Threat Analysis Using Topic Models in Large-Scale Vulnerability Databases and Security Incident Case Documents

Hiroki Koyanagi
Electrical and Information Eng., Shonan Inst. of Tech.
Cyber Physical Security Research Center, AIST
Kanagawa, Japan 20t2006@sit.shonan-it.ac.jp

Kazuo Takaragi
Cyber Physical Security Research Center, AIST
Tokyo, Japan kazuo.takaragi@aist.go.jp

Sven Wohlgemuth[1]
Intelligent Systems Laboratory, SECOM Co., Ltd.
Mitaka, Tokyo 181-8528, Japan
s-wohlgemuth@secom.co.jp

Katsuyuki Umezawa
Dept. of Information Science, Shonan Inst. of Tech.
Cyber Physical Security Research Center, AIST
Kanagawa, Japan umezawa@info.shonan-it.ac.jp

***Abstract*—It is crucial to design products bearing security in mind from the initial development stage. Consequently, many threat analysis support tools have been developed. However, it is difficult to determine the inherent threats in various designed documents written in natural language, which is used in the initial development stage. It is not uncommon to find attacks that closely resemble past attacks. In addition, many designs are limited in the number of data they can handle. We propose a method of extracting existing vulnerabilities similar to those used in the attack by collating a large vulnerability database with existing attack cases using Latent Dirichlet Allocation, one of the topic model methods. We apply the proposed method to several cases and verify its effectiveness.**

***Index Terms*—Security, Topic model, Natural language processing, Vulnerability Information**

I. Introduction

Since the 1990s, the need for information sharing on cyber threats has been increasing. Recently, the discovery of new attack methods, such as the 2015 and 2016 cyberattacks on the Ukrainian power grid and the 2016 Mirai botnet attack, is intriguing. In the world of security, where cyberattacks and malware infections frequently occur across national borders, certain mechanisms are being established for global information sharing. In this study, we assume that such mechanisms for information sharing have been provided. In this paper, we assume that the information-sharing mechanism is provided, and then, when cyber attacks or malware infections occur in external systems and the information is shared, we describe a method to analyze and evaluate whether our own systems are able to prevent the cyber attacks or malware infections, or whether they suffer damages due to insufficient countermeasures. A method to analyze and evaluate whether cyber attacks and malware infections can be prevented and whether the countermeasures are inadequate and cause damages is described. There is already an ISAC system for sharing cyber information via fields; moreover, there is a long history of ISAC operation in ICT, energy, and finance. In addition, computer security incident response teams (CSIRTs) are mechanisms for sharing information in case of cyberattacks by focusing on the information processing aspect. For example, the NIS Directive of the EU provides for the sharing of information on security measures and incidents across sectors of economic and social importance, such as energy, transportation, water supply, banking, financial market infrastructure, healthcare, digital infrastructure, and digital service providers (search engines, cloud computing services, and online markets). ISACs and CSIRTs are required to share information on security measures and incidents. In the US, certain information-sharing mechanisms, such as VEP, E-ISAC, and ICT-CERT, have been institutionalized. Japanese industries are also aware of the need to share vulnerability databases (DBs) and are making considerable efforts to do so. Currently, external vulnerability DBs, such as Common Weakness Enumeration (CWE) and CVE, are used as bases for threat analysis. Both DBs are widely provided by US cybersecurity organizations worldwide. JVN, a vulnerability information website jointly operated by IPA and JPCERT/CC in Japan, is certified as compatible with CVE. CVE is a DB of specific software vulnerabilities, and software vulnerabilities are registered when they are discovered. CVE is a DB of specific software vulnerabilities, registered when a software vulnerability is discovered, including vulnerabilities found on a desktop but have not yet manifested in the actual system. Each vulnerability information consists of an outline in natural language and software code. The objective of our work is to develop a basic method for detecting and fixing any software that creates vulnerabilities in one's organization's system when vulnerability information is given from a vulnerability DB. Therefore, we used the security requirement

[1]His contribution of editorial nature contained in this paper about this use of topic model analysis originates mainly when he belonged to Hitachi, Ltd. from February 1, 2017 until January 31, 2021.

analysis tool (TACT) [1] developed by the National Institute of Advanced Industrial Science and Technology (AIST) to collide and analyze documents written in natural language by calculating the similarity between documents using a topic model to find vulnerabilities with high similarity [2]–[5]. In those previous studies, a collision analysis was conducted between two documents using a topic model: one was a document describing the design information and vulnerability characteristics of an automobile in natural language, and the other was a document describing various vulnerabilities not limited to automobiles in natural language. As a result, we obtained a significant effect of deriving new vulnerabilities that match the automobile. However, the effect of increasing or decreasing the database space of the latter vulnerability has not been sufficiently analyzed. Therefore, we developed a tool for vulnerability analysis based on the similarity of natural language descriptions (hereinafter, the proposed tool) using the Python library for topic models. Using the proposed tool, we matched a large-scale vulnerability DB with the section on BROWSER HACKING in [6], which was used in [4] for the attack on Tesla cars, and calculated the similarity. Then, we compared the difference between the data extracted from the large-scale vulnerability DB and the corresponding vulnerability information in the entire data to show the effectiveness of the proposed tool. However, the effectiveness of the proposed tool is deficient if it is tested on only a single comparison target. Therefore, in this study, a new comparison document, the LOCAL PRIVILEGE ESCALATION section of the paper [6], verified the results using the same procedure as the existing study [7]. The rest of this article is organized as follows, Section 2 describes the preliminary description of the techniques and tools used. In Section 3, we introduce the proposed method; in Section 4, we describe the results of document classification using the proposed method. In Section 5, we discuss the results; in Section 6, we summarize the results; finally in Section 7, we describe the issues raised by our proposal.

II. Related Research

A. Large-scale Vulnerability DB

MITRE, Inc. provides Common Vulnerability and Exposure (CVE) [8]. Although there have been DBs of vulnerability information, MITRE proposed CVE in 1999 to uniquely identify vulnerability information. Currently, CVE is linked with many major vulnerability information sites, to provide integrated vulnerability information. In this proposal, we use the csv format data. The data structure consists of Names, Statuses, Descriptions, References, Phases, Notes, and Comments. Of these, Names and Descriptions are extracted and used. Names contain unique IDs, and Descriptions are natural language descriptions of vulnerabilities.

B. Latent Dirichlet Allocation(LDA)

LDA was proposed by David M. Blei in [9]. In LDA, a prior distribution is prepared as a parameter, and the posterior distribution of the parameter is estimated. In this way, the latent topic to which a word belongs and the topic distribution of a document are estimated. The prior distribution is mainly the Dirichlet distribution, and the posterior distribution is generally assumed to be the multinomial distribution. As with pLSA, the graphical model of LDA is shown in Figure 1 for ease of visualization. A description of the symbols used in the graphical model of LDA is given in Table I. The LDA model is generated from a prior distribution α, where the topic distribution θ is a hyperparameter. From the generated θ, the topic distribution z for each word in the document is generated. Another hyperparameter, β, is used to generate the probability of occurrence of a word in a topic. The z and φ can be used to generate w. The difference between pLSA and LDA is that, in LDA, Dirichlet distributions are given as hyperparameters for the word distributions of each document and topic, respectively. As a result, the generalization performance of LDA is often higher than that of pLSA. LDA can be used for various purposes, even when the parameters cannot be obtained analytically. Gibbs sampling is one of the sampling methods used in LDA.

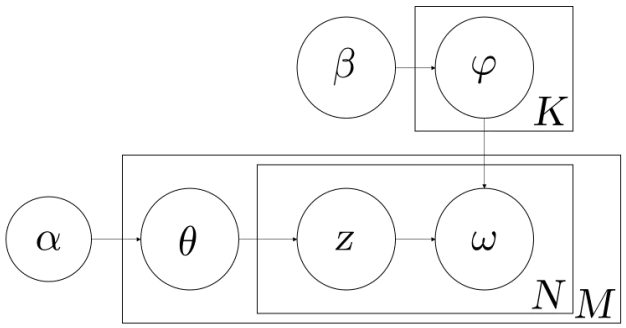

Fig. 1. Graphical model of LDA

TABLE I
SYMBOLIC TABLES IN GRAPHICAL MODELS OF LDA

Symbol	Description
M	Number of documents
N	Number of words in each document
α	Hyperparameters of the prior distribution for each document
β	Hyperparameters of the prior distribution of the word distribution per topic
θ	Topic distribution for each document
z	Topic distribution by word
w	Each word
φ	Word distribution by topic
K	Number of topics

C. Term Frequency - Inverse Document Frequency

TF-IDF is a combination of the two concepts of TF and IDF. TF denotes Term Frequency, the frequency of occurrence of a word. The symbols that appear in this section are shown in Table II. TF can be derived by the formula (1). IDF denotes Inverse Document Frequency, the inverse of the percentage of documents containing a word. It is derived as in equation (2). By multiplying the derived TF by the IDF, the TF-IDF can be obtained.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

$$idf_i = \log \frac{|D|}{|\{d : d \ni t_i\}|} \tag{2}$$

TABLE II
SYMBOLS FOR EQUATIONS(1),(2)

Symbol	Description
i, j, k	Subscript
$n_{i,j}$	Number of occurrences of t_i in document d_i.
$n_{k,j}$	Number of occurrences of a word in a document d_i.
$\|D\|$	Total number of documents
t_i	The ith word
$\|\{d : d \ni t_i\}\|$	Number of documents containing the word T_I

D. Evaluation Index

Many evaluation metrics for topic models have been proposed. Among them are Perplexity and Coherence. Perplexity is the number of branches or alternatives and is expressed as the inverse of probability, i.e., how many alternatives a word is selected from and can be thought of as a predictive metric. In general, the lower the value, the better. Coherence is used as an indicator of the quality of a topic and whether or not the topic is easy for humans to interpret. However, there is no clear definition for Coherence. A human evaluation method, [10], was first proposed for Coherence, and then an automatic evaluation method, [11], was published. In general, the higher the value, the better.

III. How to generate LDA model and calculate similarity between documents

A. Summary of the Proposal

We tested our proposed method on Windows 10 with Python 3.6.9 under Ubuntu 18.04LTS virtual OS using Oracle VM VirtualBox. Vulnerability data were obtained from CVE DB from 1999 to July 25, 2019, removing uncertain information, duplication, 119,479 data points from CVE data from 1999 to July 25, 2019, after removing uncertain information, duplication, integration, and other information. (hereinafter, CVE data). In the LOCAL PRIVILEGE ESCALATION section used in this proposal, the vulnerability mainly used is CVE-2013-6282. In the BROWSER HACKING section, the vulnerability mainly used is CVE-2011-3928. An LDA model is created from the CVE data, and the model is used to assign topic distributions to the two documents. Then, the similarity between the documents is calculated, and the documents are classified. After classification, we examine the number of extracted data and consider the accuracy of each document for comparison. The flow of the process is shown in Figure 2.

B. Creating Data for Comparison

In this proposal, the LOCAL PRIVILEGE ESCALATION and BROWSER HACKING sections in [6] are used as comparison documents. In the LOCAL PRIVILEGE ESCALATION section, the direct expressions "CVE-2013-6282" and "Tesla" were removed from the text. Figure 3 is the content of the LOCAL PRIVILEGE ESCALATION section after the processing used in this proposal. Onward, in this proposal, the LOCAL PRIVILEGE ESCALATION section after processing is called LPE. Figure 4 shows the processed document of the BROWSER HACKING section. The CVE-2011-3928 words, which are direct expressions from the original text, have been removed. Further, in the BROWSER HACKING section, we unified the words with "Google Chrome" since the browsers are different from CVE. This document will be referred to as the BH document onward. For the CVE data, we prepared three different datasets to check the difference in the number of data used. For the conventional tool, we extracted 49 and 50 cases, respectively, from the front and back of CVE-2011-3928 from the large-scale dataset, which we call the small-scale dataset. In addition, the dataset for 2011 is called the medium-scale dataset, and all data downloaded this time is called the large-scale data.

C. Creating an LDA Model with Gensim

In this proposal, we use the proposed tool in which LDA is implemented using Gensim, a Python library. Gensim is licensed under the LGPL and is an open-source library for unsupervised topic modeling and natural language. The proposed tool uses the source code available in [12] and adds the data loading part and the process to remove particles and adverbs from documents called stopwords. The LDA generation process is represented by the dashed line in the Create LDA model step in Figure 2. One of the parameters is the number of topics. The determination of the number of topics is described in Subsection 3.3. The corpus created by the above procedure is a vector representation in the form of Bag of Words (BoW), which represents a document in terms of the number of occurrences of words. In our proposal, we use the corpus weighted by TF-IDF, as described in Section 2, instead of BoW. The procedure for creating a corpus using TF-IDF (hereinafter, TF-IDF corpus) is as follows.

I) Create a TfidfModel using the TfidfModel method in the Gensim models class.

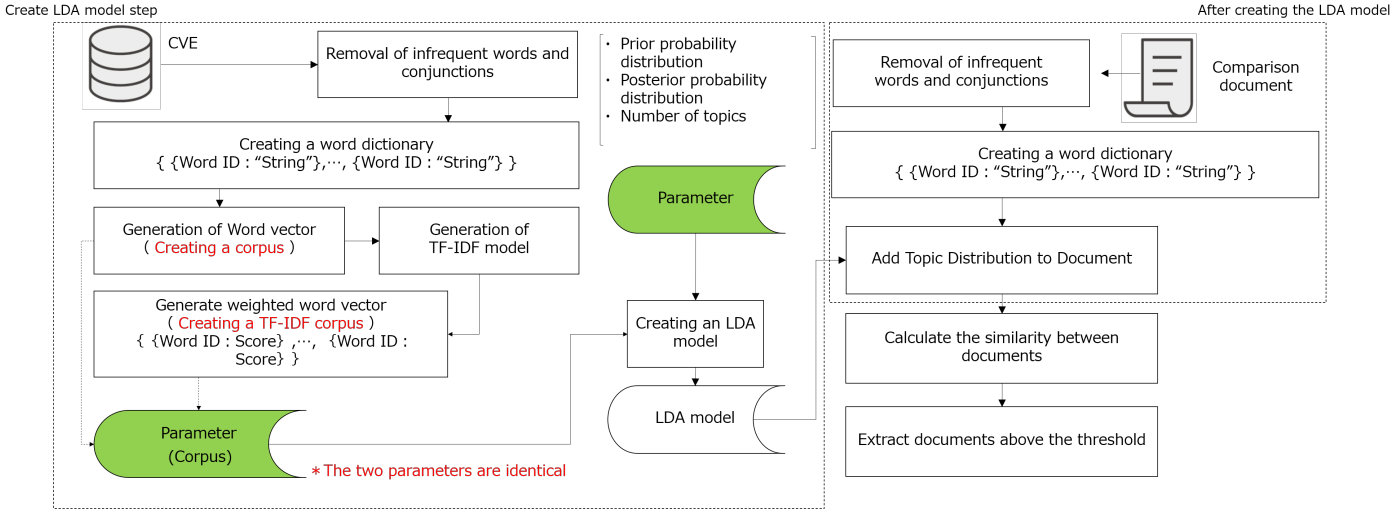

Fig. 2. List of Matching Process Flows

It seems that the Linux kernel version of CID is very old, there is nearly no exploiting mitigations on Linux kernel 2.6.36. we can get the arbitrary read/write in kernel context, it is pretty easy to write an exploit. In our exploit, firstly we patched setresuid() syscall to get the root privilege, and then we invoked reset_security_ops() to disable AppArmor. It ' s obviously that we ' re now in god mode.

Fig. 3. Contents of the LPE document

Since the User Agent of Tesla web browser is "Mozilla/5.0 (X11; Linux) AppleWebKit/534.34 Google Chrome (KHTML, like Gecko) QtCarBrowser Safari/534.34", it can be deduced that the version of QtWebkit is around 2.2.x. In such old version, there are many vulnerabilities in QtWebkit.

[70 lines omitted]

7. Get the address of the JIT memory from JSCell address and JSC::ExecutableBase structure. 8. Write shellcode to JIT memory and execute this JavaScript function. We must say it is difficult to develop a feasible and stable exploit without any debugging method and without QtCarBrowser binary from Tesla CID. However, it was deserved as the final exploit gave us the first shell from Tesla CID and the shell is very stable.

Fig. 4. Contents of the BH document

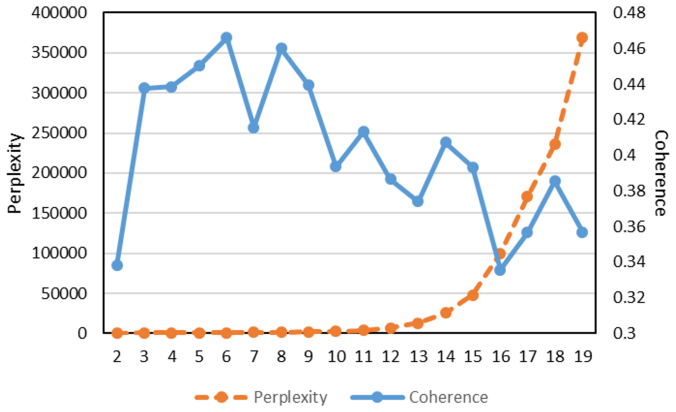

Fig. 5. Graph of Perplexity and Coherence (TF-IDF corpus)

II) The corpus created in "Create LDA model step" (BoW corpus) is passed to TfidfModel to obtain the TF-IDF corpus.

The corpus generated by the above process is used to conduct experiments. This process is performed for the small-, medium-, and large-scale datasets.

D. Determining the Number of Topics

Although it is possible to create an LDA model by following the procedure described in the previous section, it is necessary to determine the number of topics that exist in a document. However, we need to determine the number of topics in a document. We experimentally determined the number of topics in this study. As an example, Figure 5 shows the evaluation of the LDA model on the large-scale dataset with the setting to remove words with a frequency less than 2. Since the LDA model is generated based on CVE data, it is the same for both LPE and BH documents. From Figure 5, the difference is largest at the number of Topic 6, when Perplexity is low and Coherence is high. Therefore, numerically, the number of Topic 6 is the best LDA model for each topic in the graph. In the proposed [7], the number of topics considered to be good based on these evaluation results was used in the creation of the LDA model. In this proposal, the number of topics with a large difference between Perplexity and Coherence was evaluated, and the number of Topic 11 was found to be good. Therefore, the result of Topic 11 was adopted for the LPE document. For the BH document, the number of Topic 7 was used, instead of Topic 6, which was considered to be the best based on [7].

E. Assigning Topic Distribution to Attack Cases

It is necessary to assign topic distributions to the comparison documents as well. For the LDA model, we can use the model of CVE, shown in Subsection 3.2. The flow after creating the LDA model is represented by the dashed line after creating the LDA model section of Figure 2.

F. Calculation of Similarity between Documents

After creating the LDA model and assigning topic distributions to the documents to be compared, we calculate the cosine similarity between the CVE vulnerability information and the documents to be compared. The cosine similarity is calculated using the inner product of the vectors of each document. The following equation 3 is used for this purpose. where $\vec{A}$ is the vector of LDA model and $\vec{B}$ is the vector of documents to be compared.

$$\text{Similarity} = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}| \cdot |\vec{B}|} \tag{3}$$

G. Extraction of Vulnerability Information

CVE-2013-6282 is the vulnerability mainly used in the LPE document used in this study and must be detected in this proposal. Therefore, we set the cosine similarity between CVE-2013-6282 and the LPE document as the threshold. We count the number of vulnerabilities that are above the threshold. In the BH document, the vulnerability mainly used is CVE-2011-3928, so we set the cosine similarity between CVE-2011-3928 and the LPE document as the threshold.

IV. Evaluation

A. Evaluation Results

The evaluation results are summarized in Table III. The conventional tool is the tool used in [4]. First, using the conventional tool, 29 vulnerabilities (14.9% of the total) were identified as similar to BH documents for 200 small-scale data. By contrast, the proposed tool was close to the BoW corpus (15.0%) but far away from the TF-IDF corpus (62.5%) for the same dataset. Next, using the TF-IDF corpus with the proposed tool, 1,514 of 119,479 cases (1.3% of the total) were found when using the BH document and large-scale data. Further, when the proposed tool was used to collate LPE documents with the large-scale data, the number of collated documents was 5,180 of 119,479 (4.3% of the total). Although the LPE documents were not validated with the BoW corpus, we could see from the BH documents that the accuracy tended to increase with the number of documents in both corpora. Notably, the results of the BoW corpus were significantly worse for the medium-sized dataset. The TF-IDF corpus, which was the best corpus, was also considered to be large in terms of the number of cases-1,514 cases.

V. Consideration

The results of the TF-IDF corpus in the abovementioned evaluation results were much worse than those of the conventional tools for the small-scale dataset. This might be because the TF-IDF method, which is used to weight the corpus, may not be compatible with this scale of data. The results of the TF-IDF corpus improved significantly as the number of cases increased, suggesting that this assumption is not wrong. In the BoW corpus, the results were significantly worse for the medium-sized dataset, which might be due to poor parameter selection. As can be seen from the results in the TF-IDF corpus in Table III, the number of topics increased with the amount of data. Therefore, the number of topics might be low in the BoW corpus as well. However, in the BoW corpus, the candidate number of topics is five, because a large number of topics may result in too few words belonging to them. However, it is still questionable whether this is appropriate. In [7], we thought that the short description of CVE would affect the accuracy, but the verification experiment in the proposal showed that it affected the accuracy if the document to be compared was short. This could be seen in the results of the TF-IDF corpus for BH and LPE documents using the large-scale dataset. Therefore, both data used for collation should be somewhat long individually.

VI. Conclusion

Compared with the results of collating BH documents, the results of collating LPE documents were less accurate. This may be because the LPE documents were shorter than the BH documents. To improve the accuracy through the two proposals, it is necessary to increase the amount of data and introduce a new weighting method. Therefore, we would like to add preprocessing, such as normalization of documents and weighting of methods different from TF-IDF (e.g., Okapi BM25). From the current results, we would like to introduce a method to obtain results in a different direction, such as extracting attack trends from the current results by cluster analysis. Specifically, we would like to use MITRE's CWE, which classifies vulnerabilities according to their lineage, and cluster analysis of the collated data to present vulnerability trends.

Trademarks

Acknowledgment

Sven's contribution contained in the paper is done when he belonged to Hitachi, Ltd. before he joined SECOM Co., Ltd. in August 2021.

References

[1] Kenichi Handa, Hitoshi Ohsaki, and Izumi Takeuti, "Security Requirements Analysis Supporting Tool: TACT," In: Proceedings of the Information Processing Society of Japan (IPSJ) SIGSE Winter Workshop 2017 in Hida-Takayama (WWS2017), pp. 5–6, 2017.

TABLE III
NUMBER OF CASES FOR EACH DATA

Tool name	Usage data	Comparison document	Number of topics	Total number of cases	Applicable number	Applicable ratio (%)
Conventional Tools	small data	BH document	Unknown	200	29	14.5
proposed tool(BoW Corpus)	small data	BH document	5	200	30	15.0
	Medium data	BH document	6	4398	4118	93.6
	large data	BH document	6	119479	20890	17.5
proposed tool (TF-IDF Corpus)	small data	BH document	2	200	125	62.5
	Medium data	BH document	6	4398	2628	59.8
	large data	BH document	7	119479	1514	1.3
	large data	LPE document	11	119479	5180	4.3

[2] Katsuyuki Umezawa, Yusuke Mishina, Kenji Taguchi, and Kazuo Takaragi, "A Proposal of Threat Analyses using Vulnerability Databases," In: Proceedings of the 2018 Symposium on Cryptography and Information Security (SCIS 2018), pp. 1C2–6, Jan. 2018.

[3] Yusuke Mishina, Kazuo Takaragi, and Katsuyuki Umezawa, "A Method of Threat Analysis for Cyber-Physical System using Vulnerability Databases," In: Proceedings of the 18th Annual IEEE Symposium on Technologies for Homeland Security (HST2018), pp. 1–7, Oct. 2018.

[4] Katsuyuki Umezawa, Yusuke Mishina, Sven Wohlgemuth, and Kazuo Takaragi, "Threat analyses using vulnerability databases –Practical use of topic models and reuse of past analysis results–," In: Proceedings of the 2019 Symposium on Cryptography and Information Security (SCIS 2019), pp. 2F3–3, Jan. 2019.

[5] Katsuyuki Umezawa, Yusuke Mishina, and Kazuo Takaragi, "Threat analyses using vulnerability databases – Possibility of utilizing past analysis results–," In: Proceedings of the 19th Annual IEEE Symposium on Technologies for Homeland Security (HST2019), pp.1–6, Nov. 2019.

[6] Sen Nie, Ling Liu, and Yuefeng Du, "FREE-FALL: HACKING TESLA FROM WIRELESS TO CAN BUS", Briefing,B lack Hat USA 2017, Jul. 2017.

[7] Hiroki Koyanagi, Kazuo Takaragi, Yusuke Mishina, Sven Wohlgemuth, and Katsuyuki Umezawa, "Threat AnalysisMethod using Vulnerability Database: Attack case and large-scale vulnerability DB matching by topic model analysis using LDA classifier and cosine similarity". 2020, 2020-CSEC-88(38), pp. 1–6. (in Japanese)

[8] MITRE Corporation, "CVE - Common Vulnerability and Exposure", https://cve.mitre.org/ (Last accessed at: 2020-01-15).

[9] David Meir Blei, Andrew Yan-Tak Ng, and Michael Irwin Jordan, "Latent Dirichlet Allocation", Journal of Machine Learning Research (2003), pp. 1107–1135.

[10] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei, "Reading Tea Leaves: How Humans Interpret Topicmodels", Advances in NIPS, pp. 288–296, 2009.

[11] Newman, D., Lau, J. H., Grieser, K. and Baldwin, T., "Automatic Evaluation of Topic Coherence", pp. 100–108, 2010.

[12] Latent Dirichlet Allocation (LDA) Yuruhuwa nyumon - arabikinisshi (in Japanese), https://abicky.net/2013/03/12/230747 (Last accessed at 2020-01-15).

[13] Teh, Y. W.; Jordan, M. I.; Beal, M. J.; Blei, D. M., "Hierarchical Dirichlet Processes", Journal of the American Statistical Association. 101 (476), pp. 1566–1581, 2006.