An Approach to Quantify Cybersecurity Risk in Terms of Functional Safety Requirement in Connected System

YIWEN CHEN[1] TAKASHI KAWAUCHI[1] CHINATSU YAMAUCHI[1]
SATOSHI KAI[1] ERIKO ANDO[1]

Abstract: A connected control system brings about real-time information change with external world but it also brings about cyber threats giving damage on functional safety or even jeopardizing human's life. Under consideration of cybersecurity risk damaging on functional safety, we establish a risk classification scheme called Cybersecurity Level (CSL) to interpret how secure a connected system is. CSL is classified into multiple levels according to attack success period which is regarded as a criteria of quantified cybersecurity risk in terms of functional safety requirement. We propose an approach to evaluate attack success period and validate feasibility of the approach by utilizing a connected system as first trial. Through our approach, we are able to quantitatively validate necessity and sufficiency of security controls throughout entire system DevOps phase, and further clarify efficient means to reduce cybersecurity risk and enhance secure level of a connected system.

1. Introduction

A connected control system brings about real-time information change with external world but it also brings about cyber threats giving damage on functional safety or even further jeopardizing human's life. One famous case happening in Black Hat USA 2015 was that the attackers enabled to control steering wheels and an engine in an automotive through hacking to an infotainment system [1] so that human's safety was possibly harmed. Not only in the automotive industry [1, 2] but also in others industries such as healthcare [3], railway [4], aviation [5], or others safety-critical systems [6, 7], protection of functional safety against cyber threats has became an unavoidable problem.

For this reason, it is necessary to make sure both functional safety and cybersecurity throughout entire DevOps phase [8]. The safety [9] and the cybersecurity [10] risk assessment frameworks respectively helped identification of hazards and threats in design phase but they didn't execute subsequent evaluations in operation phase to confirm whether system performance actually met expected assessment results. To keep security not only in develop phase but operation phase by updating patches of security functions, Y. Tung et al. [11] proposed an integrated test framework covering whole lifecycle to improve accuracy of security test. However, the framework only concentrated on network security without considering functional safety.

In the field of functional safety, there is a risk classification scheme proposed in IEC 61508 called safety integrity level (SIL) which is able to interpret secure level of functional safety in DevOps phase. SIL indicates different secure level according to different level of controllable safety risk, one of which is probability of failure. A safety developer prescribes desired SIL of an evaluated system and adjusts number and reliability of safety functions [12] to satisfy regulated probability of failure depending on the prescribed SIL. By comparing evaluation results among different combinations of safety functions, the safety developer can understand necessity of safety functions for satisfying the prescribed goal. Moreover, probability of failure is derived by failure rate polarized to two opposing cases which are safe failure rate and dangerous failure rate. Under hypothesis of failure to functions happening in operation phase, the safety developer checks safety status of the system by validating failure rate locating in safe or dangerous zone. In other words, the safety developer takes happening of failure to functions into consideration and is able to confirm whether functional safety still maintains in operation phase even failures had happened to confirm sufficiency of safety functions.

We propose a new term called secure functional safety indicating protection of functional safety against not only safety risk but also cybersecurity risk. By referring from the concept of SIL, we establish a risk classification scheme shown in Figure 1 called cybersecurity level (CSL) to interpret how secure of secure functional safety a connected system is. CSL is capable to indicate secure level of entire DevOps phase including design, test and operation phase which means CSL is able to interpret long-term safety and security of the connected system.

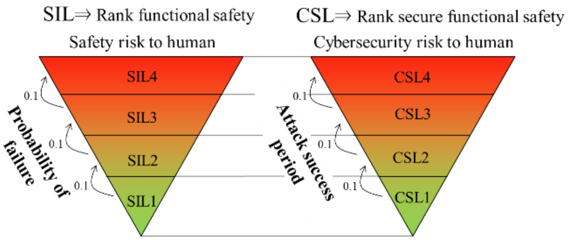


Figure 1. Concept of CSL and comparison between SIL

CSL is classified into multiple levels according to attack success period regarded as a criteria of quantified cybersecurity risk in terms of functional safety requirement. Based on the definition of an attacker's behaviors in cyber kill chain [13], we define attack success period as the period from delivery of attack to actions on objective. In other words, a high security system is requested to notice or even deal with the attack within this period as fast as possible. A security developer utilizes multiple security

[1] Hitachi, Ltd. Research & Development Group

controls in design phase or reduces number of vulnerability by updating patches in test and operation phase to shorten attack success period. In this paper, we won't provide clear number of attack success period for each level in CSL. Referring from SIL, we assume that enhancing CSL of a connected system to next higher level can be achieved by shortening attack success period to 0.1 times.

We also propose an approach to evaluate attack success period being applicable to entire DevOps phase and utilize a connected system as our first trial to validate feasibility of the approach. For design phase, we validate whether increasing number of security controls indeed makes attack success period reduce to 0.1 times or not to examine necessity of security controls. For test and operation phase, we validate whether reducing number of vulnerability also reduces attack success period to 0.1 times by executing multiple penetration tests or updating patches. To validate sufficiency of security controls, we assume successful attack happening in a specific place, and confirm attack success period didn't exceed to certain range so that we are able to claim that the system still keeps safe. As a result, through our approach of quantifying cybersecurity risk in terms of functional safety requirement, we expect to interpret secure level of a connected system throughout entire DevOps phase.

2. Research Approach

2.1 Steps of proposed approach

We propose an approach composed of five steps to evaluate attack success period regarded as quantified cybersecurity risk in terms of functional safety requirement whose flowchart is shown in Figure 2. The applicable target of the approach is a connected system consisted of both safety elements and security controls. Step1 to Step3 are derived from basic steps in security risk management which are for identifying cyber threats bringing risk to assets. Step4 and Step5 are the advanced extension from security risk management which are the cores for secure functional safety. Our approach is useful for interpreting how secure a connected system is and confirming performance of security controls in DevOps phase even an analyst isn't equipped with professional security knowledge. In the approach, we consider cyber threats coming from vulnerability without considering internal cyber threats or others human's miss. The details of each step are introduced in the following.

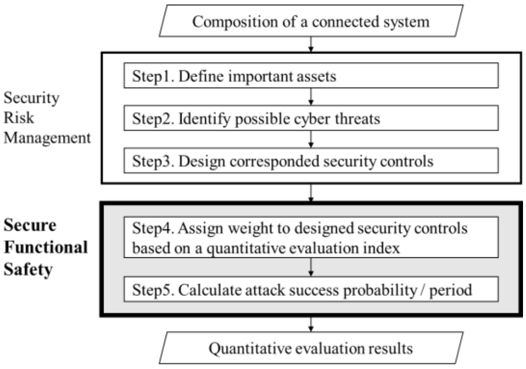

Composition of a connected system
Security Risk Management
Step1. Define important assets
Step2. Identify possible cyber threats
Step3. Design corresponded security controls
Secure Functional Safety
Step4. Assign weight to designed security controls based on a quantitative evaluation index
Step5. Calculate attack success probability / period
Quantitative evaluation results

Figure 2. Flowchart of proposed approach

Step1: Define important assets to clarify protected target

Step1 is to define important assets among safety elements in an evaluated connected system to clarify protected targets and allocate these assets to a hierarchal architecture shown in Figure 3. One big difference from conventional security risk management is that what conventional security tends to protect is information whereas the goal in here is to protect the safety elements to ensure physical functional safety not jeopardizing human's life. We propose a general hierarchical architecture consisted of Cloud layer, Information layer, Information-Physical control layer, Physical control layer being independent to each other. We assume that the protection way of connected systems is defense in depth [14] so that even cyber threats successfully passing through certain layers, functional safety can still be maintained. The protected safety elements are classified into this architecture and domain isolation between different layers is beneficial to apply the proposed approach to a large-scale system and clarify where is the place that cyber threats come from.

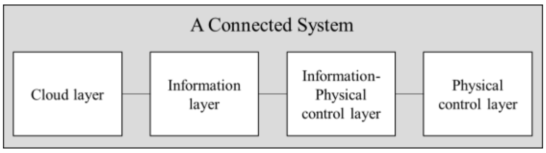


Figure 3. Hierarchal architecture of a connected system

Step2: Identify possible cyber threats damaging to defined assets

Step2 is to identify possible interfaces or entry points of cyber threats based on the defined important assets in Step1. The technique for identifying cyber threats is attack tree which is a common technique in security risk management to analyze cyber threats in a system. Attack tree is a top-down modeling technique consisting of one root and leaves. The root represents each defined important asset and the leaves derived by 5W method are possible cyber threats giving damage on each asset.

Step3: Design security controls handling identified cyber threats

Based on the identified results from Step2, Step3 is to design appropriate security controls to handle the identified cyber threats. There are two categories of security controls: direct and indirect. Direct security controls include access control, cryptography, digital sign and so on; while indirect security controls include audit, monitoring, education, incident handling and so on. Until now, the introduced steps are essential to security risk management but the novel point of our approach is that we utilize the identified results to achieve quantifying cybersecurity risk represented by attack success period.

Step4: Assign evaluation weight to designed security controls

The designed security controls handling identified cyber threats derived from Step3 form attack paths of cyber threats and Step4 is to assign evaluation weight to each security control designed in the attack paths. The happenings of cyber incidents are triggered by inadequate function defects, so we recommend a quantitative evaluation index – cover rate of known vulnerability indicating the amount of function defects. The cover rate of known vulnerability is acquired based on how much known vulnerability security controls are able to cover among total known vulnerability without considering unknown vulnerability. The total number of known vulnerability is referred from different vulnerability databases such as Common Vulnerabilities and Exposures (CVE), National Vulnerability Database (NVD), or Japan Vulnerability Notes (JVN) depending on individual test requirements. As our previous mentioned assumption, we consider cyber threats only coming from vulnerability, so cover rate of known vulnerability is able to be regarded as quantification of effectiveness of security controls. The value of cover rate of known vulnerability locates between 0 and 1, and the larger value indicates that there are less number of vulnerability in security controls.

Step5: Calculate attack success period by switching attack success probability to period

After security controls are assigned with quantitative weights, the technique of our approach is to firstly acquire attack success probability and then switch probability to period through system operation time. To acquire attack success probability in a connected system, we utilize a tree-based approach [15, 16] called event tree analysis (ETA) and integrate it with a reliability modeling in series and parallel system. ETA is a bottom-up modeling technique to analyze function defects or system failures. One biggest feature of ETA is that it disassembles multiple events in the order of occurrence of events and separates all events to two opposite statuses such as work & failure, safe & danger, normal & malfunction and so on. All analyzed events are drawn in tree shape until a certain result or event happened. In our approach, an event represents that one layer is successfully attacked so dynamic changes of attacked layers are able to be recorded through ETA.

On the other hand, the reliability modeling in series and parallel system is an analysis technique in reliability engineering. It supposes each component in a system is independent and malfunction probability of whole system R can be calculated shown in Figure 4. In series system, malfunction probability of whole system can be represented as the product of malfunction probability of each component R_n where n represents the nth component; whereas in parallel system, malfunction probability of whole system is the supplementary value of the product of the probability of all components normally working.

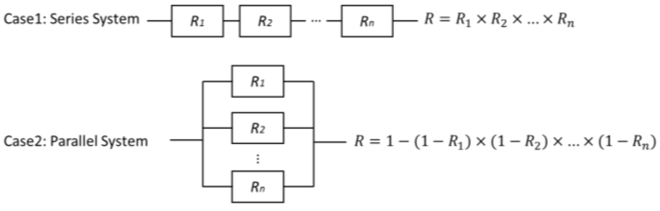

Case1: Series System
R_1
R_2
R_n
$R = R_1 \times R_2 \times \ldots \times R_n$
Case2: Parallel System
R_1
R_2
R_n
$R = 1 - (1 - R_1) \times (1 - R_2) \times \ldots \times (1 - R_n)$

Figure 4. Reliability modeling in series and parallel system

In our approach, each layer is divided to attack failure probability and attack success probability. If one layer was successfully hacked, it will bring damage to next layer until it harms to functional safety giving damage to human. Attack success probability of each single layer is calculated through the reliability modeling and replace the malfunction probability as the supplementary value of the cover rate of known vulnerability. After the calculation of attack success probability λ_i where i indicates the layer where attack comes from, attack success period t is driven by attack success probability λ_i by considering system operation time. Finally, attack success period t of a connected system can be represented as the minimum value among all attack paths defined in the following equation:

$$t = \min\{\lambda_i \times t_0\} \qquad (1)$$

where t_0 means system operation time. The reason why we choose the minimum value of products as attack success period is because cyber threats may come from different attack paths and what we want to protect is to ensure that system is still safe even the worst case happened.

2.2 Validation of necessity & sufficiency of security controls

Until here, we describe the approach to evaluate attack success period by switching probability to period through system operation time. After the calculation of attack success period of each single layer, we utilize ETA to calculate accumulated probability starting from each layer ending on physical layer to hypothesize successful attack passing through each layer. We tend to confirm the system still keeps human's safety even cyber threats successfully pass to certain layer to confirm necessity and sufficiency of security controls.

Moreover, we know that 0.1 times reduction of attack success period is able to enhance CSL to higher secure level. Since attack success period is derived from the product of attack success probability and system operation time, we can know that 0.1 times reduction of probability is also able to enhance CSL to higher secure level. We utilize this characteristic and validate whether accumulated attack success probability reduce to 0.1 times or not by (1) increasing number of security controls and (2) reducing number of vulnerability of security controls. Therefore, we can confirm whether secure level of the system is improved or not in develop and operation phase.

3. Trial Application

We validated the feasibility of our approach by utilizing a connected system as our first trial to verify whether we were capable to quantify attack success period through the approach. We especially validated the feasibility of Step4 and Step5 in Figure 3 to confirm that assigning weight to designed security controls based on a quantitative evaluation index is a feasible way to quantify cybersecurity risk in terms of functional safety requirement. Moreover, under the hypothesis of success attack happening, we (1) increased number of security controls and (2) reduced number of vulnerability of security controls, and validated accumulated attack success probability indeed reduce to 0.1 times or not to understand the necessity and sufficiency of security controls.

3.1 Introduction of first trial

We classified the security controls in the trial into five categories shown in different lines in Table-1 which were access control, cryptography, communications security, operations security, and physical security. The five categories of security controls were commonly utilized in security design of automotive based on a statistical result [16]. We assigned cover rate of known vulnerability to each category based on the same result [16] and the classification and the assigned results were shown in Table-1.

To evaluate secure level in test and operation phase, we suggested a new variable called test level representing completeness of executed vulnerability test. We utilized this variable to validate the reduction of number of vulnerability indeed help reduce attack success period or not. For example, when we set test level to 0.5, it means that number of uncovered vulnerability reduce to half so that cover rate of known vulnerability is able to be increased.

We utilized the five categories of security controls and designed allocation of them based on empirical experience. We validated the feasibility of the approach by confirming whether we could calculate accumulated probability and quantify attack success period or not. Then, we increased the number of security controls by following the risk assessment results in JASO TP15002 and reduced the number of vulnerability by setting test level to 0.5 from 1.0. Through examining whether accumulated probability was reduced to 0.1 times, we were able to validate the necessity and the sufficiency of security controls.

Table-1 Classification of security controls and cover rate of known vulnerability of each category under different test level

Category	Security Controls	Cover rate of known vulnerability (Test level = 1.0)	Cover rate of known vulnerability (Test level = 0.5)	Pattern in attack map
Access Control	Device authentication Connection authentication User authentication	0.7	0.85	◌
Cryptography	Cryptography	0.17	0.585	○
Communications Security	Message authentication	0.75	0.875	◯
Operations Security	Center monitoring Behavior monitoring	0.43	0.715	◌
Physical Security	Hardware countermeasure	0.36	0.68	◌

3.2 Verification results

3.2.1 Verification of empirical design

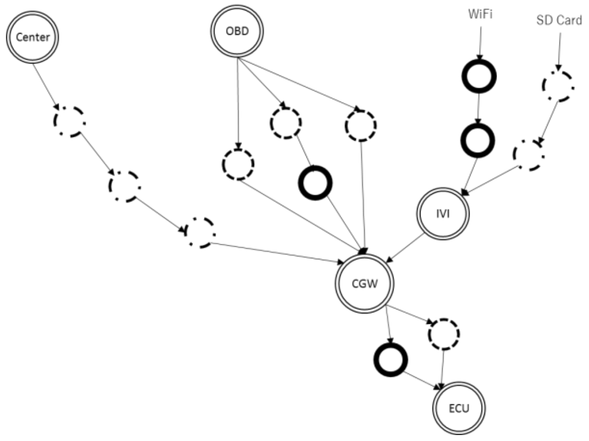


Figure 5. Attack paths in empirical design

In the empirical design, we defined the important assets as in-vehicle infotainment system (IVI), Central Gateway (CGW), and electronic control unit (ECU) and divided the connected system to four layers by referring from the hierarchal architecture in Step1. We designed security controls based on the identified cyber threats through Step2 and Step3, and the designed results were shown in the attack paths in Figure 5. There were four access control, four communication security, five operations security in the empirical design to protect the important assets. We calculated attack success probability of each layer by assigning cover rate of known vulnerability to each security control under test level equaling to 1.0 in Step4. The results of attack success probability of IVI, CGW, and ECU were 0.36709, 0.63069, and 0.47500 respectively shown in top part of Figure 6. On conclude, we were able to acquire attack success probability of each single layer and acquire attack success period by multiplying attack success probability to system operation time.

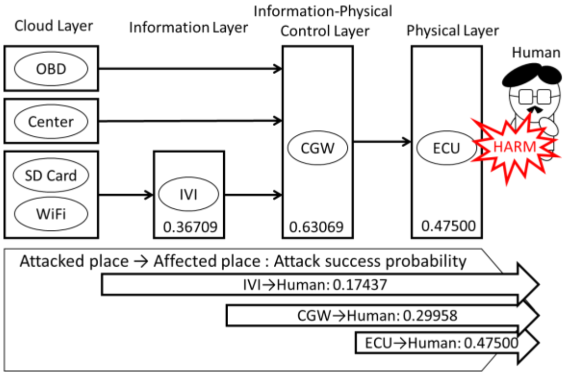


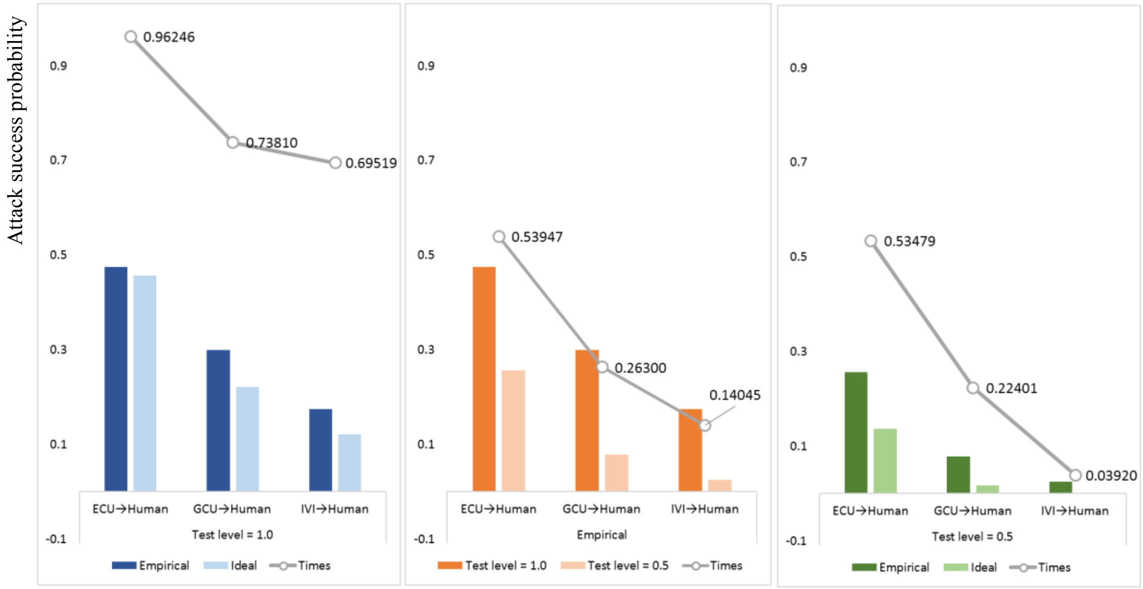
Figure 6. Quantitative results of attack success period in each layer and results of accumulated attack success probability

3.2.2 Verification of necessity & sufficiency of security controls

We calculated each accumulated probability based on ETA shown in bottom part of Figure 6. The top arrow indicated attack success probability giving damage to human when all important assets were successfully hacked. The middle arrow hypothesized cyber threats coming from CGW and both CGW and ECU were successfully attacked; and so on. Then, we evaluated the accumulated probability under (1) increasing number of security controls but with same number of vulnerability (2) reducing number of vulnerability but with same number of security controls. In here, we didn't set the quantitative threshold to attack success probability but we chose to compare two improved designs with the empirical design. By validating whether the improved designs were able to reduce cybersecurity risk to 0.1 times or not, we were able to validate the necessity and the sufficiency of security controls.

(1) Increase of number of security controls

We increased number of security controls by following a risk assessment result in JASO TP15002 whose design result was shown in Figure 7. The defined assets were same as the empirical design but utilized seven access control, three cryptographies,



(1) (2) (3)

Figure 8. Comparison results between different cases. (1) Increase of number of security controls (2) Reduce number of vulnerability (3) Simultaneous utilization of (1) and (2)

nine communications security, fifteen operations security, and one physical security. We assigned cover rate of known vulnerability to security controls under same test level as the empirical design and calculated reduction ratio of accumulated probability between the empirical design and the improved design shown in Figure 8(1). The results showed that best improvement was 0.695 times meaning that increase of number of security controls in design phase indeed helped reduce cybersecurity risk but it failed to enhance CSL to higher secure level.

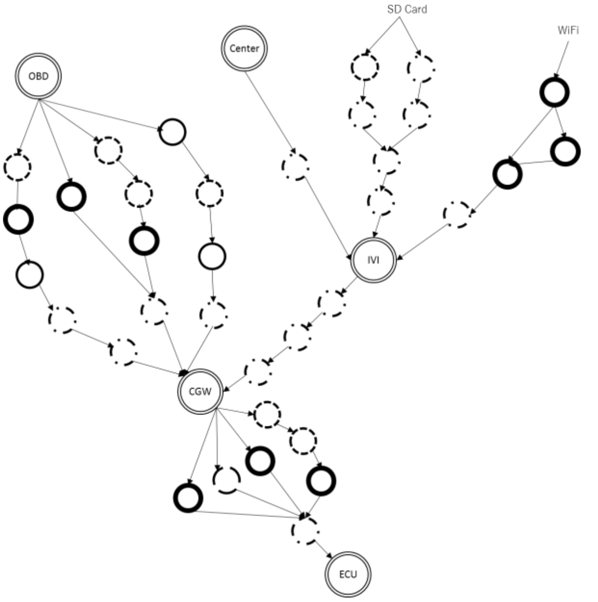


Figure 7. Increase of number of security controls

(2) Reduce number of vulnerability

In second situation, we utilized same number of security controls but reduced number of vulnerability by setting test level to 0.5. The comparison results with the empirical design in Figure 8(2) showed that the reduction of number of vulnerability also helped reduce cybersecurity risk. Best improvement was 0.14045 times which performed stronger impact than increasing number of security controls. Therefore, we were able to conclude that reducing number of vulnerability in test and operation phase had bigger influence than increasing number of security controls in design phase. Next, we tended to know best performance of security controls so we utilized (1) and (2) simultaneously whose results were shown in (3).

(3) Simultaneous utilization of (1) and (2)

The results in Figure 8(3) showed that the best improvement of reducing cybersecurity risk by simultaneously utilizing the two means is 0.0392 times. We could know that utilizing both made the system not only reduce cybersecurity risk but also enhance secure level which was the best performance of security controls that system stakeholders desired.

4. Discussion

In this study, in order to interpret how secure a connected system is, we have established a risk classification scheme CSL divided to multiple levels according to attack success period. We proposed an approach to evaluate attack success period and utilized a connected system as the first trial to confirm the feasibility of our approach. We showed the evaluation results of attack success probability, and claimed to acquire period by the product of attack success probability and system operation time based on equation 1.

However, here we didn't provide the clear calculation results of attack success period while we tend to discuss a premise of the approach. Currently, we evaluated attack success probability based on each possible attack path where cyber threats may happen. However, depending on an attacker's ability or knowledge which is also suggested in Common Criteria, the attacker may fail to hack to the connected system even though there are entry points or attack interfaces existing. In other words, we believe that it requires an approach to quantify the attacker's ability which haven't be defined in this paper.

On the other hand, to validate the necessity and the sufficiency of security controls, we increased number of security controls and reduced number of vulnerability, and calculated accumulated probability based on the improved designs. The results showed that reducing number of vulnerability in test and operation phase gave better performance than increasing number of security controls in design phase. Also, the results showed that separated utilization of two means failed to reduce probability to 0.1 times but simultaneously utilizing both gave the best performance. Therefore, we could conclude that it is necessary to balance these two improvement means and find out the key security control to achieve higher CSL.

5. Summary

A connected control system brought about real-time information change with external world but it also brought about cyber threats damaging on functional safety or even further jeopardizing human's life. To interpret secure level of secure functional safety in DevOps phase, we had established a risk classification scheme called CSL classified into multiple levels according to attack success period. We proposed an approach of evaluating attack success period regarded as the quantified cybersecurity risk in terms of functional safety requirement. We verified the feasibility of the approach by utilizing a connected system as the first trial and evaluated attack success probability of each layer where cyber threats might come from. Furthermore, we validated the necessity and the sufficiency of security controls by utilizing (1) higher number of security controls but with same vulnerability and (2) same number of security controls but with less vulnerability. The results showed that both of them were able to reduce risk but only simultaneously utilizing two means was able to enhance CSL to higher secure level. Our future work is to deal with dependent security controls locating in different layers and validate the feasibility of the approach in others connected systems. Finally, we expect to set up the completed risk classification scheme CSL and clarify value of the quantitative criteria depending on each level in the scheme.

6. References

[1] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle", 2015.

[2] S. Nie and W. Zhang, "TESLA Hacking 2016 and 2017", presented at the 15th Conf. escar Europe, Berlin, Germany 2017.

[3] K. C. T. Hing, C. C. Tin, T. K. Li, Y. D. Chuah, A. H. L. and Cheum, T. R. Wei, "Development of Home Monitoring System with Integration of Safety and Security Modules", in Sustainable Utilization and Development in Engineering and Technology (STUDENT), Kuala Lumpur, 2012.

[4] J. Braband and M. Seemann, "On the Relationship of Hazards and Threats in Railway Signaling", presented at the proceedings of the International IET System Safety Conf., Manchester, UK 2012.

[5] C. Quanxin, Y. Linfang, C. Bin, and F. Chenchen, "Enhancing Network Security Strategies against External Threats to Civil Aircrafts", presented at the IEEE 18th International Conference on High Performance Computing and Communications, Sydney, Australia 2016.

[6] W. Young and N. G. Leveson, "An Integrated Approach to Safety and Security Based on Systems Theory", Communications of the ACM, vol. 57, pp. 31-35, 2014.

[7] Z. Huo, D. Zeckzer, P. Liggesmeyer, and O. Mackel, "Identification of Security-Safety Requirements for the outdoor robot RAVON using Safety Analysis Techniques", presented at the 5th International Conf. on Software Engineering Advances (ICSEA), 2010.

[8] C. Schmittner, Z. Ma, and E. Schoitsch, "Combined Safety and Security Development Lifecycle", presented at the IEEE 13th International Conference on Industrial Informatics (INDIN), Cambridage, UK 2015.

[9] F. Rodrigues, A. Barbosa, and J. S. Baptista, "Hazards Identification during Design Phase", Occupational Safety and Hygiene V, pp. 415- pp. 426, 2017.

[10] G. Macher, E. Armengaud, E. Brenner, and C. Kreiner, "A Review of Threat Analysis and Risk Assessment Methods in the Automotive Context", presented at the International Conf. on Computer Safety, Reliability, and Security (SAFECOMP 2016), pp. 130-141, 2016.

[11] Y. Tung, S. Lo, J. Shih, and H. Lin, "An Integrated Security Testing Framework for Secure Software Development Life Cycle", presented at 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), Kanazawa, Japan 2016.

[12] V. d. Dianous and C. Fievex, "ARAMIS project: A more explicit demonstration of risk control through the use of bow-tie diagrams and the evaluation of safety barrier performance", Journal of Hazardous Materials, vol. 130, issue 3, pp. 220-233, 2006.

[13] Lockheed Martin, "Cyber Kill Chain," http://www.lockheedmartin.com/us/what-we-do/information-technology/cyber-security/cyber-kill-chain.html, accessed June 30, 2015.

[14] W. C. Neumann, T. E. Corby, and G. A. Epps, "System for secure computing using defense-in-depth architecture", U.S. Patent 10919631, issued August 17, 2004.

[15] C. Cho., W. Chung, and S. Kuo, "Using Tree-Based Approaches to Analyze Dependability and Security on I&C Systems in Safety-Critical Systems", IEEE Systems Journal, 2017.

[16] R. Kumar and M. Stoelinga, "Quantitative Security and Safety Analysis with Attack-fault Trees", 2017 IEEE 18th International Symposium on High Assurance Systems Engineering, 2017.

[17] P3 communications, "P3 carries out global automotive cybersecurity benchmark of connected vehicles", http://telematicswire.net/p3-carries-out-global-automotive-cybersecurity-benchmark-of-connected-vehicles/