Threat Analysis using Vulnerability Databases
– Matching Attack Cases to Vulnerability Database by Topic Model Analysis –

Katsuyuki Umezawa
Department of Information Science
Shonan Institute of Technology
Fujisawa, Kanagawa 251–8511, Japan
e-mail: umezawa@info.shonan-it.ac.jp

Yusuke Mishina
Information Technology Research Institute (ITRI)
Advanced Industrial Science and Technology (AIST)
Koto-ku, Tokyo 135–0064, Japan
e-mail: yusuke.mishina@aist.go.jp

Sven Wohlgemuth
Research & Development Group
Hitachi, Ltd.
Yokohama, Kanagawa 244–0817, Japan
e-mail: sven.wohlgemuth.kd@hitachi.com

Kazuo Takaragi
Information Technology Research Institute (ITRI)
Advanced Industrial Science and Technology (AIST)
Koto-ku, Tokyo 135–0064, Japan
e-mail: kazuo.takaragi@aist.go.jp

***Abstract*—In this paper, we propose a threat analysis method utilizing vulnerability databases and system design information. The method is based on attack tree analysis. We created an attack tree on a evaluation target system and some attack trees on a known vulnerability, and combined the two types of attack trees to create more concrete attack trees. This enables us to calculate the probability of occurrence of a safety accident and to utilize attack trees in future analysis. Since any document has a latent topic and keywords can be generated from that topic, our vulnerability analysis algorithms use topic model analysis for natural language processing to create and analyze attack trees. The National Institute of Advanced Industrial Science and Technology (AIST) has developed a security requirement analysis support tool using topic model analysis technology. Specifically, we performed matching of attack case papers to vulnerability databases and could find about 20 items, including exact matches, from 500 items of a vulnerability database on the basis of an attack method description.**

Keywords–Threat Analysis; Vulnerability Information; Attack Tree.

I. INTRODUCTION

Interference and interruption to safety due to security incidents are recognized as a big problem in safety critical systems, such as those for electric power, information communication, automobile, aviation, railway, and medical care. Regarding the security of in-vehicle communication in the EVITA project [1], risk analysis, security requirement setting, architecture design, and prototyping, as well as a demonstration of a Hardware Security Module (HSM) by using Field Programmable Gate Arrays (FPGAs), were conducted. An attack tree was used for risk analysis in the EVITA project. One way to analyze the causal relationship between safety (hazard) and security (threat) is to express that relationship with a combination of a Fault Tree (FT) and Attack Tree (AT) [2].

The US-based MITRE Corporation provides several tools for vulnerability reporting and aggregation in a vulnerability database (DB). In Common Vulnerabilities and Exposures (CVE) [3], individual software vulnerabilities are stored in a DB. In Common Weakness Enumeration (CWE) [4], common vulnerabilities are cataloged with a focus on the cause of the vulnerability. Furthermore, Common Attack Pattern Enumeration and Classification (CAPEC) [5] is a DB classified by attack pattern.

Scientific literature related to safety analysis using FTs is, nowadays, mature [2]. However, the complexity of the problem has significantly increased in security analysis. Elaborate attacks occur with multiple combinations of those vulnerabilities. It is not easy to create an AT that comprehensively captures such possibilities.

We have focused on such problems and proposed a threat analysis method using a vulnerability DB as a practical approach [6][7]. First, we assumed that many attacks were imitations or minor changes of known attacks. Therefore, we believed that expressing attack cases that occurred in the past by using an AT could enable a designer (defender) to become aware of related attacks (recognize the danger). By gradually and continuously applying this approach, it can be useful for reducing vulnerability.

We proposed an algorithm that includes a process for matching each node of an AT described in natural language [6][7]. However, the matching method utilized was not specified. In this paper, we evaluate the feasibility of this unspecified matching process using a topic model analysis method.

In Section II, we summarize the threat analysis method we proposed in [6] and [7]. In Section III, we introduce topic model analysis. In Section IV, we verify the feasibility of matching attack cases to vulnerability DBs and show the result. Section V concludes this paper by summarizing the key points and give an outlook on future activity.

II. THREAT ANALYSIS USING VULNERABILITY DATABASES

This section presents a summary of our proposed method [7]. An overview of the threat analysis method using the vulnerability DB is shown in Figure 1. The proposed threat analysis method conducts the following three procedures:

- Create vulnerability model information.
- Create lower-level component information embedded in software.
- Perform threat analysis on the basis of design information of analysis target system.

A. *Creating vulnerability model information*

The MITRE Corporation has published several forms of vulnerability DBs [3]–[5]. However, it is difficult to create an AT for a concrete target (for example, a connected car) simply by referring to these DBs. We will create an AT with a reference to existing attack case literature, reports, etc.

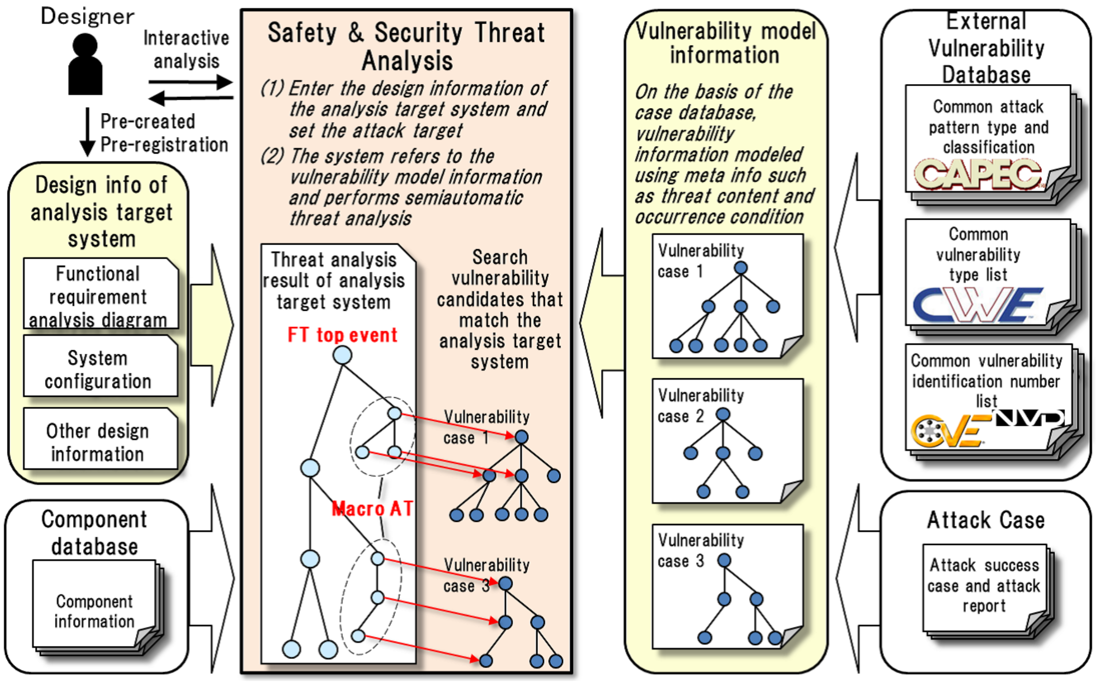

Designer
Interactive analysis
Pre-created
Pre-registration
Design info of analysis target system
Functional requirement analysis diagram
System configuration
Other design information
Component database
Component information
Safety & Security Threat Analysis
(1) Enter the design information of the analysis target system and set the attack target
(2) The system refers to the vulnerability model information and performs semiautomatic threat analysis
Threat analysis result of analysis target system
FT top event
Macro AT
Search vulnerability candidates that match the analysis target system
Vulnerability case 1
Vulnerability case 3
Vulnerability model information
On the basis of the case database, vulnerability information modeled using meta info such as threat content and occurrence condition
Vulnerability case 1
Vulnerability case 2
Vulnerability case 3
External Vulnerability Database
Common attack pattern type and classification
CAPEC
Common vulnerability type list
CWE
Common vulnerability identification number list
CVE
NVD
Attack Case
Attack success case and attack report

Figure 1. Overview of proposed threat analysis method

Thus, let the AT be obtained from the existing vulnerability DB and existing attack report be called the first_AT. This first_AT is hierarchically drawn into a top node, a collection of intermediate nodes and bottom nodes. A single first_AT is created for each vulnerability. A vulnerability DB such as CVE monotonically increases, so it is not necessary to recreate the first_AT once it has been generated. As will be described later, second_AT can be used as a first_AT in subsequent analysis, so that each time an analysis is performed, the quantity of first_ATs will increase.

B. Proposal of component database

In embedded systems, such as those for automobiles and general Internet of Things (IoT) devices, required lower-level components embedded within the software, not the software itself, are incorporated. However, a vulnerability DB such as CVE only includes vulnerability information for software as a whole and does not describe information on the lower-level components embedded within the software. Therefore, a correspondence table between the software version and the version for its lower-level components would be beneficial. This makes it easy to check vulnerability information at the manufacturing stage of embedded systems such as those in IoT devices. The method to create a component DB is outside the scope of this proposal.

C. Threat analysis algorithm

This section describes the threat analysis algorithm. It corresponds to the “Safety & Security Threat Analysis” section in Figure 1. The algorithm, which is based on the vulnerability model shown in Section II-A, the component DB shown in Section II-B, and the design information of the analysis target system, is as follows:

(1) Create a second_AT with the top node as a safety accident related to the evaluation target system. At this time, even if the component is not directly included in the evaluation target system, a component judged to be related by referring to the component DB is included in the second_AT (the black circle node in Figure 2 (2)). The second_AT is hierarchically depicted using the top node, the multiple intermediate nodes, and the lowest nodes. Thus, a second_AT is created (Figure 2 (2)).

(2) One of the top nodes or intermediate nodes of the second_AT is selected and Natural Language Processing (NLP) is used to mechanically determine whether there is a first_AT having a natural language expression similar to nodes of the second_AT (Figure 2 (3)). If this is the case, the first_AT is temporarily added to the second_AT (Figure 2 (4)). OR gate is attached to the node of the second_AT as a temporary cause, and the first_AT is pasted below it. This is done for all nodes of the second_AT. As a result, the second_AT is expanded more after considering the existing vulnerability database, that is, the entire set of the first_AT.

(3) The focus is now on the temporary added nodes in the expanded second_AT. We check whether the added node is necessary. Specifically, we define a node unrelated to the component of the second_AT (different components or different versions) as FALSE nodes, and the FALSE node and the AND gate that is just above the FALSE node are deleted (Figure 2 (5)).

(4) Repeat steps 1–3 for all the first_ATs that are related to the second_AT as described above. After the modification, we evaluate the occurrence probability of the top node by using the modified second_AT.

In addition, [7] describes the mathematical formulation of this proposed algorithm, calculation of attack probability, and application of actual cases of car attacks [8][9].

ISBN: 978-1-61208-683-5

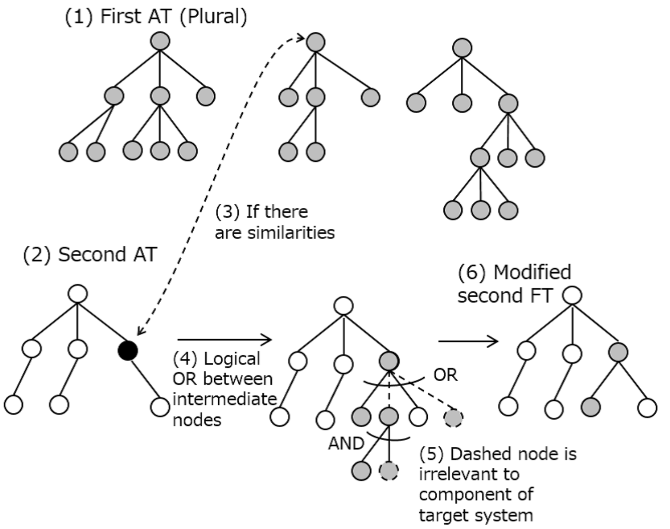
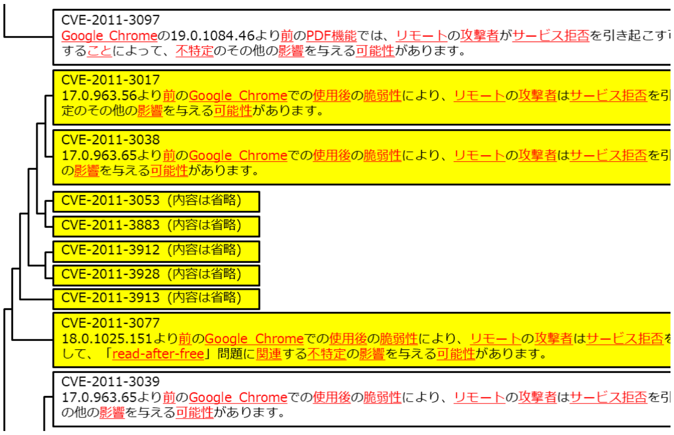

(1) First AT (Plural)
(3) If there are similarities
(2) Second AT
(6) Modified second FT
(4) Logical OR between intermediate nodes
OR
AND
(5) Dashed node is irrelevant to component of target system

Figure 2. Threat analysis algorithm (cited from [7])


CVE-2011-3097
Google Chromeの19.0.1084.46より前のPDF機能では、リモートの攻撃者がサービス拒否を引き起こす
することによって、不特定のその他の影響を与える可能性があります。

CVE-2011-3017
17.0.963.56より前のGoogle Chromeでの使用後の脆弱性により、リモートの攻撃者はサービス拒否を引
定のその他の影響を与える可能性があります。

CVE-2011-3038
17.0.963.65より前のGoogle Chromeでの使用後の脆弱性により、リモートの攻撃者はサービス拒否を引
の影響を与える可能性があります。

CVE-2011-3053（内容は省略）

CVE-2011-3883（内容は省略）

CVE-2011-3912（内容は省略）

CVE-2011-3928（内容は省略）

CVE-2011-3913（内容は省略）

CVE-2011-3077
18.0.1025.151より前のGoogle Chromeでの使用後の脆弱性により、リモートの攻撃者はサービス拒否を
して、「read-after-free」問題に関連する不特定の影響を与える可能性があります。

CVE-2011-3039
17.0.963.65より前のGoogle Chromeでの使用後の脆弱性により、リモートの攻撃者はサービス拒否を引
の他の影響を与える可能性があります。

Figure 3. (Part of) Hierarchy of vulnerability DB CVE

III. Topic Model Analysis

A. Latent Dirichlet Allocation (LDA)

A topic model formalized a document's properties in having a latent topic and each keyword of the document is regarded to be generated from that topic. In topic model analysis, we estimate latent topics from keywords. One of the analysis methods of topic models is Latent Dirichlet Allocation (LDA) [10]. This is a language model that assumes the probability distribution of the topic (parameter θ of the multinomial distribution) follows the Dirichlet distribution. In LDA, topics are selected in accordance with the Dirichlet distribution and words are selected in accordance with the probability distribution of words for that topic.

B. Topic model analysis tool

The National Institute of Advanced Industrial Science and Technology (AIST) has developed a security requirement analysis support tool using topic model analysis technology including LDA [11]. We preliminarily used this tool to verify whether the vast number of vulnerabilities CVE [3] listed in the order of discovery can be organized into a hierarchical structure by topic model analysis. Figure 3 shows the result of using 1500 cases from CVE-2011-3001 to CVE-2011-4500 after translating it to Japanese using Google Translate [12]. As shown in Figure 3, we see that similar vulnerabilities are classified near the hierarchical structure.

IV. Matching attack cases to vulnerability database

A. Outline explanation

As mentioned in Section II-C(2), we used NLP when matching and connecting the first_AT and the second_AT nodes. We verified the feasibility of this matching process.

We searched various reports to find vulnerabilities that should be related in the second_AT of the target system. However, depending on the report, the procedure of attack is shown but the concrete CVE number is not specified. Even in such a case, we can extract the corresponding CVE number from the attack description described in natural language.

To achieve this, we must find a node of the second_AT that conceivably matches the description in CVE. However, a mechanical word matching process will probably not lead to a correct result as it is dependent on the words used to describe sentences. The context or meaning of the known attack description in each report should be thoroughly examined. Therefore, we targeted the sentences of existing papers. Specifically, we targeted the actual case of a car attack [8]. The process flow is as follows.

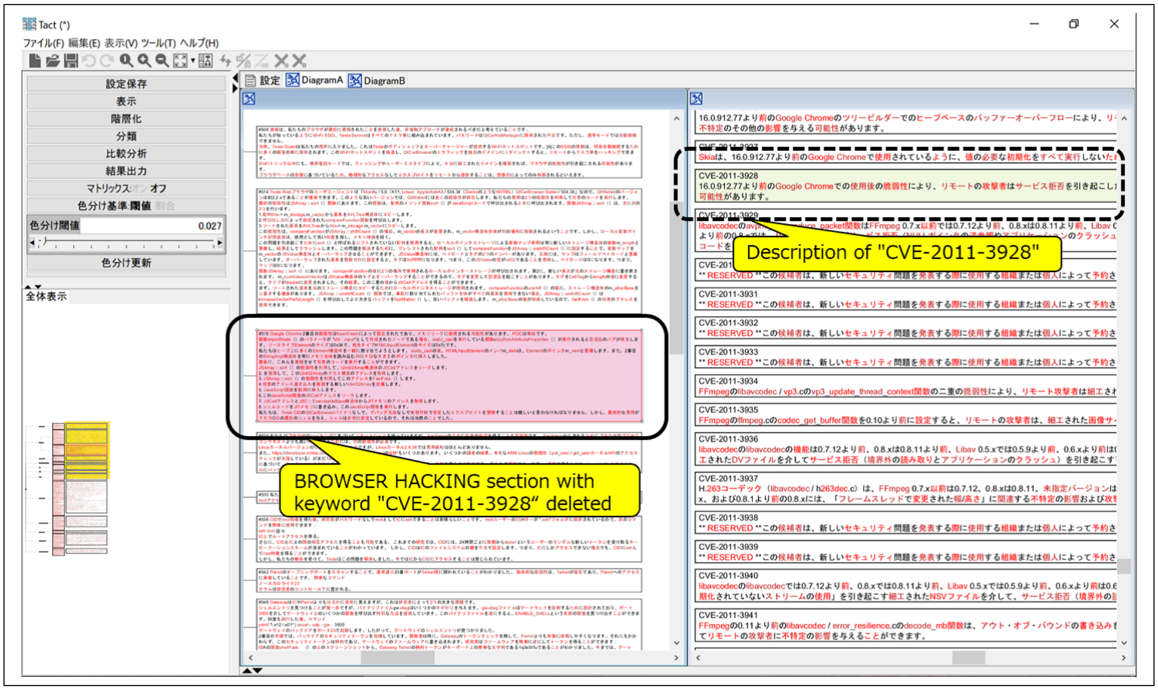
We translated the paper [8] into Japanese by using Google Translate because the tool we used only corresponded to Japanese. An advantage of utilizing such a translation is that it can prevent notation fluctuation of terms. However, since the section on BROWSER HACKING is long and its content is related to two vulnerabilities, it was divided into two. The vulnerabilities in question were CVE-2011-3928 and CVE-2013-6282. CVE-2011-3928 is described in the section on BROWSER HACKING, and CVE-2013-6282 is described in the section on LOCAL PRIVILEGE ESCALATION. If "CVE-2011-3928" or "CVE-2013-6282" is included as a keyword, it may be detected by keyword matching, so the keywords "CVE-2011-3928" and "CVE-2013-6282" were deleted from BROWSER HACKING and LOCAL PRIVILEGE ESCALATION, respectively.

However, regarding BROWSER HACKING, there is a problem of component inclusion relationship stated in the section II-B, and the keyword "Google Chrome" is added to the sentences in which WebKit is described. This is considered to be equivalent to referring to the component DB of the proposed method. Since the topic analysis tool used has an upper limit on the number of items to be handled, it was not possible to cover all CVEs, so we targeted 500 items before and after including the target vulnerability. The limitation of 500 items is not a constraint of the topic model analysis, but an implementation limitation of the tools we used.

We specifically targeted CVEs from CVE-2011-3501 to CVE-2011-4000 including CVE-2011-3928 and those from CVE-2013-6001 to CVE-2013-6500 including CVE-2013-6282. For each section of the paper and each CVE vulnerability, similar sentences were evaluated by topic model analysis. The keyword extraction method was known as "noun and Kana", the feature quantity extraction method was "LDA", and the sentence similarity "Cosine" option was used.

B. Analysis result

The result of matching each section of the paper to each CVE vulnerability is shown in Figure 4. When we click on



16.0.912.77より前のGoogle Chromeのツリービルダーでのヒープベースのバッファーオーバーフローにより、リ
不特定のその他の影響を与える可能性があります。

CVE-2011-3927
Skiaは、16.0.912.77より前のGoogle Chromeで使用されているように、値の必要な初期化をすべて実行しないた

CVE-2011-3928
16.0.912.77より前のGoogle Chromeでの使用後の脆弱性により、リモートの攻撃者はサービス拒否を引き起こし
可能性があります。

CVE-2011-3929
libavcodecのavpr
より前の0.8.xでは
コードを

CVE-201
** RESERVED **この候補者は、新しいセキュリティ問題を発表する際に使用する組織または個人によって予約さ

CVE-2011-3931
** RESERVED **この候補者は、新しいセキュリティ問題を発表する際に使用する組織または個人によって予約さ

CVE-2011-3932
** RESERVED **この候補者は、新しいセキュリティ問題を発表する際に使用する組織または個人によって予約さ

CVE-2011-3933
** RESERVED **この候補者は、新しいセキュリティ問題を発表する際に使用する組織または個人によって予約さ

CVE-2011-3934
FFmpegのlibavcodec / vp3.cのvp3_update_thread_context関数の二重の脆弱性により、リモート攻撃者は細工され

CVE-2011-3935
FFmpegのffmpeg.cのcodec_get_buffer関数を0.10より前に設定すると、リモートの攻撃者は、細工された画像サイ

CVE-2011-3936
libavcodecのlibavcodecの機能は0.7.12より前、0.8.xは0.8.11より前、Libav 0.5.xでは0.5.9より前、0.6.xより前は
工されたDVファイルを介してサービス拒否（境界外の読み取りとアプリケーションのクラッシュ）を引き起こす

CVE-2011-3937
H.263コーデック（libavcodec / h263dec.c）は、FFmpeg 0.7.x以前は0.7.12、0.8.xは0.8.11、未指定バージョンは
x、および0.8.1より前の0.8.xには、「フレームスレッドで変更された幅/高さ」に関連する不特定の影響および攻

CVE-2011-3938
** RESERVED **この候補者は、新しいセキュリティ問題を発表する際に使用する組織または個人によって予約さ

CVE-2011-3939
** RESERVED **この候補者は、新しいセキュリティ問題を発表する際に使用する組織または個人によって予約さ

CVE-2011-3940
libavcodecのlibavcodecでは0.7.12より前、0.8.xでは0.8.11より前、Libav 0.5.xでは0.5.9より前、0.6.xより前は0.6
期化されていないストリームの使用」を引き起こす細工されたNSVファイルを介して、サービス拒否（境界外の

CVE-2011-3941
FFmpegの0.11より前のlibavcodec / error_resilience.cのdecode_mb関数は、アウト・オブ・バウンドの書き込み
てリモートの攻撃者に不特定の影響を与えることができます。

Figure 4. Matching attack cases to vulnerability DBs

a sentence in the left pane, this tool will highlight similar sentences in the right pane. The solid lined area in the left pane is the BROWSER HACKING section with the keyword "CVE-2011-3928" deleted. When clicking on this area, the dashed lined area, which is the description of CVE-2011-3928 in the right pane, is highlighted and is judged to be similar. The number of items that included the appropriate CVE from the original 500 was filtered down to 22. It can be said that the smaller the number, the better. Regarding CVE-2013-6282, a similar result was obtained by matching the information of LOCAL PRIVILEGE ESCALATION with that of CVE, in this case 23 out of the 500.

V. Conclusion

In this paper, we performed matching of attack case paper with vulnerability DBs instead of matching nodes of ATs created from design information and attack cases with those created from vulnerability DBs. We confirmed the feasibility of matching known attack cases to vulnerability DBs using a topic model analysis tool. However, this approach does not guarantee the discovery and prevention of new sophisticated attacks that are completely different from those that occurred in the past. We believe that it is necessary to apply this method to threat analysis that utilize vulnerability DBs and system design information [6][7] and evaluate it in actual cases.

Acknowledgment

Sven Wohlgemuth's contribution to this work is based on his research at Albert-Ludwig University, Freiburg, Germany, and other organizations before he joined Hitachi, Ltd. in February 2017. This work was supported by the Council for Science, Technology and Innovation (CSTI), Cross-ministerial Strategic Innovation Promotion Program (SIP), "Cyber-Security for Critical Infrastructure" (funding agency: NEDO).

References

[1] A. Ruddle et al., "Deliverable D2.3: Security requirements for automotive on-board networks based on dark-side scenarios," Seventh Research Framework Programme of the European Community, July 2008, pp. 1–138.

[2] I. N. Fovino, M. Masera, and A. D. Cian, "Integrating cyber attacks within fault trees," Reliability Engineering and System Safety 94, 2009, pp.1394–1402.

[3] MITRE Corporation, "CVE - Common Vulnerability and Exposure," https://cve.mitre.org/ [retrieved: September, 2018]

[4] MITRE Corporation, "CWE List - Common Weakness Enumeration," https://cwe.mitre.org/data/ [retrieved: September, 2018]

[5] MITRE Corporation, "CAPEC - Common Attack Pattern Enumeration and Classification," https://capec.mitre.org/ [retrieved: September, 2018]

[6] K. Umezawa, Y. Mishina, K. Taguchi, and K. Takaragi, "A Proposal of Threat Analyses using Vulnerability Databases," Proceeding of the Symposium on Cryptography and Information Security (SCIS2018), 1C2-6, January 2018, pp. 1–8.

[7] Y. Mishina, K. Takaragi, and K. Umezawa "A Proposal of Threat Analyses for Cyber-Physical System using Vulnerability Databases", 2018 IEEE International Symposium on Technologies for Homeland Security (IEEE HST), October 2018.

[8] S. Nie, L. Liu, and Y. Du, "Free-Fall: Hacking Tesla from Wireless to Can Bus," Briefing, Black Hat USA 2017, July 2017. pp. 1–16.

[9] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," Briefing, Black Hat USA 2015, pp. 1–91.

[10] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation", in Journal of Machine Learning Research, 2003, pp. 1107–1135.

[11] K. Handa, H. Ohsaki, and I. Takeuti, "Security Requirements Analysis Supporting Tool: TACT," Information Processing Society of Japan (IPSJ) SIG Software Engineering (SIGSE), Proceeding of the Winter Workshop 2017. pp. 5–6.

[12] Y. Wu et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," arXiv:1609.08144, 2016. pp. 1–23.