supply Chain for Manufactured Products: Reassessing Partnership Sourcing, Portland International Conference on Engineering Management and Technology, 2001.
[3] Mabel C. Chou and A. Ruchika, "An In-depth Study of the Software Supply Chain", 2006 IEEE International Conference on Industrial Informatics, August 2006, pp. 753–758.
[4] Roy Oberhauser and Rainer Schmidt, Improving the Integration of the Software Supply Chain via the Semantic Web, International Conference on Software Engineering Advances, August 2007, pp.79.
[5] Slinger Jansen, Sjaak Brinkkemper, Gerco Ballintijn and Arco van Nieuwland, Integrated Development and Maintenance of Software Products to Support Efficient Updating of Customer Configurations: A Case Study in Mass Market ERP Software, Journal of Software Maintenance and Evolution: Research and Practice, Vol. 18, No. 2, 2006, pp. 133–151.
[6] Christopher J. Alberts, Audrey J. Dorofee, Rita Creel, Robert J. Ellison and Carol Woody, A Systemic Approach for Assessing Software Supply-Chain Risk, 2011 44th Hawaii International Conference on System Sciences, January 2011, pp. 1–8.
[7] Robert J. Ellison, Christopher Alberts, Rita Creel, Audrey Dorofee and Carol Woody, Software Supply Chain Risk Management: From Products to Systems of Systems, December 2010.
[8] Iván Arce and Elias Levy, Poisoning the Software Supply Chain, IEEE Security & Privacy, vol. 1, May-June 2003, pp.70–73.
[9] Robert J. Ellison, Christopher Alberts, Rita Creel, Audrey Dorofee, and Carol Woody, Software Supply Chain Risk Management: From Products to Systems of Systems, 2010.
[10] Robert J. Ellison, John B. Goodenough, Charles B. Weinstock and Carol Woody, "Evaluating and Mitigating Software Supply Chain Security Risks", 2010.
[11] Dr. Robert J. Ellison and Dr. Carol Woody, "Considering Software Supply Chain Risks©", 2010.
[12] Defense Science Board, Report of the Defense Science Board Task Force on Mission Impact of Foreign Influence on DoD Software, 2007. Available: http://www.cyber.st.dhs.gov/docs/Defense%20Science%20Board%20Task%20Force%20-%20Report%20on%20Mission%20Impact%20of%20Foreign%20Influence%20on%20DoD%20Software%20%282007%29.pdf
[13] Dr. Robin A. Gandhi, Dr. Harvey Siy, and Yan Wu, Studying Software Vulnerabilities, CROSSTALK, Vol.23, No.5, 2010, pp.16–20.
[14] S. Nurmaya Musa, Supply Chain Risk Management: Identification, Evaluation and Mitigation Techniques, 2012. Available: http://liu.diva-portal.org/smash/get/diva2:535627/FULLTEXT01
[15] Ila Manuj and John T Mentzer, GLOBAL SUPPLY CHAIN RISK MANAGEMENT, Journal of Business Logistics, Vol. 29, No. 1, 2008, pp. 133-155.
[16] Robert J Ellison and Carol Woody, Supply-Chain Risk Management: Incorporating Security into Software Development, 2010 43rd Hawaii International Conference on System Sciences, 2010, pp. 1–10.
[17] Stacy Simpson, Fundamental Practices for Secure Software Development 2nd Edition, 2011. Available: http://www.safecode.org/publications/SAFECode_Dev_Practices0211.pdf
[18] Frank Swiderski and Window Snyder, Threat Modeling, Microsoft Press, 2004.
[19] NATIONAL STRATEGY FOR GLOBAL SUPPLY CHAIN SECURITY, JANUARY 2012. Available: http://www.whitehouse.gov/sites/default/files/national_strategy_for_global_supply_chain_security.pdf
[20] Wayne Jackson and Sonatype, Open Source and the Software Supply Chain: A Look at Risks vs. Rewards, CROSSTALK, Vol.26, No. 2, 2013, pp. 16–19.
[21] Mary Beth Chrissis, Mike Konrad and Michele Moss, "Ensuring Your Development Processes Meet Today's Cyber Challenges", CROSSTALK, Vol.26, No. 2, 2013, pp. 29–33.
[22] Stacy Simpson, The Software Supply Chain Integrity Framework: Defining Risks and Responsibilities for Securing Software in the Global Supply Chain, 2009. Available: http://www.safecode.org/publications/SAFECode_Supply_Chain0709.pdf
[23] Tyson Storch, Toward a Trusted Supply Chain: A Risk Based Approach to Managing Software Integrity, Microsoft Corp, 2011.
[24] Donald F. Donahue, The Public-Private Partnership and Supply Chain Resilience, 2009. Available: http://www.dtcc.com/downloads/leadership/speeches/Donald_F_Donahue_FS-ISAC_Speech.pdf
[25] Stacy Simpson, Software Integrity Controls: An Assurance-Based Approach to Minimizing Risks in the Software Supply Chain, 2010.
[26] Jason Dedrick, Sean Xin Xu and Kevin Xiaoguo Zhu, How Does Information Technology Shape Supply-Chain Structure? Evidence on the Number of Suppliers, Journal of Management Information Systems, Vol. 25, No. 2, 2008, pp. 41-72.
[27] Kristin Brennan and Coverity, Managing Risk in the Software Supply Chain Through Software Code Governance, CROSSTALK, Vol.26, No. 2, 2013, pp. 8–9.
[28] Johan Gregoire, Koen Buyens, Bart De Win, Riccardo Scandariato and Wouter Joosen, On the Secure Software Development Process: CLASP and SDL Compared, 29th International Conference on Software Engineering Workshops, 2007.
[29] Brian Chess and Brad Arkin, "Software Security in Practice", IEEE Security & Privacy, Vol. 9, No.2, 2011, pp. 89-92.

ISBN: 978-988-19252-3-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)