Threat analyses using vulnerability databases –Possibility of utilizing past analysis results–

Katsuyuki Umezawa
Depertment of Information Science
Shonan Institute of Technology
Kanagawa, Japan
umezawa@info.shonan-it.ac.jp

Yusuke Mishina, Kazuo Takaragi
Cyber Physical Security Research Center (CPSEC)
National Institute of Advanced Industrial Science and Technology (AIST)
Tokyo, Japan
{yusuke.mishina, kazuo.takaragi}@aist.go.jp

***Abstract*—We propose a threat analysis method utilizing topic model analysis and vulnerability databases. The method is based on attack tree analysis. We create an attack tree on an evaluation target system and some attack trees of known vulnerability. And we combine the two types of attack trees to create more concrete attack trees, that is, attack trees of target system that contain vulnerabilities. Specifically, matching processing of attack trees nodes written in natural language was automated using latent Dirichlet allocation and cosine similarity. The concrete attack tree enables us to calculate the probability of occurrence of a safety accident. In this paper, we show that our proposed method can use the results of past threat analysis for the next one using the case of Tesla model S, Jeep Cherokee and IRB140 industrial robot.**

***Index Terms*—Threat Analysis, Vulnerability Information, Attack Tree, Topic Model Analysis, LDA.**

I. Introduction

The MITRE Corporation in the US provides several specification for vulnerability databases. In CVE (Common Vulnerability and Exposure) [1], individual software vulnerabilities are stored in a database. In CWE (Common Weakness Enumeration) [2], common vulnerabilities are cataloged focusing on the cause of the vulnerability. Furthermore, CAPEC (Common Attack Pattern Enumeration and Classification) [3] is a database classified by attack pattern.

The scientific literature related to safety analysis using Fault tree (FT) is mature today [4]. On the other hand, in security analysis, the complexity of analysis is significantly increased. Elaborate attacks occur with multiple combinations of those vulnerabilities. Furthermore, it is not easy to create Attack tree (AT) that comprehensively captures their possibilities.

We have focused on such problems and proposed a threat analysis method using a vulnerability DB as a practical approach [5] [6]. First, we assumed that many attacks were imitations or minor changes of known attacks. Therefore, we can say 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 [5] [6]. However, the matching method utilized was not specified. We evaluated the feasibility of this unspecified matching process using a topic model analysis method [8].

In this paper, we show that we can use the case of attack trees created in the past threat analysis for the next analysis in order to show that "the continuous application of our proposed approach can help to reduce risks".

In Section 2, we summarize the threat analysis method we proposed in [5] and [6]. In Section 3, we verify the feasibility of matching attack cases to vulnerability DBs and show the result. We apply the proposed algorithm to the case of Tesla model S, Jeep Cherokee and IRB140 industrial robot in Section 4. In Section 5, we describe a consideration about three cases. Section 6 concludes this paper by summarizing the key points and providing an outlook on future activities.

II. Threat analysis using vulnerability databases

This section presents a summary of our proposed method [6]. 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.

The proposed method is equivalent to creating a first_AT such as Figure 3 or 4 from the vulnerability DB, and creating a second_AT such as Figure 5 or 6 from the design information of the system to be analyzed.

A. Creating vulnerability model information

The MITRE Corporation has published several forms of vulnerability databases [1] [2] [3]. However, it is difficult to create AT for a concrete target (for example, for a connected car) simply by referring to these DBs. We will create AT with reference to existing attack case literature or reports, etc. Thus, let AT obtained from the existing vulnerability DB and existing attack report be called the first_AT. This first_AT is hierarchically drawn into a top event, a plurality of intermediate events, and a bottom event. One first_AT is created for each vulnerability.

B. Proposal of component database

In embedded systems such as automobiles and IoT devices, it is not necessary to install the existing software as it is but to incorporate the necessary lower level components as needed. On the other hand, a vulnerability database such as CVE describes vulnerability information for certain software, but it does not describe information on subordinate components embedded in the software. Therefore, a correspondence table between the software version and the version of the lower-level component used internally by the software would help. This makes it easy to check vulnerability information at the manufacturing stage of embedded devices such as IoT devices.

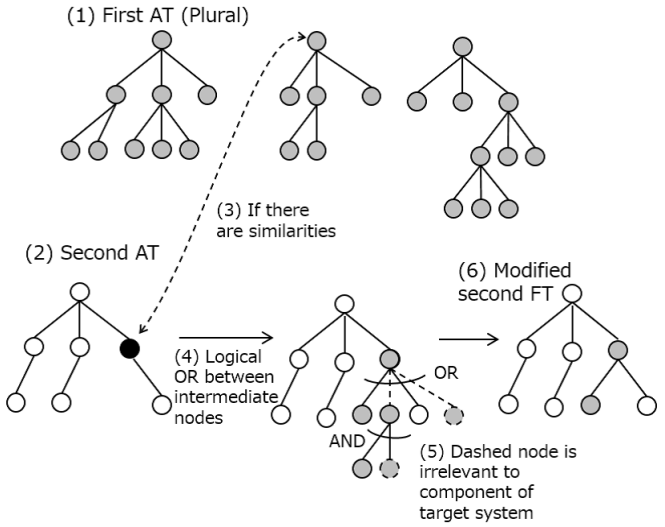

Fig. 1. Threat analysis algorithm (cited from [6])

C. Threat analysis algorithm

The threat analysis algorithm 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 shown.

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 1 (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 1 (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 1 (3)). If this is the case, the first_AT is temporarily added to the second_AT (Figure 1 (4)). An 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 1 (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.

III. 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. Specifically, we matched the text of each chapter of Tesla's attack paper [11] with the description of the vulnerability database CVE. 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 using Latent Dirichlet Allocation (LDA) and Cosine similarity.

We translated the paper [11] 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.

B. Analysis result

The result of matching each section of the paper to each CVE vulnerability is shown in Figure 2. When we click on 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. The dashed area in the right pane is the description of CVE-2011-3928, one of the vulnerabilities identified as similar to BROWSER HACKING section on the left.

As a result of similarity evaluation by LDA, 22 out of the 500 CVEs were judged to be similar to the sentence in the BROWSER HACKING section of Tesla's attack paper, and one of the 22 was an appropriate CVE-2011-3928. Regarding CVE-2013-6282, a similar result was obtained by matching the sentence of LOCAL PRIVILEGE ESCALATION section with that of CVE, in this case 23 out of the 500.

The basic idea is to widely consider suspiciously suspects to possibly prevent false negatives (missing of criminals). The number twenty of possibly false positives can be processed at

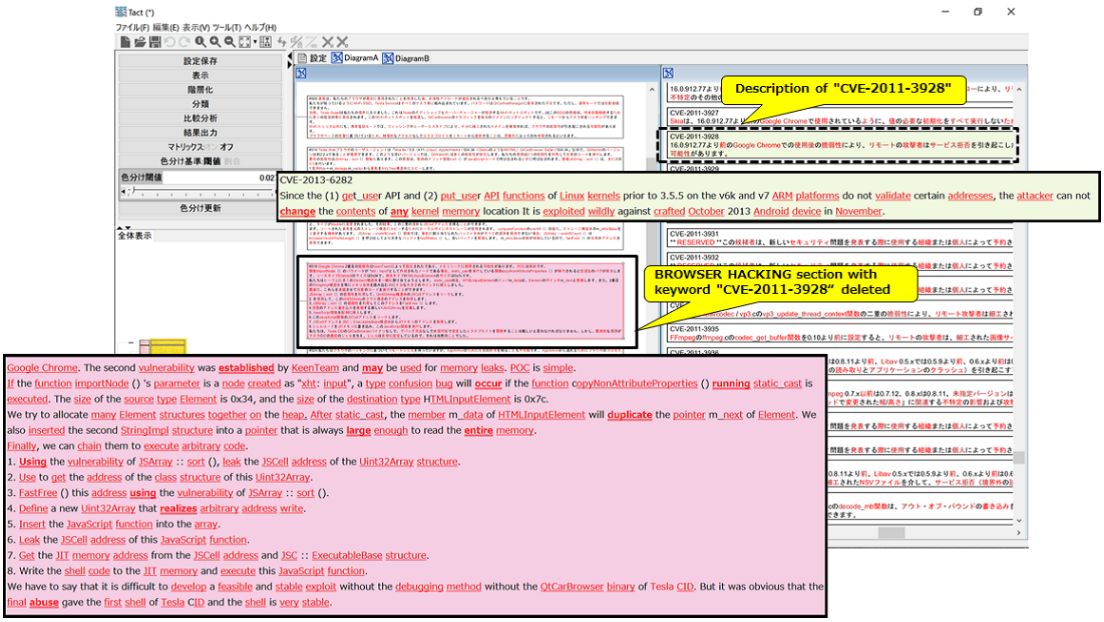

Fig. 2. Matching attack cases to vulnerability DBs

the time of the second attack tree generation without much effort in this method.

From this result, it was found that by using LDA, it is possible to find an appropriate CVE from a huge number of CVEs described in natural language. By applying this method, it was found that the nodes of the first_AT and the nodes of the second_AT written in natural language can be matched.

IV. Application of Proposed Method to Actual Case

In this section, we will use the cases of Jeep Cherokee, IRB140 industrial robot and Tesla model S to show that we can use the attack tree created in the past threat analysis for future analysis.

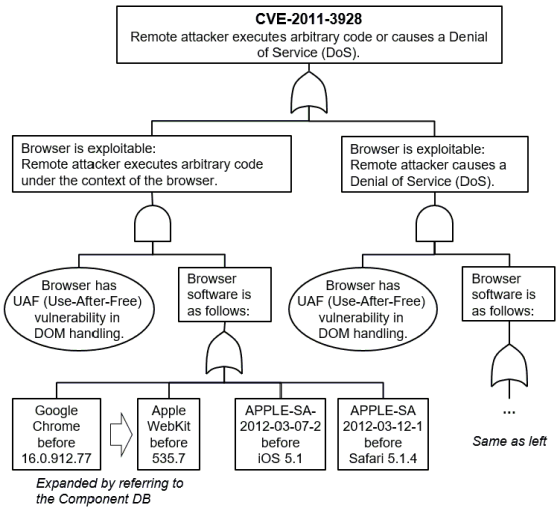

Fig. 3. The first_AT generated from CVE-2011-3928(cited from [6])

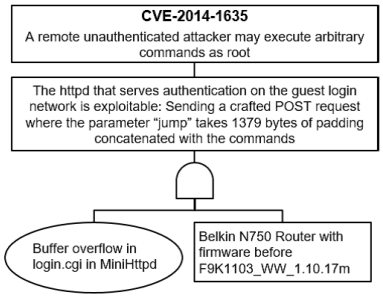

Fig. 4. The first_AT generated from CVE-2014-1635

A. Creating first_AT

First, a first_AT is created on the basis of the vulnerability DB. We must create first_ATs for all vulnerabilities. In this case, we created first_ATs for CVE-2011-3928 and CVE-2014-1635 as shown in Figure 3 and Figure 4.

B. Apply to the case of Jeep Cherokee

First, we created a second_AT for the case of Jeep Cherokee [9]. The second_AT we created is shown in Figure 5. The dashed part of the figure 5 is the subtree joined as a result of the matching with the first_AT shown in Figure 4. Also, the dashed-dotted part of the figure is the part that will be matched and reused when this second_AT will be used as the first_AT for the next analysis (the case of Tesla).

C. Apply to the case of IRB140 industrial robot

Next, we created a second_AT for the case of IRB 140 industrial robot [10]. The second_AT we created is shown in Figure 6. In this case, CVE is not referred to, but refers to another vulnerability database (a vulnerability database

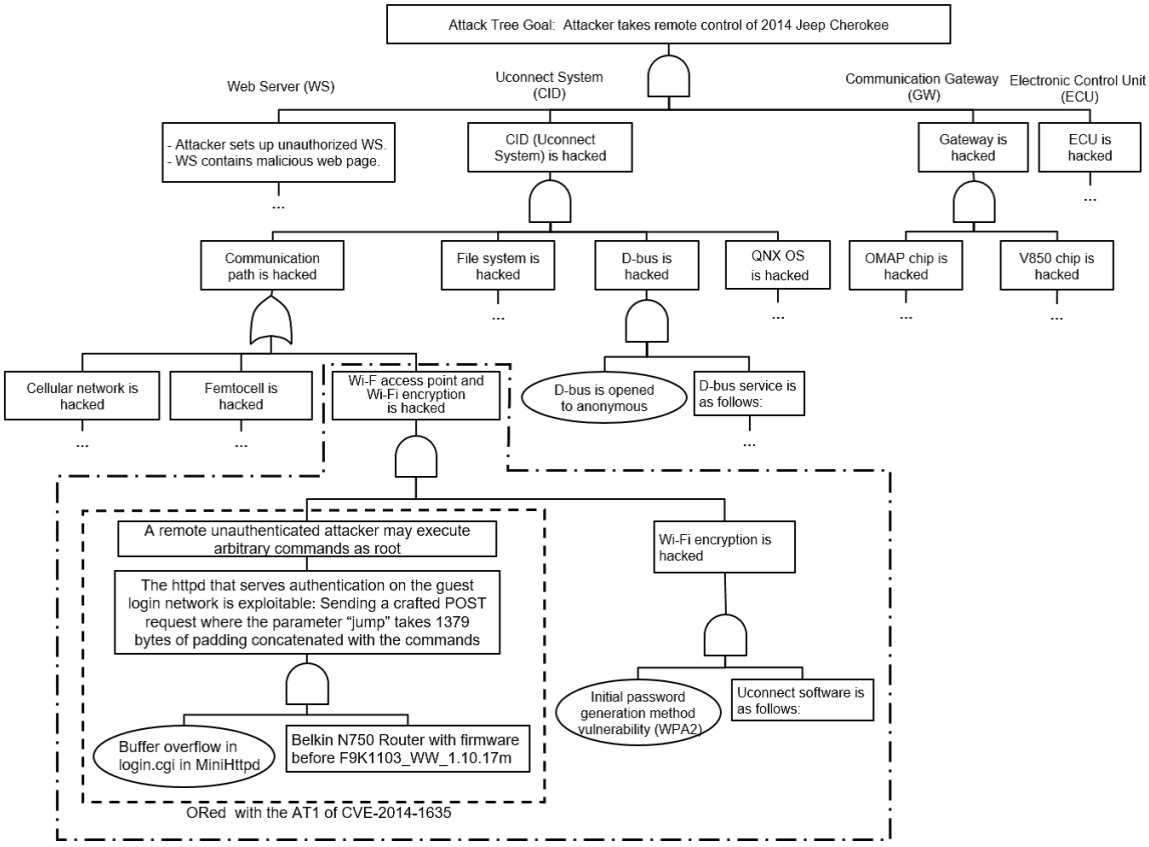

Fig. 5. The second_AT generated from the case of Jeep Cherokee

for robots provided by ABB). It is effective to generate the first_AT from various vulnerability DBs as well as CVE though we omitted this time. The dotted part of the figure is the part that will be matched and reused when this second_AT will be used as the first_AT for the next analysis (the case of Tesla). In addition, this time, we performed a matching check with human power, and could not conduct an exhaustive check because of the amount of work. In the future, it will remain as an issue to conduct an exhaustive match between the IRB140 industrial robot and Tesla model S by computer processing.

D. Apply to the case of Tesla model S

Finally, we created a second_AT for the case of Tesla model S [11]. The second_AT we created is shown in Figure 7. The dashed part of Figure 7 is the subtree joined as a result of the matching with the first_AT shown in Figure 3. Also, the dashed-dotted part is the combined subtree as a result of using the second_AT generated in the case of Jeep Cherokee as the first_AT of this threat analysis. Furthermore, the dotted part is the combined subtree as a result of matching the second_AT generated in the case of the IRB 140 industrial robot.

The actual attacked vulnerability in the Tesla case is that for Wi-Fi, the SSID and password were hard-coded in the CID. Also, the password is hard-coded in the firmware for bypassing the GW user authentication. Both attack methods are elementary. It is important to note that the actual Tesla attack method was included in the subtree joined from the Jeep and Robot cases. Thus, the feasibility of "the second_AT is available as the first_AT in future threat analysis" in the final step of the proposed algorithm was shown.

V. CONSIDERATION

In all three cases, arbitrary commands can be sent illegally by repeating (a) acquisition of access path, (b) acquisition of access permission, and (c) acquisition of system root permission. As a relation between robot and automotive, we showed that the factors of attack on IRB 140 robot such as "Wi-Fi access point and Wi-Fi encryption is hacked" and "Bypass the User Authentication System" are also applicable to Tesla under certain conditions.

In addition, as the relationship between cars, although the system configuration is more complex as compared to robots, the above procedure is repeated for both Tesla and Cherokee, reaching the GW, rewriting the GW firmware, and sending a message to CAN. In the case of Cherokee, the A850 chip acts as a Gateway that handles messages to CAN centrally. The A850 chip is not separated by a network. However, the attacker repeatedly acquired the access path to the Uconnect System, acquired the access permission, acquired the root permission of the system (it was actually possible to execute anonymously),

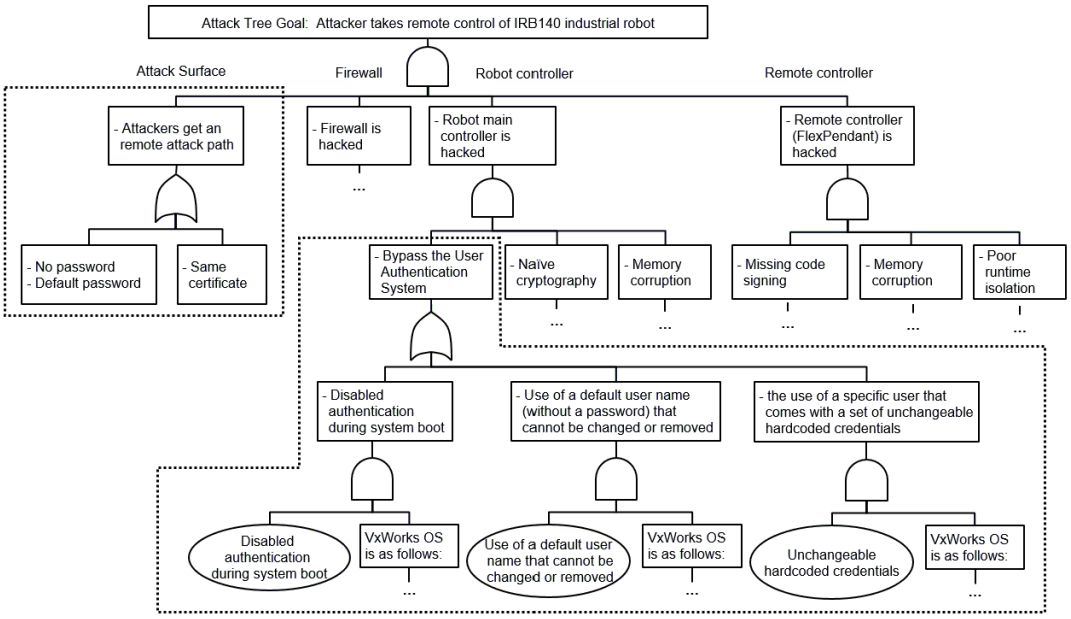

Fig. 6. Second_AT generated from the case of IRB140 industrial robot

and reached the A850 (Gateway). Then, the A850's firmware was rewritten so that messages could be transferred to CAN. As in the case of Tesla, the flow of intrusion is the same as described above, and it is also the same as finally rewriting Gateway's Firmware to send a message to CAN. It is thought that both Tesla and Cherokee prevented the attack by making it impossible to rewrite the firmware of chip that controls the message to CAN.

In the field of AI, there are successful cases of accident occurrence prediction assuming "co-occurrence". Under the assumption that this co-occurrence is seen in attacks on connected cars or robots with similar functional parts, we were able to confirm that. We believe that analysis time and effort will be significantly reduced compared to the case where co-occurrence is not assumed.

VI. Conclusion

We explained our proposed threat analysis method utilizing topic model analysis and vulnerability databases. We also showed that LDA and cosine similarity can be used to automatically match nodes written in natural language. Furthermore, we showed that we can use the second_AT created in the past threat analysis as the first_AT in the next analysis and apply our proposed method gradually and continuously to help reduce the risk.

Acknowledgment

This work was supported by the Cabinet Office (CAO), Cross-ministerial Strategic Innovation Promotion Program (SIP), "Cyber-Security for Critical Infrastructure" (funding agency: NEDO).

References

[1] MITRE Corporation, "CVE - Common Vulnerability and Exposure," https://cve.mitre.org/
[2] MITRE Corporation, "CWE List - Common Weakness Enumeration," https://cwe.mitre.org/data/
[3] MITRE Corporation, "CAPEC - Common Attack Pattern Enumeration and Classification," https://capec.mitre.org/
[4] I. N. Fovino et al., "Integrating cyber attacks within fault trees," Reliability Engineering and System Safety 94 (2009) p.p.1394–1402.
[5] Katsuyuki Umezawa, Yusuke Mishina, Kenji Taguchi and Kazuo Takaragi, "A Proposal of Threat Analyses using Vulnerability Databases," Proceeding of the 2018 Symposium on Cryptography and Information Security (SCIS2018), pp.1–8, 2018.
[6] 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), 2018.
[7] Katsuyuki Umezawa, Yusuke Mishina, Kazuo Takaragi and Sven Wohlgemuth, "Utilization of topic model analysis tool for threat analysis method using vulnerability database," Proceedings of the 2018 IEICE Society Conference, p.32, Sept. 2018.
[8] K. Umezawa, Y. Mishina, S. Wohlgemuth, and K. Takaragi, "Threat Analysis using Vulnerability Databases – Matching Attack Cases and Vulnerability Database by Topic Model Analysis –," Proceeding of the Third International Conference on Cyber-Technologies and Cyber-Systems (CYBER 2018), p.p. 74-77, Nov. 2018.
[9] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," Briefing, Black Hat USA 2015.
[10] D. Quarta, M. Pogliani, M. Polino, A.M. Zanchettin, and S. Zaner, "Rogue Robots: Testing the Limits of an Industrial Robot's Security," Briefing, Black Hat USA 2017, July 2017.
[11] S. Nie et al., "FREE-FALL: HACKING TESLA FROM WIRELESS TO CAN BUS," Briefing, Black Hat USA 2017, July 2017.
[12] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation", in Journal of Machine Learning Research, 2003, pp. 1107-1135.
[13] Kenichi Handa, Hitoshi Ohsaki and Izumi Takeuti, "Security Requirements Analysis Supporting Tool: TACT," Proceeding of the Information Processing Society of Japan (IPSJ) SIGSE Winter Workshop 2017 in Hida-Takayama (WWS2017), pp.5–6, 2017.
[14] Google translater https://translate.google.com/

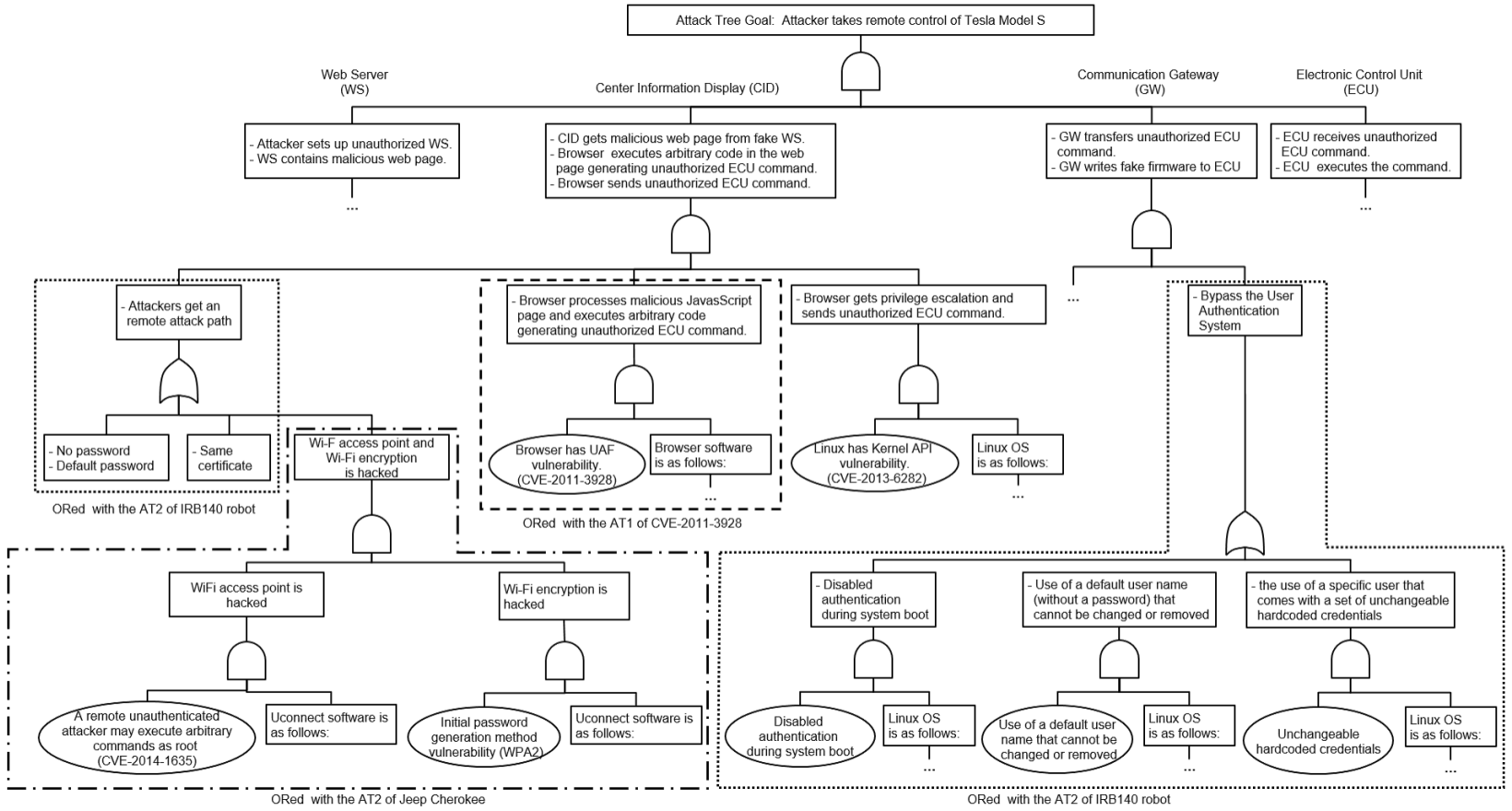

Fig. 7. Second_AT generated from the case of Tesla model S