IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2017.Doi Number*

# A Structured Cyber Attack Representation Model based on the Diamond Model and Adversary States

**JUNYA FUJITA[1], Member, IEEE, TAKASHI OGURA[1], Non-Member, IEEE, KAZUYA OKOCHI[2], Non-Member, IEEE, NORITAKA MATSUMOTO[1], Member, IEEE, KENJI SAWADA[3], Member, IEEE, and OSAMU KANEKO[3], Member, IEEE**

[1]Hitachi, Ltd. Research & Development Group, 7-1-1, Omika-cho, Hitachi, Ibaraki 319-1292, Japan

[2]Hitachi, Ltd. Service & Platform Business Unit, 27-18, Minami-Ohi, Shinagawa, Tokyo 140-8572, Japan

[3]The University of Electro-Communications, 1-5-1, Chofugaoka, Chofu, Tokyo 102-0076, Japan

Corresponding author: Junya Fujita (e-mail: junya.fujita.so@hitachi.com).

**ABSTRACT** Threat modeling is essential for selecting effective cybersecurity measures to secure industrial control systems (ICS). However, in-depth cybersecurity knowledge is required to perform the process. The results can vary from one security analyst to the next, particularly concerning prioritizing cyber attack scenarios. Cyber attack scenarios should be represented using a structured model to reduce the dependency on analysts. This study proposes a new structured model that expresses cyber attack scenarios for ICSs. The suggested model is based on the Diamond model and can express adversarial behavior against a targeted system in a structured manner. As a result, it can represent structured cyber attack scenarios and reduce the dependency on analysts in threat modeling. This paper describes the details of the proposed model to eliminate personal judgment in threat modeling and presents the evaluation of the validity and effects of the model.

**INDEX TERMS** Industrial Control System, Cyber Security, Threat modeling, Structured scenario, Diamond model, ATT&CK

## I. INTRODUCTION

Industrial Control Systems (ICS) have made it possible to build and operate systems more efficiently through the active introduction of information technology. FIGURE 1 shows a typical system structure of ICS. The contemporary ICSs are interconnected through networks comprised of diverse devices, which subsequently interface with the broader enterprise infrastructure. However, ICS are now having to deal with cybersecurity risks. For some control targets, an ICS can have significant safety and business implications. There have also been reports of sophisticated attacks by organization-level threat actors such as Stuxnet, Industroyer, and TRITON that intentionally aim to disrupt society by destroying ICSs [1][2][3]. The security of ICS, including protection against these sorts of advanced attackers, remains an issue.

It is recommended that security measures for ICS should be designed based on threat modeling [4][5], which is a procedure for ascertaining and prioritizing potential security threats in an ICS, and then clarifying the requirements for countermeasures to detect and protect against high-priority threats, or minimize the damage caused by these threats if they occur in practice. In general, threat modeling can be implemented at the system design phase or even when the actual system cannot be accessed. In addition, its results are helpful to identify necessary security features to reduce risks in the system. As another example, threat modeling can be combined with asset management [6].

On the other hand, threat modeling has to be performed by engineers with specialized knowledge and methodologies, which raises issues such as labor costs and the effort needed to establish practical methods capable of delivering reasonable results [7][8].

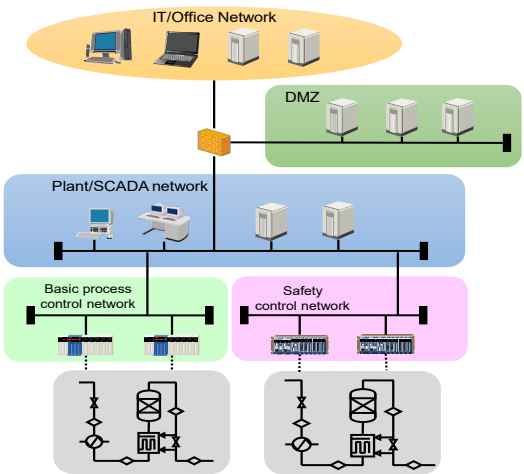


**FIGURE 1. The typical structure of ICS**

Against this background, various reports have described ways of making threat modeling for ICS more efficient. According to one proposed approach, for example, if an attack targets a particular event such as disabling safety functions, then this event is associated with an existing threat model such as Cyber Kill Chain [9], OWASP Attack Surfaces [10] or CAPEC (Common Attack Pattern Enumeration and Classification) [11], and lower-order threat events related to this event are mechanically identified to automatically create an attack tree [12]. This method can improve the reusability of threat models, which makes it possible to automatically generate an exhaustive attack tree that leads to the ultimate attack target. In another related study, a method has been proposed for decomposing threat expressions into their constituent elements and identifying patterns by focusing on the context (Who, What, When, Where, and Why) of threat expressions [13]. These methods make it possible to extract threat events in an exhaustive and reusable format.

On the other hand, prioritizing identified threat events is still problematic. Threat events are prioritized based on risk factors, i.e., the severity of their impact should they occur and their likelihood of occurring actually [5]. In threat modeling, the likelihood should be evaluated objectively based on case studies. Still, the metrics defined by recognized standards need to be more specific, and judgments tend to vary depending from one security analyst to the next. For example, IEC 62443-2-1:2010 defines the indicators shown in TABLE I [14]. When the likelihood of a certain threat event is determined according to the vague metrics shown in this table, different analysts tend to estimate different likelihood values.

Other studies have proposed likelihood evaluation methods based on standards such as CVSS (Common Vulnerability Scoring System) and CWSS (Common Weakness Scoring System) [13][15][16][17], where the likelihood of a threat event is ranked by a scoring system based on the metrics of the CVSS and CWSS vulnerability scoring systems [18][19]. Security analysts can evaluate the likelihood of threat events based on clearer criteria than those of TABLE I.

TABLE I
AN EXAMPLE OF A LIKELIHOOD SCALE [14]

| Likelihood | Description |
|---|---|
| High | A threat/vulnerability whose occurrence is likely in the next year |
| Medium | A threat/vulnerability whose occurrence is likely in the next ten years |
| Low | A threat/vulnerability for which there is no history of occurrence and for which the likelihood of occurrence is deemed unlikely |

For example, TABLE II shows the evaluation criteria of AC (attack complexity), which is a CVSS metric. The details of threat events must be clarified until it is obvious which of the criteria shown in TABLE II is applicable.

TABLE II
THE DEFINITION OF AC IN CVSS

| AC | Description |
|---|---|
| High | ・It is necessary to collect information in a suspicious manner, such as privilege escalation or obfuscation, before attacking<br>・Attack is possible only when the target system has specific settings |
| Medium | ・Some information needs to be collected before the attack<br>・Specific settings with non-standard are required to exploit |
| Low | ・Systems can be attacked without any special conditions |

To evaluate the likelihood of a threat event based on these metrics in a way that can be agreed upon by the parties concerned, threat events must be appropriately represented so that the basis for judgment can be clearly understood. In other words, the more abstract the expression of a threat event is, the more difficult it is to judge how likely it is.

In this study, we aim to establish a method whereby threat events can be represented in a reusable format so that threats requiring priority action can be dealt with preferentially without relying on personal judgment.

Personal judgment in threat modeling is contingent upon variations stemming from the assessor's knowledge, the ambiguity in the representation of threat events, and the criteria for evaluating high-priority threats. Within personal judgment in threat modeling,

the most predominant factor is the ambiguity in representing threat events. The more ambiguous the elements that require judgment are, the more assessors rely on their subjective perceptions to fill in information gaps in identifying possible threat scenarios and determining the likelihood of each scenario. Consequently, the influence of personal judgment becomes notably apparent in the results of threat modeling.

To solve this issue, it is essential to rationalize and specify the representation of threat scenarios and to establish a logic for determining priority threat scenarios. To achieve this objective, we have studied ontological models for representing threat events and developed a model structurally representing an adversary's behavior and intermediate states. In this study, we have designed the logical structure of mechanisms by which threats occur in ICS, focusing on a method to determine priority threat events without relying on personal judgement in threat modeling.

Initially, we have designed the logical structure of the mechanisms leading to threats and evaluated the validity and effects of its representation models, which are presented in Sections II, III-A, B, C. Subsequently, we have attempted to quantify security risks based on the proposed method and assessed the feasibility of eliminating personal judgement in determining the "likelihood" (presented in Section III-D). Finally, we have evaluated the computational impacts of our proposed method when either the attacker or the target system becomes more complex (presented in Section III-E).

This article delves into the design, assessment, and discourse of the proposed model. Section II of this paper describes the details of the proposed model, section III presents the evaluation results obtained with this model including benchmarking, and section IV presents our conclusions and areas for future study.

## II. PROPOSED MODELS

### A. THE BASE MODEL

An adversary targeting an ICS will generate various events from the initial intrusion to the final attack before being able to succeed in the desired attack. In order to analyze this attack process, the Diamond model has been developed [20]. As shown in FIGURE 2, this model represents attack events in terms of four core elements (adversary, capability, victim, and infrastructure) and their interrelationships. In this model, "adversary" represents the person or group responsible for the actual attack, "capability" represents the attack methods and tools possessed by the adversary, "victim" represents the target of the attack, and "infrastructure" represents the system structures between the adversary and the victim (i.e., the network hardware and other systems through which the attack is made). The lines between these core elements indicate the relationships between events that occur during an attack.

The relationships indicated in FIGURE 2 show that the adversary uses the infrastructure and the adversary's capability when attempting to attack the victim. This relationship is based on the principle that whenever an adversary attacks a victim, it will always cause an event associated with infrastructure or capability [20]. Since the Diamond model is a model for analyzing an adversary's post-intrusion behavior, it is suitable for modeling its preconditions and attack processes.

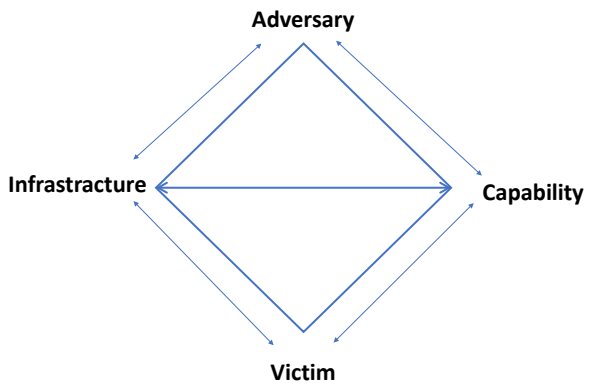


**FIGURE 2. The Diamond model**

In this study, we have examined a model of the attack process based on the Diamond model.

### B. ADVERSARIAL BEHAVIOR MODEL

We studied a structural model that consists of the parties involved in the adversary, the target system, and those behavioral models that represent the adversary's behavior on the target system during the attack based on the Diamond model.

#### (a) THE STRUCTURAL MODEL

The structural model of adversarial behavior consists of a threat actor ($ThreatActor$) who attempts to penetrate the ICS, and the ICS targeted by the attack ($TargetSystem$). The conditions of this model are as follows.

(Cond. i) $TargetSystem$ is the ICS model of the attack target, which comprises the constituent components of the ICS (server equipment, PLCs, etc.), the network that shows the logical connections between the components, and the physical locations where these components are situated (hereinafter referred to as the physical area).

(Cond. ii) The $Victim$ exists on $TargetSystem$. Also, the adversary's $Infrastructure$ includes elements of

$TargetSystem$.

(Cond. iii) The $ThreatActor$ has an explicit attack objective ($Goal$), and there are one or more adversaries ($Adversary$) with a given attack capability ($Capability$).

(Cond. iv) The $Goal$ comprises a $Victim$ and a set of attack actions.

Equations (1) to (3) show the results of formulating the above conditions based on set notation.

$$TargetSystem = \{Component, PhysicalArea, Network\} \cdots\cdots (1)$$

$$Victim = \{v \mid v \in Component\} \cdots\cdots (2)$$

$$Infrastructure = \{ C_{inf}, P_{inf}, N_{inf} \mid C_{inf} \in Component \cup Component_{Adv}, P_{inf} \in PhysicalArea \cup PhysicalArea_{Adv}, N_{inf} \in Network \cup Network_{Adv} \} \cdots\cdots (3)$$

Equation (1) corresponds to (Cond. i), whereby $TargetSystem$ is treated as a family of sets consisting of a component group, a physical area group, and a network group that from which the ICS is constructed. In Equation (1), $Component$ is a finite set whose elements are the constituent components of $TargetSystem$. For example, if the constituent elements of $TargetSystem$ are $\mathrm{comp1}, \mathrm{comp2}, \mathrm{comp3}, \ldots$, then it is expressed as $Component = \{\mathrm{comp1}, \mathrm{comp2}, \mathrm{comp3}, \ldots\}$. Similarly, $PhysicalArea$ is a finite set of the constituent physical area elements of $TargetSystem$, and if the constituent physical area components of $TargetSystem$ are $\mathrm{area1}, \mathrm{area2}, \mathrm{area3}, \ldots$, then this is expressed as $PhysicalArea = \{\mathrm{area1}, \mathrm{area2}, \mathrm{area3}, \ldots\}$. Likewise, $Network$ is a finite set of the constituent network elements of $TargetSystem$, e.g., $Network = \{\mathrm{nw1}, \mathrm{nw2}, \mathrm{nw3}, \ldots\}$.

Equations (2) and (3) correspond to (Cond. ii). In equation (2), the set of attack targets $Victim$ consists of the $Component$ elements of $TargetSystem$. In Equation (3), $Component_{Adv}$ is the set of components under the control of the adversary that does not present on $TargetSystem$. For example, $Component_{Adv}$ could include an attacking terminal used by the adversary that does not exist in $TargetSystem$. Similarly, $PhysicalArea_{Adv}$ is the set of physical areas owned by the adversary that do not exist in $TargetSystem$, and $Network_{Adv}$ is the set of networks owned by the adversary that do not exist in $TargetSystem$.

The configuration of $TargetSystem$ is subject to certain constraints. For example, it cannot exist if there are no components. This is a constrain condition relating to the pattern of constituent elements in $TargetSystem$. The relationships between the elements of $TargetSystem$ in this setup are shown in FIGURE 3.

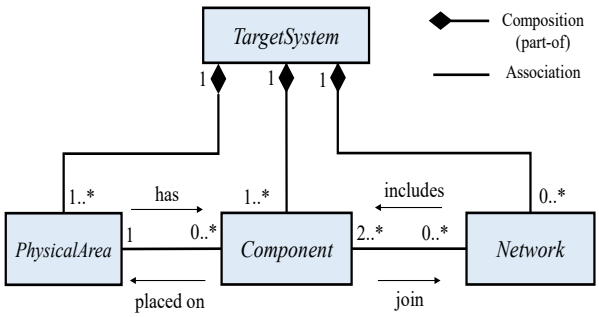


**FIGURE 3. The structure of $TargetSystem$ (class diagram)**

FIGURE 3 depicts the class diagram representing the structural framework of $TargetSystem$. The structure of $TargetSystem$ comprises at least one physically existing $Component$, at least one $PhysicalArea$, and a $Network$ with zero or more segments. A $PhysicalArea$ has zero or more $Component$s, and a $Component$ exists in one member of $PhysicalArea$. A $Network$ contains at least two $Component$s, and the minimum unit is a peer-connected pair. If there is only one $Component$ present in the $TargetSystem$, this means that there is no $Network$.

Next, the $ThreatActor$ model consists of the $Adversary$ in the Diamond model (a set of adversary entities, such as a group of attackers), a $Capability$ (a set of attack patterns such as one or more attack methods or tools used by an arbitrary $ThreatActor$), and the $Goal$ of the attack (the adversary's ultimate objective) (Cond. iii). The $ThreatActor$ is represented as a collection of elements in each set which can be formulated as shown in Equation (4).

$$ThreatActor = (adv, Cp, G) \cdots\cdots (4)$$

Here, $adv$ is an adversary entity that is involved in $ThreatActor$ and satisfies $adv \in Adversary$, $Cp$ is a collection of techniques and tools used by the adversary that satisfies $Cp \subset Capability$,

and $G$ is the set of the adversary's goals, of which there may be more than one, so if $Goal$ is the set of ultimate goals, then $G \subset Goal$.

To derive a formula for $Goal$, we represent the elements of $Goal$ as the Cartesian product (group) of two groups comprising the components present in $TargetSystem$ ( $v_{last}$ ) and the attack actions ($act$) (condition iv). This results in the formula shown in Equation (5).

$$Goal = \{(v_{last}, act) \mid v_{last} \in Component,\ act \in Capability\} \quad \cdots\cdots (5)$$

Here, $act$ is an element of $Capability$, and is based on the theorem that the goal cannot be achieved if the adversary does not have the ability to perform the attack action [21].

Next, we'll discuss the conditions for a successful attack action. If $Capability_{Adv}$ is a finite set of the attack patterns available to $Adversary$, then the necessary condition for the success of an attack action can be expressed as shown in Equation (6).

$$act \in Capability_{Adv} \quad \cdots\cdots (6)$$

### *(b) THE BEHAVIOR MODEL*

Based on the structural model described above, we defined a behavioral model of attacks. The conditions of the model are as follows.

(Cond. v) A series of attacks by a $ThreatActor$ ($AttackScenario$) includes intermediate states of the adversary (between the successful completion of one attack action and an attempt on the next attack action; referred to hereinafter as intermediate attack states).

(Cond. vi) At any given intermediate attack state, a $ThreatActor$ will attempt an attack pattern specified by $Capability$.

(Cond. vii) The attack pattern at any given intermediate attack state depends on the attack phase and victim and includes the adversary's attack position and the conditions at the time of the attack (referred to hereinafter as attack conditions).

(Cond. viii) The attack conditions are a set of conditions that the adversary has to make an attack action worth doing at an intermediate attack state.

(Cond. ix) The $Victim$ may change during the course of an attack.

According to (Cond. v), $AttackScenario$ is represented by an ordered set whose elements are the intermediate attack states of $ThreatActor$. If we denote the intermediate attack state of attack step $k$ as $AttackState_k$, then $AttackScenario$ can be formulated as shown in Equation (7).

$$AttackScenario = AttackState_1 \prec AttackState_2 \prec AttackState_3 \prec \cdots \prec AttackState_k \prec \cdots \prec AttackState_{n-1} \prec AttackState_n \quad \cdots\cdots (7)$$

Here, the relational operator $A \prec B$ signifies that element $A$ precedes element $B$ in the ordered set, subscripts $k$ and $n$ are integers that satisfy $\{k, n \in \mathbb{N} \mid 0 \leq k \leq n\}$, and $AttackState_n$ represents the $n$-th intermediate attack state.

Next, we derive a formula for $AttackState_k$. Based on (Cond. vi) through (Cond. xi), $AttackState_k$ can be represented by the set of variables shown in Equation (8).

$$AttackState_k = (p_k, v_k, Cp_k, lo_k, Cond_k) \quad \cdots\cdots (8)$$

In equation (8), $AttackState_k$ is the $k$-th attack phase (the elements $p_k \in AttackPhase$ of the set $AttackPhase$ consisting of the basic elements of the attack phase), which comprises the victim $v_k \in Component$ of this attack phase, the finite set $Cp_k \subset Capability_{p_k, v_k}$ of the attack capabilities for attack phase $p_k$ and victim $v_k$, the $ThreatActor$ attack location $lo_k \in TargetSystem$, and the finite set of attack conditions $Cond_k \subset AttackCondition$.

$AttackCondition$ is the set of attack conditions shown in (Cond. viii). Specifically, it consists of $Cred_{ALL}$, which is the finite set of all credential information such as passwords before the use of a component that exists in the $TargetSystem$, and $Admin_{ALL}$, which is the finite set of administrative privileges (referred to as "access privileges" hereinafter) of all the components in $TargetSystem$. $AttackCondition$ is formulated as shown in Equation (9).

$$AttackCondition = \{Cred_{ALL}, Admin_{ALL}\} \quad \cdots\cdots (9)$$

$ThreatActor$ uses the above attack conditions in the attack process to obtain an attack advantage. For example, if there is a

$Component$ with the same password, the authentication can easily be broken by reusing this password. Access privileges are another condition that can be used to the attacker's advantage in a similar fashion. For example, suppose there is a component on $TargetSystem$ ($\mathrm{comp_A} \in TargetSystem$) that has the same access privileges as another component on $TargetSystem$ ($\mathrm{comp_B} \in TargetSystem$). Under this condition, if $ThreatActor$ acquires access privileges for $\mathrm{comp_A}$, then it is also possible to gain administrator access to $\mathrm{comp_B}$ without any additional attack actions.

$Cond_k$ is a subset of $AttackCondition$ that is obtained by the adversary on reaching $AttackState_k$. For example, Equation (10) shows $Cond_k$ for the case where the adversary has the credential information ($cred_1$, $cred_2$) and access privileges ($admin_1$, $admin_2$) of $\mathrm{comp_1}$ and $\mathrm{comp_2}$.

$$Cond_k = \{cred_1,\ cred_2,\ admin_1, admin_2\} \cdots\cdots\cdots\cdots (10)$$

A necessary condition for advancing from $AttackState_k$ to $AttackState_{k+1}$ is that there exists an attack pattern that is available to the adversary. In other words, Equation (11) must be satisfied.

$$Cp_k \neq \emptyset \cdots\cdots\cdots\cdots (11)$$

In other words, if Equation (11) does not hold during attack phase $p_k$, then there is no attack pattern, and the attack scenario ends at $p_k$.

Finally, we consider the reachability of the ultimate goal of the $AttackScenario$. If $AttackState_{END}$ represents the intermediate attack state at the point where the $Goal$ of the $AttackScenario$ shown in Equation (5) has been reached, then the reachability can be evaluated in terms of whether or not $AttackScenario$ can reach $AttackState_{END}$.

If reachability is expressed by a positive integer $k$ in our proposed model, then at the point where an attacker moves from $AttackState_k$ to $AttackState_{k+1}$, if any element of the set defined by $Goal$ shown in Equation (5) (i.e., any ultimate target component $v_{last}$ paired with an attack action $act$) is achieved, then $AttackState_{k+1} = AttackState_{END}$. In other words, if $AttackState_n = AttackState_{END}$ is satisfied for some positive integer $n$, then this is a necessary and sufficient condition for reachability.

### (c) ATTACK PHASE AND CAPABILITY MODELS

For the set $AttackPhase$ to which $p_k$ of Equation (8) belongs and the set $Capability$ that has $Cp_k$ as a subset, we built a model based on ATT&CK® (referred to as ATT&CK hereinafter). This is a standard model of an adversary's TTP (Tactics, Techniques and Procedures) developed by MITRE, and consists of a matrix representation of the adversary's tactics and the techniques used to implement them. Varieties of ATT&CK include ATT&CK for Enterprise [22], which is aimed at enterprise systems, and ATT&CK for Industrial Control Systems [23], which is specialized for use with ICS. Here, we define the adversary behavior model based on ATT&CK. If $ATT\&CK_{Tac}$ is a finite set of tactics defined by ATT&CK (referred to as ATT&CK Tactics hereinafter) and $ATT\&CK_{Tech}$ is a finite set of techniques defined by ATT&CK (referred to as ATT&CK techniques hereinafter), then $AttackPhase$ and $Capability$ can be expressed as shown in Equations (12)–(14).

$$AttackPhase = \{\, p \mid p \in ATT\&CK_{Tac} \,\} \cdots\cdots\cdots\cdots (12)$$

$$Capability = \{\, c \mid c \in ATT\&CK_{Tech} \,\} \cdots\cdots\cdots\cdots (13)$$

$$Capability_{p,v} = \{\, c \mid c \in ATT\&CK_{Tech,p,v} \,\} \cdots\cdots\cdots\cdots (14)$$

In Equation (14) $ATT\&CK_{Tech,p,v}$ is a finite set of ATT&CK techniques that can be performed on victim $v$ in attack phase $p$. Or to put it another way, $ATT\&CK_{Tech,p,v}$ is a finite set whose elements are techniques that are valid against victim $v$ from among the ATT&CK techniques in the single column of the ATT&CK matrix for Tactics $p$.

### (d) GENERATING ATTACK PATHS

Next, we discuss the possible attack paths of $ThreatActor$ (the change history of attack position $lo_k$). In the initial state $AttackState_0 = (p_0, v_0, Cp_0, lo_0, Cond_0)$, the $ThreatActor$ exists at the starting position $lo_0$, and attacks are attempted in order to reach the final victim ($v_{last}$) from $lo_0$ as indicated by the $Goal$ of the $ThreatActor$. Here, $lo_0$ is an element of the set of physical areas in which a component $\{c \mid c \in Component\}$ is installed $\{PhysicalArea_{Dep} \mid PhysicalArea_{Dep} \subset PhysicalArea\}$, or the set of external networks such as the Internet that can be the source of intrusion by the $ThreatActor$ $\{Network_{Ext} \mid Network_{Ext} \subset Network\}$, or the set of components that can be the source of intrusion via the supply chain

$\{Component_{Sup} | Component_{Sup} \subset Component\}$ . $lo_0$ is therefore expressed by the formula shown in Equation (15).

$$lo_0 = \{e \mid e \in \{Component_{Sup}, PhysicalArea_{Dep}, Network_{Ext}\}\} \cdots\cdots\cdots\cdots (15)$$

The attack path is determined from a graph consisting of each constituent element of $TargetSystem$. This graph is represented as an undirected graph whose vertices are the elements of the set $\{PhysicalArea, Component\}$ and whose edges are the elements of $Network$. Then, the attack path graph of the $TargetSystem$ ( $PathGraph, Node\ V, Edge\ E$ ) are expressed as shown in Equations (16)–(18).

$$PathGraph = \{V, E\} \cdots\cdots\cdots\cdots (16)$$

$$V = \{Component, PhysicalArea, Dummy\} \cdots\cdots\cdots\cdots (17)$$

$$E = \{Network, local\} = \{(Dummy, comp1), (area1, comp1), (comp1, comp2), (area2, comp2), \ldots\} \cdots\cdots\cdots\cdots (18)$$

An example of a graph in $TargetSystem$ where each finite set is represented by Equations (19)–(21) is shown in FIGURE 4.

$$Component = \{comp1, comp2, comp3, comp4, comp5\} \cdots\cdots\cdots\cdots (19)$$

$$PhysicalArea = \{area1, area2, area3\} \cdots\cdots\cdots\cdots (20)$$

$$Network = \{nw1, nw2, nw3, internet \mid nw1, nw2, nw3 \notin Network_{Ext}, internet \in Network_{Ext}\} \cdots\cdots\cdots\cdots (21)$$

In this example, unless $Component$ is a stand-alone system, it is connected to one of the four networks ( $nw1, nw2, nw3, internet$ ). When an element of $Network_{Ext}$ exists (such as $internet$), the component to which it is connected is regarded as unknown and is defined as a dummy node ( $dummy\ node$ ). As physical placement information, each component is placed in one of the physical locations ( $area1, area2, area3$ ). These physical locations and the component perform local access and are connected by virtual edges ($local$), that means locally accessible.

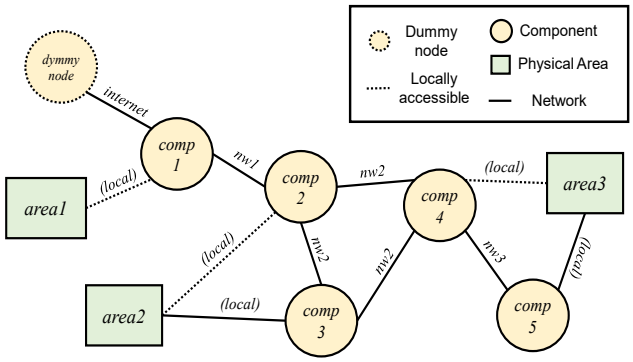


**FIGURE 4.** **An example of *PathGraph***

Taking the $PathGraph$ of FIGURE 4 as an example, the attack path generation procedure for a $ThreatActor$ with the ultimate goal of $(comp5, act)$ who is at the starting position ($l_0 = area1$) is as follows.

(1) Launch an attack from $l_0 = area1$ to victim $v_0 = comp1$ using local access.

(2) After a successful attack and takeover of $comp1$, attempt an attack on the next victim ($comp2$). During this attack, if there is no attack pattern available at step $k$, i.e., if element $Cp_k$ of $AttackPhase_k$ satisfies Equation (11), then the attacks can continue no further and the process ends.

(3) If it is possible to reach the attack target component ($v_{last} = comp5$) and execute the attack pattern $act$ specified in the ultimate goal of the $ThreatActor$ for $comp5$, then the attack is considered successful and the attack path generation process is completed.

Next, we discuss the validity of attacks and how to determine the applicability of an initial position $l_0$. The feasibility of an attack depends on the characteristics of the target. For example, in ATT&CK techniques, the feasibility of an attack technique depends on the software platform of the target component, such as its operating system (hereinafter referred to as the platform) [21]. If an attack is presented as a technique for Linux™ (hereinafter referred to as "Linux"), then the platform of the target component must be Linux.

Regarding the attack conditions, a component has the credential information and operating privileges of the components on the $TargetSystem$, including itself. The adversary obtains these credentials during the attack process and uses them for subsequent

attacks by expanding the attack conditions.

As for the feasibility of an attack path, it is necessary to clarify the physical area in which a component is in determining whether or not the component is accessible from the physical area of the $TargetSystem$.

In terms of components, there is a possibility that one component is a network control device such as a network switch, which requires the association of attack techniques specific to network control devices. Furthermore, regarding the initial location $l_0$ it is possible to determine whether $TargetSystem$ element $e$ satisfies Equation (14). On the other hand, it is necessary to define information for judging whether element $e$ should be included.

Based on the above, the attribute information is assigned to the $Component$ and $Network$ elements of $TargetSystem$. Each element of attribute information is denoted by $Component_{Attr}$ and $Network_{Attr}$, and is expressed as a pair as shown in Equations (22) and (23).

$$Component_{Attr} := (name, pf, Admin, Cred, area) \quad \cdots\cdots (22)$$

$$Network_{Attr} := (name, Ctldev, isExt) \quad \cdots\cdots (23)$$

In Equation (22), $name$ is the identifier of a component that does not overlap with the $TargetSystem$, $pf$ denotes the component's software platform, including operating systems such as Windows® (hereinafter referred to as Windows) and Linux, $Admin$ is the set of credential information held by the component, $Cred$ is the set of the component's access privileges, which must have the access privileges of the component itself, and $area$ is the physical area where the component is located, which satisfies $area \in PhysicalArea$.

In equation (23), $name$ is the identified of a non-overlapping network on the $TargetSystem$, $Ctldev$ is a control device such as a switch on the target network, which satisfies $Ctldev \in Component$, and $isExt$ is a Boolean value that is True if $nw \in Network_{Ext}$ or False if $nw \notin Network_{Ext}$.

Based on this model, the conditions for the generation of an attack path are determined based on the attribute information shown above.

### *(e) GENERATING ATTACK ACTIONS*

If $ThreatActor$ succeeds in the attack action at $AttackState_k$, it moves on from $AttackState_k$ to $AttackState_{k+1}$. This $AttackState_k$ transition history corresponds to $AttackScenario$ in Equation (7). The details of this model are described below.

- In $AttackState_0 = (p_0, v_0, cp_0, lo_0, cond_0)$, $p_0$ is the initial penetration (the Initial Access phase in ATT&CK), which is the state in which $ThreatActor$ makes its first attempt to enter $TargetSystem$.
- In $AttackState_k = (p_k, v_k, cp_k, lo_k, cond_k)$, when $v_k$ is not the ultimate goal component $v_{last}$ of $ThreatActor$, the $ThreatActor$ makes every effort to expand laterally (the Lateral movement phase in ATT&CK).
- After the initial penetration or lateral deployment, the attack code is executed to achieve the attack phase on $v_k$.
- After executing the attack code, if $v_k \neq v_{last}$ then expand laterally to the next component. Or if $v_k = v_{last}$, attempt the action specified in $Goal$.
- When advancing $k \rightarrow k+1$, if the lateral expansion is successful, then the adversary's position becomes the component before the lateral expansion ($lo_{k+1} = v_k$), and $v_{k+1}$ becomes the next victim component.
- $cond_k$ is a set of parameters obtained by the adversary during the attack and is updated when any changes occur during a successful attack. For example, when credential information $cred_c$ is obtained from a component $c \in Component$ at $v_k$, then $cond_{k+1} = \{cred_c\}$. In this case, when the attack $k+1$ is attempted, the adversary can reuse $cred_c$. In essence, if a distinct component utilizes the identical credential information $cred_c$ as that is used by component c, it significantly simplifies the task for an adversary to breach the authentication of the component. Also, if the access privileges of component $c$ ($admin_c$) are acquired (completing the Execution phase and Privilege Escalation phase in ATT&CK), then $cond_{k+1} = \{admin_c\}$. In this case, the adversary can use all the functions of component $c$ when attempting attack $k+1$.

The attack phase $p$ of $AttackState_k$ is updated according to the constraints between the elements of $ATT\&CK_{Tac}$. An example of the constraints (transition rules) is shown in FIGURE 5. Here, the adversary attempts an initial access to the $TargetSystem$ via a remote or local path (initial access phase), and then executes the attack code (execution phase). If the component here is the component specified as the $Goal$, then an attack that affects the ICS is attempted (Impact phase), otherwise a search is performed inside the component for which the attack is successful (Discovery phase) and credential information is acquired (Credential Access phase). After that, further attempts are made to penetrate the network (Discovery phase), and if the component specified as the $Goal$ is

found, an attack on this component is attempted (Impact phase). During this attack attempt, if the adversary already has access privileges for this component, it attempts to attack the component using these privileges (Impact phase). Or, if the adversary does not have access privileges, an attack is attempted remotely. If this remote attack is not possible, or if this is not a component specified as a $Goal$, then an attempt is made to recapture the access privileges of this component. The above processing is attempted until a component specified as a $Goal$ is reached.

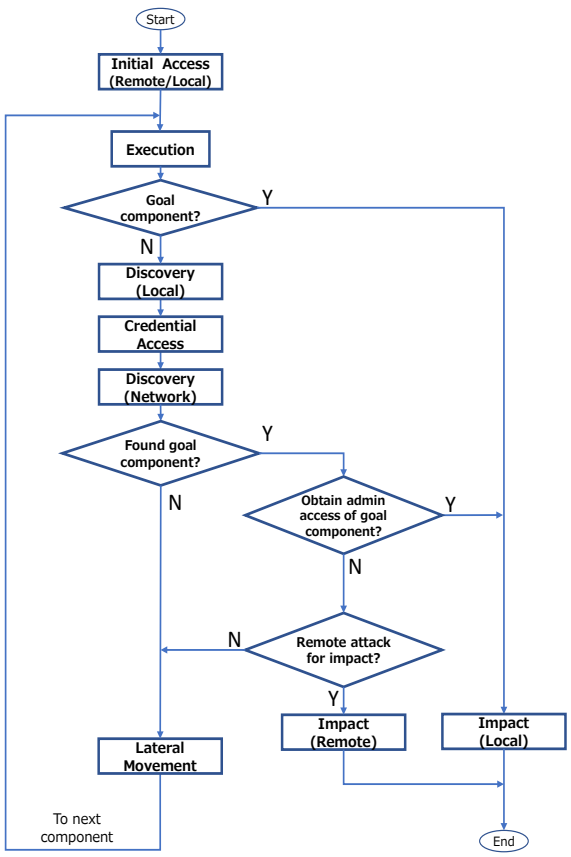


**FIGURE 5. An example of transition rules (flowchart)**

Here, to fit with the attack model of the $ThreatActor$, the ATT&CK model needs to be extended. The tactic defined in ATT&CK is that the Initial Access phase, Discovery phase, and Impact phase are the same for local and network attack attempts. The tactics defined in ATT&CK for the Initial Access phase, the Discovery phase, and the Impact phase do not differ between local and network attack attempts. Therefore, to fit into this attack action model, the tactics of the above phases are distinguished between local attacks and network attacks. This transition rule is different according to $ThreatActor$, and is rearranged according to $ThreatActor$. For example, to attempt the Privilege Escalation phase, $ThreatActor$ is a transition flowchart that includes the Privilege Escalation phase. Following the procedure described above, we can reproduce the attacker's behavior on the $TargetSystem$.

## III. EVALUATION AND CONSIDERATION

### A. MODELS FOR EVALUATION

For the evaluation target model shown in FIGURE 6. FIGURE 6 presents the abstract structure of the target system as delineated for the purpose of this study. The target system consists of three logical networks: Internet, IT network (IT-NW), and OT network (OT-NW). The Internet is a public network in which unspecified systems and devices participate. The IT-NW is an internal network that provides communication infrastructure among enterprise computers. It contains a host computer that works as a human-machine interface (HMI-PC), a gateway that controls communication between the Internet and IT-NW (Gateway), and a network switch device that manages IT-NW (IT-SW). The OT-NW is an internal network that provides communication information among control system components. It contains an engineering workstation, a host computer that is capable of engineering for PLC (EWS), a programmable logic controller that controls equipment in the control system field (PLC), a server machine that acts as a communication gateway between IT-NW and OT-NW (Server), and a network switch device which manages OT-NW (OT-SW). The target system includes the Plant, a series of equipment controlled by PLC. In this study, the equipment of the Plant is not regarded as a component of the target system. The target system contains two physical areas: A control area where components participating in the OT-NW is placed (Ctrl-Area) and a machine area where components participating in the IT-NW is placed (M-Area).

We generated an attack scenario according to the attack model shown in the previous section. TABLE III lists the components of the model we evaluated, which has attribute information based on Equation (22). Each column of TABLE III shows the attribute information for each component, which consists of the component's instance name (Name), the software platform (Platform) used by the component, its access privileges over other components (Admin), its authentication credentials (cred.), the component's physical location (Area) and networks which the component is connected (NW).

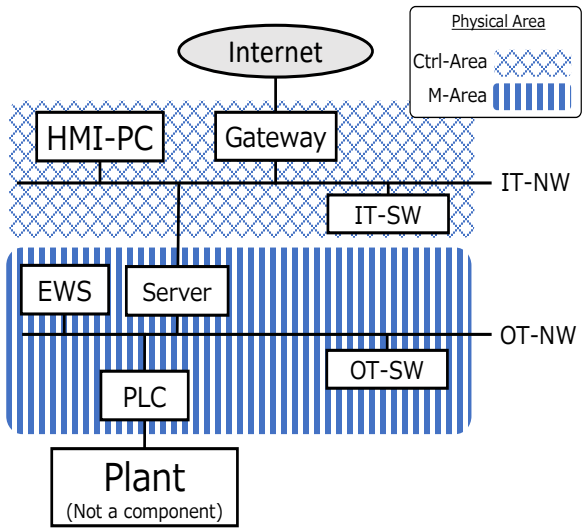


**FIGURE 6.** **An image of the target system for the evaluation**

TABLE III

A LIST OF ***Component*** ELEMENTS IN THE TARGET SYSTEM

| # | Name | Platform | Admin | Cred. | Area | NW |
|---|---|---|---|---|---|---|
| 1 | HMI-PC | Windows | $Admin_{HMI\text{-}PC}$ | $cred_1$ | Ctrl-Area | IT-NW |
| 2 | Server | Linux | $Admin_{Server}$ | $cred_1$ | M-Area | IT-NW<br>OT-NW |
| 3 | EWS | Windows | $Admin_{EWS}$<br>$Admin_{PLC}$ | $cred_1$ | M-Area | OT-NW |
| 4 | PLC | Other | $Admin_{PLC}$ | - | M-Area | OT-NW |
| 5 | Gateway | Linux | $Admin_{Gateway}$ | - | Ctrl-Area | IT-NW<br>Internet |
| 6 | IT-SW | Other | $Admin_{IT\text{-}SW}$ | - | Ctrl-Area | IT-NW |
| 7 | OT-SW | Other | $Admin_{OT\text{-}SW}$ | - | M-Area | OT-NW |

TABLE IV lists the networks used in the evaluation model, with attribute information according to Equation (23). Each column of this table shows the attribute information of the corresponding network, which consists of the network's name, the control device that controls the network, and whether the network is externally connected.

TABLE IV

A LIST OF ***Network*** ELEMENTS IN THE TARGET SYSTEM

| # | Name | ControlDevice | External? |
|---|---|---|---|
| 1 | Internet | - | True |
| 2 | IT-NW | IT-SW | False |
| 3 | OT-NW | OT-SW | False |

TABLE V shows the threat actor's adversary and goal. In this example, the adversary is called APTxx, and its goal is loss of control. The goal of this threat actor follows Equation (4) and TABLE V, and is expressed in the form (PLC, "T1485 - Data Destruction"). If any of the attacks listed in the Techniques column for PLC is successful, then the attack against the PLC is deemed successful.

TABLE V

***Adversary*** AND ***Goal*** OF ***ThreatActor*** IN THE EVALUATION

| Adversary | Goal | Target | Techniques |
|---|---|---|---|
| APTxx | Loss of Control | PLC | TA0040 - Impact (Local/Remote)<br>- T1485 - Data Destruction<br>- T1565 - Data Manipulation<br>- T1499 - Endpoint Denial of Service |

TABLE VI lists the capabilities that this threat actor is assumed to possess. This threat actor is a hypothetical adversary group based on multiple cases of attacks on ICSs, such as Dragonfly [24][25], Lazarus [26] and Darkside [27], and its capabilities consist of ATT&CK techniques listed under "Technique" and ATT&CK tactics corresponding to each technique shown under "Tactics".

TABLE VI

A LIST OF ***Capability*** ELEMENTS IN THE EVALUATION

| # | Tactics | Techniques |
|---|---|---|
| 1 | TA0001 - Initial Access (Remote) | T1189 - Drive-by Compromise<br>T1133 - External Remote Services<br>T1566 - Phishing |
| 2 | TA0001 - Initial Access (Local) | T1078 - Valid Accounts |
| 3 | TA0002 - Execution | T1059 - Command and Scripting Interpreter<br>T1203 - Exploitation for Client Execution<br>T1204 - User Execution |
| 4 | TA0007 - Discovery (Local) | T1087 - Account Discovery<br>T1201 - Password Policy Discovery<br>T1518 - Software Discovery |
| 5 | TA0006 - Credential Access | T1110 - Brute Force<br>T1555 - Credentials from Password Stores |
| 6 | TA0007 - Discovery (Network) | T1040 - Network Sniffing<br>T1049 - System Network Connections Discovery |
| 7 | TA0008 - Lateral Movement | T1210 - Exploitation of Remote Services<br>T1021 - Remote Services<br>T1550 - Use Alternate Authentication Material |
| 8 | TA0040 - Impact (Local) | T1485 - Data Destruction<br>T1565 - Data Manipulation<br>T1499 - Endpoint Denial of Service |
| 9 | TA0040 - Impact (Remote) | T1565 - Data Manipulation<br>T1499 - Endpoint Denial of Service |

**AttackScenario I**

AttackState[0]: {point: Internet, target: HMI-PC, phase: INITIAL_ACCESS(Remote), conds: {creds: [], admins: []}}

AttackState[1]: {point: HMI-PC, target: HMI-PC, phase: EXECUTION, conds: {creds:[], admins: []}}

AttackState[2]: {point: HMI-PC, target: HMI-PC, phase: DISCOVERY(Local), conds: {creds: [], admins: [HMI-PC]}}

AttackState[3]: {point: HMI-PC, target: HMI-PC, phase: CREDENTIAL_ACCESS, conds: {creds: [], admins: [HMI-PC]}}

AttackState[4]: {point: HMI-PC, target: ???, phase: DISCOVERY(Network), conds: {creds: [cred1], admins: [HMI-PC]}}

AttackState[5]: {point: HMI-PC, target: Server, phase: LATERAL_MOVEMENT, conds: {creds: [cred1], admins: [HMI-PC]}}

AttackState[6]: {point: Server, target: Server, phase: EXECUTION, conds: {creds: [cred1], admins: [HMI-PC]}}

AttackState[7]: {point: Server, target: Server, phase: DISCOVERY(Local), conds: {creds: [cred1], admins: [HMI-PC, Server]}}

AttackState[8]: {point: Server, target: Server, phase: CREDENTIAL_ACCESS, conds: {creds: [], admins: [HMI-PC, Server]}}

AttackState[9]: {point: Server, target: ???, phase: DISCOVERY(Network), conds: {creds: [cred1], admins: [HMI-PC, Server]}}

AttackState[10]: {point: PLC, target: PLC, phase: IMPACT(Remote), conds: {creds: [cred1], admins: [HMI-PC, EWS, Server, PLC]}}

**AttackScenario II**

AttackState[0]: {point: M-Area, target: HMI-PC, phase: INITIAL_ACCESS(Local), conds: {creds: [], admins: []}}

AttackState[1]: {point: OT-SW, target: HMI-PC, phase: EXECUTION, conds: {creds: [], admins: []}}

AttackState[2]: {point: OT-SW, target: HMI-PC, phase: DISCOVERY(Local), conds: {creds: [], admins: [OT-SW]}}

AttackState[3]: {point: OT-SW, target: HMI-PC, phase: CREDENTIAL_ACCESS, conds: {[creds: [], admins: [OT-SW]]}

AttackState[4]: {point: OT-SW, target: ???, phase: DISCOVERY(Network), conds: {creds: [], admins: [OT-SW]}}

AttackState[5]: {point: EWS, target: EWS, phase: LATERAL_MOVEMENT, conds: {creds: [], admins: [OT-SW]}}

AttackState[6]: {point: EWS, target: EWS, phase: EXECUTION, conds: {creds: [], admins: [OT-SW, EWS]}}

AttackState[7]: {point: EWS, target: EWS, phase: DISCOVERY(Local), conds: {creds: [], admins: [OT-SW, EWS]}}

AttackState[8]: {point: EWS, target: EWS, phase: CREDENTIAL_ACCESS, conds: {creds: [], admins: [OT-SW, EWS]}}

AttackState[9]: {point: EWS, target: ???, phase: DISCOVERY(Network), conds: {creds: [cred1], admins: [OT-SW, EWS]}}

AttackState[10]: {point: PLC, target: PLC, phase: IMPACT(Local), conds: {creds: [cred1], admins: [OT-SW, EWS]}}

**FIGURE 7.** **Examples of expressions representing attack scenarios against the target system in this study**

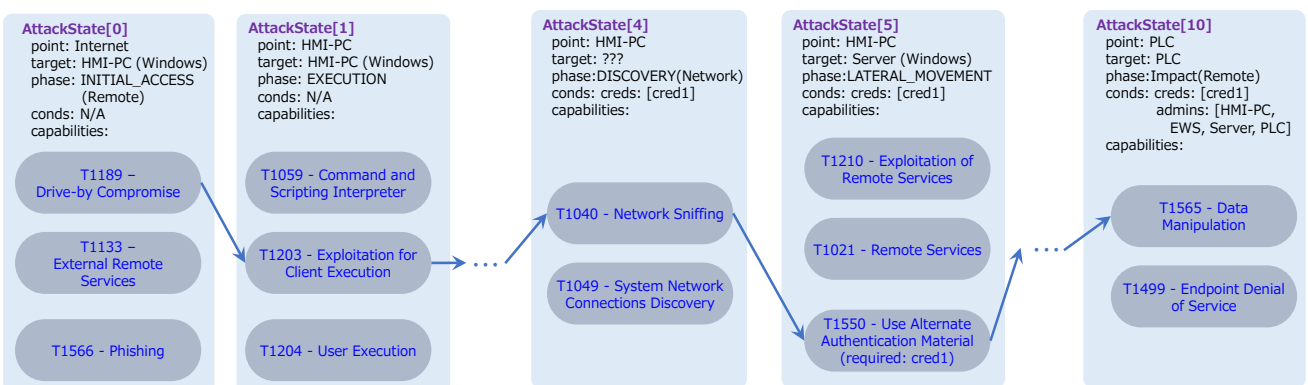


**FIGURE 8.** **An illustration of attack technique selection at each phase of an attack**

### B. SIMULATION PROCESS AND RESULTS

Based on the evaluation system described in the previous section, the scenario generation process according to this model is shown below.

1. Generate a $TargetSystem$ model.
2. Generate the graph shown in FIGURE 4 from the $TargetSystem$.
3. Generate a $ThreatActor$ model from the adversary information ($Goal$, $Capability$).
4. Determine $AttackState_0$ based on $TargetSystem$ and $ThreatActor$. The initial position $l_0$ is based on the adversary's $Capability$, and specifies an element on the $TargetSystem$ that may become the starting point of an attack.
5. Following the transition flowchart shown in FIGURE 5, attempt attack actions from $AttackState_0$ until the $Goal$ is achieved. When an attack can be made (i.e., when Equation (11) is satisfied at $AttackState_k$), move on to the next attack phase. Conversely, if Equation (11) is not satisfied, further scenario development is deemed impossible, and the processing ends.

An example of an attack scenario generated based on the system under evaluation is shown in FIGURE 7. AttackScenario I is an attack scenario that originates from an external network (the Internet). The adversary switches its victim from the Internet (network) to HMI-PC (component), IT-NW (network), Server (component), OT-NW (network), and finally PLC (target) while implementing an attack scenario aimed at the $Goal$. Also, AttackScenario II is a scenario that starts from a physical area (M-Area) and switches its victim from M-Area (physical area) to OT-NW Switch (component), OT-NW (network), EWS (component), and finally PLC (target). The attack on the PLC in AttackScenario II is performed by using access privileges obtained from EWS ($Admin_{PLC}$ in TABLE III), and attempts to perform unauthorized operations in the PLC. Also, in the "DISCOVERY (network)" phase, the next victim that is shown as "target" in FIGURE 7**.** Since the victim of this attack phase is uncertain, it is shown as "???". The target is determined on completing this attack phase and can be any of the attackable components participating in this network (in this example, EWS is selected).

In each phase of the attack, the attacker generates a series of attack techniques. The example of the phase transition of $AttackState$ is shown in FIGURE 8**.** This figure is based on the attack pattern specified as $Capability$ (Techniques in TABLE VI). If any of the attack patterns shown in TABLE VI succeeds, the attacker considers this attack phase to be complete and moves on to the next attack phase.

For some attacks, such as "Use Alternate Authentication Material (T1550)", which reuses credential information obtained earlier, it is necessary to add further conditions for the attack to be successful (such as credential information). In this case, this attack phase is only considered to be complete only if these attack conditions are satisfied in the corresponding step of the attack step.

In this way, we have shown that the proposed model can structurally represent attack scenarios.

### C. DISCUSSIONS ON UTILIZATIONS OF THE MODELS

With the proposed representation model, we have shown that threat events can be structurally represented from the viewpoint of the adversary's behavior on ICS. Here, we discuss the reusability of threat events and their usefulness in terms of threat prioritization, which is the purpose of this study.

First, we discuss the reusability of threat events. By predefining $ThreatActor$ patterns, it is possible to reuse these patterns for the $TargetSystem$ represented by this proposed model. In other words, the behavior of a $ThreatActor$ can be reproduced even in different system configurations. The same applies to the conditions for transitioning between attack phases. By predefining the success conditions of each attack ( $Capability$ ) corresponding to $AttackState_k$ shown in Equation (8), they can be reused in the form of a template.

The $ThreatActor$ patterns must be defined individually. Since the parameters set in the definition of a $ThreatActor$ have a high degree of freedom, they are expected to vary widely when they are defined individually due to differences in personal judgment. This issue can be addressed by defining patterns based on standard models such as ATT&CK. For example, ATT&CK provides a TTP model of adversary groups called ATT&CK Groups (e.g., Dragonfly, an adversary group that targets power control systems) [24][25]. By using this model in the $ThreatActor$ definition, it is possible to reproduce the behavior of standard attacker groups and eliminate differences of personal judgment from the $ThreatActor$ definition.

Next, we consider the usefulness of the proposed model in terms of threat prioritization. By defining separate models for $TargetSystem$ and $ThreatActor$, we can evaluate whether there are any attack scenarios capable of achieving the goal of the

$ThreatActor$. If an achievable scenario exists, then it is possible for the define $ThreatActor$ to launch an attack on the $TargetSystem$, which can be regarded as a threat event with a high degree of likelihood. This makes it possible to evaluate threat events and attack scenarios in terms of whether it can reach the adversary's ultimate goal and has the advantage of making it easy to obtain agreement among the parties concerned on how threat events should be prioritized.

$ThreatActor$ patterns also provide a basis for prioritizing threat events. For example, referring to the definition of likelihood shown in TABLE I, a high likelihood index corresponds to threats and vulnerabilities that are highly likely to occur within the next year.

This is applied to the threat events expressed in the proposed model. For example, a threat event with a high likelihood can be represented as a judgment of the possibility of achieving the attack goal of $ThreatActor_H$, where $ThreatActor_H$ is a threat actor who posed an active threat at the time of threat modeling. In addition, the information about threat actors who pose an active threat can be obtained by gathering threat intelligence [28]. In other words, the proposed model also makes it possible to prioritize threat events based on threat intelligence.

Based on the above discussion, we conclude the proposed methodology enables us to represent the process by which a predefined adversary progresses from possible initial points of system intrusion to final goals. Consequently, it facilitates the determination of attack scenarios and prioritizing qualitatively eliminating personal judgement.

### D. RISK QUANTIFICATION

In the subsequent, we concentrate on the methodology for mitigating personal judgment in prioritizing threat events quantitatively. Ideally, threat events would be prioritized by quantitative metrics. However, it is generally difficult to estimate the risk of each event quantitatively [29]. Since this method reveals the internal structure of threat events, it would be possible to express the risk of threat events by quantitative measures. Consequently, in the prioritization of threat events, there is potential for further reduction of elements of personal judgement. In this section, the risk of each threat event by quantitative metrics of risk based on this proposed method is discussed.

According to the definition provided by ISO 31000:2018, risk is referred to as the "effect of uncertainty on objectives," and the degree of its influence is termed as risk value [30]. Within the context of security, this refers to the extent to which a given threat event impacts the entity susceptible to the threat's harm. The higher the risk value of a threat event, the greater its associated risk. The risk value is represented by equation (24) and is expressed as the product of *Impact* and *Likelihood*.

$$\boldsymbol{Risk = Impact \times Likelihood} \tag{24}$$

The Impact refers to the negative consequences an ICS owner faces as a result of an attack. Examples include business disruptions due to system outages, effects on safety and the environment, and business losses associated with the extraction of trade secrets. The Likelihood, on the other hand, pertains to the certainty with which an Impact event materializes. Within the realm of security, this is indicative of the probability of an attack's success.

The magnitude of risk is proportionate to both the severity of the impact when a threat event materializes and the likelihood of its occurrence. Consequently, threat events with higher associated risks tend to be prioritized. By employing the proposed method for calculating risk value, the structure of the threat event becomes more tangible, making it feasible to expect a quantitative representation of the risk value. In this section, we attempt a quantitative representation of risk for scenarios depicted using the proposed method.

Firstly, we consider the aspect of the Impact. In attack scenarios depicted by the proposed model, the $ThreatActor$ executes a series of malicious activities at each $AttackState$, aiming to infiltrate and ultimately target a specific entity. Upon reaching the final target, the attack is executed. While the objectives of attackers targeting conventional information systems can be diverse, in the context of ICS, the primary objectives typically revolve around halting or sabotaging the system or extracting trade secrets related to production and control processes.

Considering a continuous attack scenario, while there might be damage incurred during the attack, within such a scenario, the Impact indicated by the 'Goal' represents the most significant impact. Therefore, if the impact occurring in a particular attack scenario is denoted as $\boldsymbol{Impact_{Sc}}$ and the impact at the time of achieving the final objective is defined as $\boldsymbol{Impact_{Last}}$, this relationship can be represented by equation (25).

$$\boldsymbol{Impact_{Sc} \approx Impact_{Last}} \tag{25}$$

The method proposed herein determines the attack pattern based on the $ThreatActor$'s $Goal$. Consequently, the Impact of an $AttackScenario$ is defined by its $Goal$. When the $Goal$ is constant, it becomes feasible to estimate risk by determining the most likely attack pattern stemming from a certain entry point leading up to the anticipated $Goal$.

Given that the most severe event from the perspective of the ICS owner is chosen when setting the $Goal$, we omit further discussion on Impact in this context and focus on quantifying Likelihood. According to the Diamond model, the probability of a successful attack on the "Victim" is determined by the manifestation probability of the "Adversary", the Adversary's capability for successful attack "Capability", and the degree of difficulty stemming from the "Infrastructure". Based on this model, the probability of success for an attack scenario can be formulated as equation (26).

$$\boldsymbol{Likelihood} = \boldsymbol{P_{ThreatActor}} \times \prod_{k=0} \boldsymbol{\varphi_{C,k}\varphi_{I,k}} \qquad (\mathbf{26})$$

Here, $\boldsymbol{P_{ThreatActor}}$ represents the manifestation probability of the threat actor. $\boldsymbol{\varphi_{C,k}}$ is the probability of a successful attack at $AttackState_k$ based on the threat actor's attack capability, while $\boldsymbol{\varphi_{I,k}}$ denotes the probability of a successful attack at $AttackState_k$ dependent on the characteristics of the attack execution point.

Next, we discuss the numerical indicators for each parameter. $\boldsymbol{P_{ThreatActor}}$ represents the manifestation probability of the attacker and is a value determined based on historical records. $\boldsymbol{\varphi_{C,k}}$ is an element based on the attacker's capabilities in phase $\boldsymbol{k}$ and corresponds to the proficiency with which the attacker can utilize specific attack techniques.

The $\boldsymbol{\varphi_{I,k}}$ is considered the potential variations in attack difficulty based on the nature of the targeted system. For instance, even for the same type of attack, the complexity of mounting a successful assault may vary between host devices, such as Windows machines, and embedded devices running on real-time operating systems. As an illustrative example, a canonical DoS attack, the flood attack, is often more effective against embedded devices than host devices. Conversely, attacks that involve executing arbitrary code on a device, seizing control of it, and subsequently performing further malicious actions may find host devices with abundant computational resources and auxiliary functions more susceptible. Therefore, for the same attack pattern, if the targeted system is more prone to succumbing to an attack, $\boldsymbol{\varphi_{I,k}}$ has a higher value. If the targeted system is less susceptible, $\boldsymbol{\varphi_{I,k}}$ assumes a lower value.

Next, using the target system described in the previous chapter as an example, we identify the kill chain and calculate the probabilities using specific numbers. To reiterate, using the target system highlighted in the prior chapter as our case, we identify the kill chain and compute the probabilities by assigning specific numerical values. Initially, we define the values for $\boldsymbol{P_{ThreatActor}}$, $\boldsymbol{\varphi_{C,k}}$, and $\boldsymbol{\varphi_{I,k}}$ for each parameter. In this instance, we assume a hypothetical threat actor. In this study, we hypothesized that APTxx has a manifestation probability of 30% ($\boldsymbol{P_{ThreatActor} = 0.3}$).

Subsequently, we determine $\boldsymbol{\varphi_{C,k}}$ and $\boldsymbol{\varphi_{I,k}}$. Numerous discussions and studies have been conducted regarding the quantification of vulnerability to attacks [31][32] with one of the most standardized metrics being the Common Vulnerability Scoring System (CVSS) [33]. In this context, we utilize the values from the standard model that indicates vulnerability severity, specifically the CVSS version 3 (CVSSv3). CVSSv3 prescribes a numerical value known as Exploitability as an indicator of the potential success of an attack. The formula for Exploitability is represented as equation (27).

$$\boldsymbol{Exploitability} = \mathbf{8.22} \times \boldsymbol{AV} \times \boldsymbol{AC} \times \boldsymbol{PR} \times \boldsymbol{UI} \qquad (\mathbf{27})$$

For simplicity in this study, we utilize normalized AC values for the evaluation of both $\boldsymbol{\varphi_{C,k}}$ and $\boldsymbol{\varphi_{I,k}}$. Specifically, the "High" value is defined as 0.77 divided by 0.44, which equals 0.571, and the "Low" value is defined as 0.44 divided by 0.44, resulting in 1.00. When applying these values, if the complexity evaluation of attacks at each step of the kill chain is all categorized as "Low", then the condition $\boldsymbol{\varphi_{C,k\in\mathbb{N}} = \varphi_{I,k\in\mathbb{N}} = 1.00}$ holds, making the overall complexity of the kill chain 1.0 that is the easiest way for attackers to complete their $Goal$. Based on this rule, we assess the likelihood of a successful attack on the kill chain identified for the evaluation model presented in the previous chapter.

First, we define numerical values for the capability ($\boldsymbol{\varphi_C}$) of APTxx to utilize each available Technique. The defined table is shown as TABLE VII. Here, "Conditions" indicate the conditions under which the capability for each Technique is satisfied. For instance, in the case of the technique T1550 (User Alternate Authentication Material), if the attacker already possesses the credential information of the attack point ($cred_{AttackPoint} \in$

TABLE VIII

The example of attack scenarios generated by the proposed method

| Likelihood | Attack Pattern |
|---|---|
| 0.1713 | (#0 Phase:TA0001_2 Point:Internet Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1566)<br>➞ (#1 Phase:TA0002 Point:Gateway Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1204)<br>➞ (#2 Phase:TA0007_1 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1087)<br>➞ (#3 Phase:TA0006 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1110)<br>➞ (#4 Phase:TA0007_2 Point:Gateway Target:IT-NW Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1049)<br>➞ (#5 Phase:TA0008 Point:IT-NW Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1210)<br>➞ (#6 Phase:TA0002 Point:Server Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1204)<br>➞ (#7 Phase:TA0007_1 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1087)<br>➞ (#8 Phase:TA0006 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1110)<br>➞ (#9 Phase:TA0007_2 Point:Server Target:OT-NW Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1049)<br>➞ (#10 Phase:TA0040_2 Point:OT-NW Target:PLC Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1499)<br>➞ (END) |
| 0.0182 | (#0 Phase:TA0001_2 Point:Internet Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1133)<br>➞ (#1 Phase:TA0002 Point:Gateway Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1203)<br>➞ (#2 Phase:TA0007_1 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1087)<br>➞ (#3 Phase:TA0006 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1110)<br>➞ (#4 Phase:TA0007_2 Point:Gateway Target:IT-NW Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1040)<br>➞ (#5 Phase:TA0008 Point:IT-NW Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1550)<br>➞ (#6 Phase:TA0002 Point:Server Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1204)<br>➞ (#7 Phase:TA0007_1 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1201)<br>➞ (#8 Phase:TA0006 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1110)<br>➞ (#9 Phase:TA0007_2 Point:Server Target:OT-NW Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1049)<br>➞ (#10 Phase:TA0008 Point:OT-NW Target:PLC Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1550)<br>➞ (#11 Phase:TA0002 Point:PLC Target:PLC Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1204)<br>➞ (#12 Phase:TA0040_1 Point:PLC Target:PLC Cond:{'admin': ['Gateway', 'Server', 'PLC'], 'cred': ['cred_1']} AttackTech:T1499)<br>➞ (END) |
| 0.0059 | (#0 Phase:TA0001_1 Point:Ctrl-Area Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1078)<br>➞ (#1 Phase:TA0002 Point:Gateway Target:Gateway Cond:{'admin': [], 'cred': []} AttackTech:T1203)<br>➞ (#2 Phase:TA0007_1 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1518)<br>➞ (#3 Phase:TA0006 Point:Gateway Target:Gateway Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1555)<br>➞ (#4 Phase:TA0007_2 Point:Gateway Target:IT-NW Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1040)<br>➞ (#5 Phase:TA0008 Point:IT-NW Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1210)<br>➞ (#6 Phase:TA0002 Point:Server Target:Server Cond:{'admin': ['Gateway'], 'cred': []} AttackTech:T1203)<br>➞ (#7 Phase:TA0007_1 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1201)<br>➞ (#8 Phase:TA0006 Point:Server Target:Server Cond:{'admin': ['Gateway', 'Server'], 'cred': []} AttackTech:T1555)<br>➞ (#9 Phase:TA0007_2 Point:Server Target:OT-NW Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1040)<br>➞ (#10 Phase:TA0008 Point:OT-NW Target:PLC Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1550)<br>➞ (#11 Phase:TA0002 Point:PLC Target:PLC Cond:{'admin': ['Gateway', 'Server'], 'cred': ['cred_1']} AttackTech:T1204)<br>➞ (#12 Phase:TA0040_1 Point:PLC Target:PLC Cond:{'admin': ['Gateway', 'Server', 'PLC'], 'cred': ['cred_1']} AttackTech:T1485)<br>➞ (END) |
| 0.0033 | (#0 Phase:TA0001_1 Point:Ctrl-Area Target:HMI-PC Cond:{'admin': [], 'cred': []} AttackTech:T1078)<br>➞ (#1 Phase:TA0002 Point:HMI-PC Target:HMI-PC Cond:{'admin': [], 'cred': []} AttackTech:T1203)<br>➞ (#2 Phase:TA0007_1 Point:HMI-PC Target:HMI-PC Cond:{'admin': ['HMI-PC'], 'cred': []} AttackTech:T1518)<br>➞ (#3 Phase:TA0006 Point:HMI-PC Target:HMI-PC Cond:{'admin': ['HMI-PC'], 'cred': []} AttackTech:T1555)<br>➞ (#4 Phase:TA0007_2 Point:HMI-PC Target:IT-NW Cond:{'admin': ['HMI-PC'], 'cred': ['cred_1']} AttackTech:T1040)<br>➞ (#5 Phase:TA0008 Point:IT-NW Target:Server Cond:{'admin': ['HMI-PC'], 'cred': ['cred_1']} AttackTech:T1021)<br>➞ (#6 Phase:TA0002 Point:Server Target:Server Cond:{'admin': ['HMI-PC'], 'cred': ['cred_1']} AttackTech:T1059)<br>➞ (#7 Phase:TA0007_1 Point:Server Target:Server Cond:{'admin': ['Server', 'HMI-PC'], 'cred': ['cred_1']} AttackTech:T1087)<br>➞ (#8 Phase:TA0006 Point:Server Target:Server Cond:{'admin': ['Server', 'HMI-PC'], 'cred': ['cred_1']} AttackTech:T1555)<br>➞ (#9 Phase:TA0007_2 Point:Server Target:OT-NW Cond:{'admin': ['Server', 'HMI-PC'], 'cred': ['cred_1']} AttackTech:T1040)<br>➞ (#10 Phase:TA0008 Point:OT-NW Target:PLC Cond:{'admin': ['Server', 'HMI-PC'], 'cred': ['cred_1']} AttackTech:T1021)<br>➞ (#11 Phase:TA0002 Point:PLC Target:PLC Cond:{'admin': ['Server', 'HMI-PC'], 'cred': ['cred_1']} AttackTech:T1059)<br>➞ (#12 Phase:TA0040_1 Point:PLC Target:PLC Cond:{'admin': ['Server', 'HMI-PC', 'PLC'], 'cred': ['cred_1']} AttackTech:T1499)<br>➞ (END) |

$\boldsymbol{creds}$), then $\varphi_C$ is "Low". If not, since there is a need to search for available credential information, $\varphi_C$ is "High". The $\varphi_C$ values defined in this study are presented in TABLE VII.

TABLE VII

The list of $\boldsymbol{\varphi_C}$ in each technique

| # | Tactics | Techniques | $\varphi_C$ | Conditions |
|---|---|---|---|---|
| 1 | TA0001/Initial Access (Remote) | T1189 - Drive-by Compromise | High (0.571) | N/A |
| | | T1133 - External Remote Services | High (0.571) | N/A |
| | | T1566 - Phishing | Low (1.00) | N/A |
| 2 | TA0001/Initial Access (Local) | T1078 - Valid Accounts | Low (1.00) | N/A |
| 3 | TA0002/Execution | T1059 - Command and Scripting Interpreter | High (0.571) | N/A |
| | | T1203 - Exploitation for Client Execution | High (0.571) | N/A |
| | | T1204 - User Execution | Low (1.00) | N/A |
| 4 | TA0007/Discovery (Local) | T1087 - Account Discovery | Low (1.00) | N/A |
| | | T1201 - Password Policy Discovery | Low (1.00) | N/A |
| | | T1518 - Software Discovery | Low (1.00) | N/A |
| 5 | TA0006/Credential Access | T1110 - Brute Force | Low (1.00) | N/A |
| | | T1555 - Credentials from Password Stores | Low (1.00) | N/A |
| 6 | TA0007/Discovery (Network) | T1040 - Network Sniffing | High (0.571) | N/A |
| | | T1049 - System Network Connections Discovery | Low (1.00) | N/A |
| 7 | TA0008/Lateral Movement | T1210 - Exploitation of Remote Services | High (0.571) | N/A |
| | | T1021 - Remote Services | High (0.571) | N/A |
| | | T1550 - Use Alternate Authentication Material | Low (1.00) | $cred_{AttackPoint} \in \boldsymbol{creds}$ |
| | | | High (0.571) | N/A |
| 8 | TA0040/Impact (Local) | T1485 - Data Destruction | Low (1.00) | N/A |
| | | T1565 - Data Manipulation | Low (1.00) | N/A |
| | | T1499 - Endpoint Denial of Service | Low (1.00) | N/A |
| 9 | TA0040/Impact (Remote) | T1565 - Data Manipulation | Low (1.00) | N/A |
| | | T1499 - Endpoint Denial of Service | Low (1.00) | N/A |

Next, we define the numerical values for the attack success probability $\varphi_I$ determined by the nature of the infrastructure. $\varphi_I$ is determined for each Technique based on the elements of the infrastructure ( $Component$ , $PhysicalArea$ , $Network$ ). These values serve as a knowledge base, and there is a need to determine values for each entry. For simplifying in this study, if the attack starting point is in the $PhysicalArea$, we define it as "High," and for all other instances, we define it as "Low." In that case, $\varphi_{I,k}$ is defined as equation (28).

$$\varphi_{I,k} = \begin{cases} \mathbf{0.571}, & \boldsymbol{k = 1 \; and \; lo_0 \in PhysicalAre} \\ \mathbf{1.000}, & \boldsymbol{k \geq 1 \; or \; k = 1 \; and \; lo_0 \notin PhysicalArea} \end{cases} \quad (28)$$

Under the conditions described above, we identified possible attack scenarios in the target system presented in the previous section and calculated the Likelihood for each scenario. First, we calculated the total number of possible attack scenarios. The computation conditions took every node as the starting point and exhaustively searched for all possible attack paths (all Simple paths obtained in the graph of the $TargetSystem$) and attack patterns (all patterns of Techniques possible for each attack path). Under these conditions, 679,644 attack scenarios were identified.

Subsequently, we calculated their Likelihood according to the conditions previously mentioned for each obtained scenario. The results are presented in FIGURE 9. Under the conditions of this study, the obtained Likelihood of each scenario is discrete, stemming from the simple discrete model employed for defining $\varphi_C$

and $\varphi_I$. As a result of this, the Likelihood of each kill chain also takes on discrete values. From these findings, it was obtained that 0.037% of the entire set of attack scenarios resulted in the kill chain with the highest confidence.

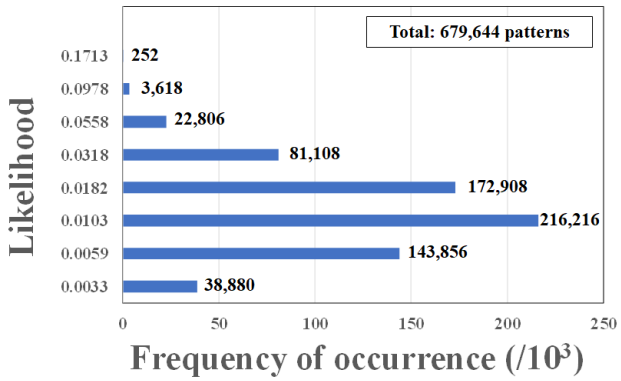


**FIGURE 9. The bar chart showing the frequency of attack scenario occurrence per likelihood in the $TargetSystem$**

Next, a sample of the kill chain is presented in TABLE VIII. TABLE VIII displays the kill chain, randomly selected, for when the Likelihood is {0.1713, 0.0182, 0.0059, 0.0033}. A discernible trend in the obtained kill chains is that the fewer the $AttackState$ and the higher the $\varphi_C$ and $\varphi_I$ values an $AttackState$ possesses, the greater the Likelihood. This tendency is expected. By considering various $ThreatActor$ and its $Goal$, and by modeling the target system using this method, it becomes possible to determine the attack scenarios with the highest likelihood of success. This is achieved by conducting path searches based on graph route exploration and calculating the Likelihood based on $\varphi_C$ and $\varphi_I$ of each technique in the transition conditions of each phase.

Therefore, by structurally describing attack scenarios using this method, it becomes possible to evaluate the feasibility of achieving the $Goal$ by the predefined $ThreatActor$ at the granularity of attack steps, detailed numerically as the Likelihood of each scenario. Consequently, this result allows for the prioritization of detailed scenarios.

On the other hand, in order to obtain more quantitative results, there remains the challenge of further quantifying the values of $\boldsymbol{P_{ThreatActor}}$, $\varphi_C$ and $\varphi_I$. To quantitatively determine $\boldsymbol{P_{ThreatActor}}$, there is a possibility that it can be determined based on threat intelligence and empirical data from a variety of security sensors deployed on the internet [36].

To quantitatively define $\varphi_C$, some form of measurement is necessary. While open knowledge databases like MITRE ATT&CK define the techniques used by $ThreatActor$, they do not publish the success probabilities of each technique. Therefore, there is a need to observe the actual attacks of $ThreatActor$ through systems like IDS, evaluate the methods they are genuinely employing, and determine $\varphi_C$ for each technique. One concrete example could involve observing the techniques used by real $ThreatActor$ through security sensors, or alternatively, simulating attacks using penetration testers with equivalent attack skills based on a hypothetical $ThreatActor$. By empirically testing the success probability of each technique, a more quantitative definition of $\varphi_C$ might be attainable.

Regarding $\varphi_I$, it is possible to define its probability by referencing the ATT&CK framework. For instance, ATT&CK for Enterprise offers Techniques that can be applied on specific platforms such as Windows or Linux. This means that by referencing ATT&CK, if a technique corresponds to a platform, one can assert that the attack is feasible ($\varphi_I$=1). However, like the discussion for $\varphi_C$, the actual success rate is not quantitatively defined and needs to be empirically evaluated. Thus, the quantification of $\varphi_C$ and $\varphi_I$ remains a subject for future work.

The quantification of the Impact which was omitted in this discussion, also presents challenges in how it should be quantitatively expressed. Specifically, it's essential to evaluate the ripple effects of the impact and eliminate ambiguous elements. However, its logical structure is still unclear and debatable for further researches [34][35][36][37].

In this study, the target system evaluated had a limited number of assets, permitting an exhaustive search owing to the employment of a simple $ThreatActor$ with low capabilities. However, when the target system has many assets, and a high capability $ThreatActor$ model is employed, there's a risk of computational explosion. Computational complexity will be discussed in the following section.

### *E. DISCUSSIONS ON COMPUTATION*

We consider the effect of increasing the number of elements in the model. When the number of constituent elements in $TargetSystem$ and $ThreatActor$ is increased, the number of possible combinations in each attack phase also increases. The computational feature affects the computational time to identify rational attack scenarios from starting points to the end point of adversaries mechanically based on the proposed model. In this study, we evaluated the computational time shown above under the following assumptions.

i. A system for the evaluation is based on the system described in the section "A. MODELS FOR EVALUATION" in this

chapter.

ii. To simplify the evaluation, networks in the system are fixed at the three instances shown in TABLE IV, and only the number of components connected to each network is increased.

iii. The unit of increasing the number of components in the system is a set consisting of 7 instances of the components listed in TABLE III, and the system scale is increased by the set unit multiplied by the natural number of components (N times).

iv. In the generation process of attack scenarios, an adversary, defined in section III.A attempts brute-force attacks against each component that exists on the same network. Attack steps are continued until the final target of the threat actor is reached or no further attack methods available for the adversary.

We have evaluated the computation processing time of the proposed method under the above assumptions. In this evaluation, we have measured the computational processing time required for the identification of attack scenarios comprehensively and the generation of a list of them for target systems with various number of components. FIGURE 10 shows a graph of the scale of target systems (x-axis) versus computation processing time (y-axis). Especially, the x-axis indicates the number of components in a target system, and the y-axis indicates the measured computation time of the above process. Each plot in FIGURE 10 has the average computation time obtained from ten experiments. The dotted line in the graph indicates an exponential approximated curve calculated from the plots ($R^2 \sim 0.9767$). The computation environment is the following:

- CPU: Intel Corei7-8665U 2.11GHz (4 cores/8 threads)
- Memory: DDR4 32GB memory
- OS: Windows 10 Pro (Ver 10.0.19042.1237)
- Programming language: Python3.9
- Graph calculation module: NetworkX 3.1

Based on the result, the computation time increases according to the number of components in a target system by exponential order approximately. It means that a large-scale $TargetSystem$ is liable to cause a combinatorial explosion so that processing cannot be completed in real time. The computation time depends on not only the number of components but also on the network structure, attributes of $ThreatActor$ and attributions which components have. However, the general trend is that as the number of components increases, the computational complexity increases in exponential order, unless search algorithms are devised.

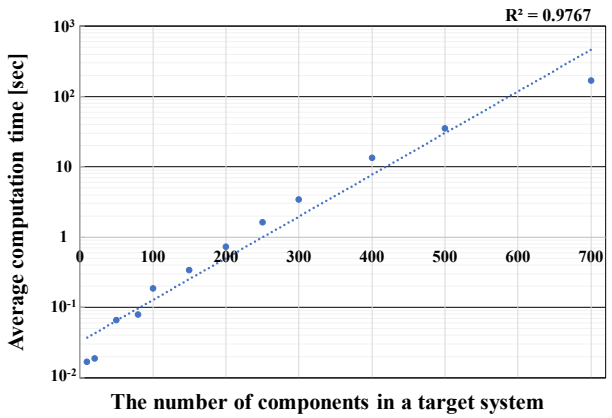


**FIGURE 10. A graph of the average computation processing time versus the number of components in a target system (The logarithmic scale)**

In prior research, there has been an investigation into the computation time for generating attack graphs in relation to the number of nodes, $n$ [38]. While the computational complexity varies depending on the method, the computational order for generating the attack graph ranges from $O(2^n)$ [39][40] to, at the very least, $O(nlog(n))$ [41]. Although these methods are based on models different from the proposed approach, since our method also derives attack scenarios through graph exploration, it is anticipated that our approach will exhibit similar computational complexity. In the implementation in this study, as observed in FIGURE 10 with the increasing trend in computational time, it is hypothesized that the computational complexity lies between $O(2^n)$ and $O(n^m)$ when $m$ is set to any positive integer. Consequently, by ingeniously designing the graph exploration algorithm based on the model assumed in our proposed method, there is potential to reduce the computational complexity.

One possible way to deal with this problem is to weigh the attack patterns chosen by the adversary in each attack phase according to previous cases and the intermediate state that the attacker is in. For example, starting by defining a $ThreatActor$ decision-making model based on statistics of past cases and game theory [42]. This model can then be used to determine the weighting of each attack pattern according to the state of $ThreatActor$. In this way, by using statistics of past cases and decision models of adversarial behavior, it may be possible to extract only those attack scenarios that require priority attention. For one of future works, we study how to avoid this combinatorial explosion in a large-scale $TargetSystem$.

### F. RELATED WORKS AND BENCHMARK

In this section, we discuss research related to the proposed method and benchmarks pertaining to the same. Our methodology addresses

TABLE IX
The benchmark related to the proposed method in this study

| Work | Objective | Reusability | Express detailed system structure | Define adversary's capabilities | Evaluate dynamic actions | Evaluate likelihood |
|---|---|---|---|---|---|---|
| This study | Determine priority scenarios | ✓ | ✓ | ✓ | ✓ | ✓ |
| G. Falco, et al.[12] | Comprehensive threat identification | ✓ | - | - | - | - |
| G. Chu, et al.[46] | Attack simulation | ✓ | - | - | ✓ | - |
| P. Johnson et al.[45] | Describe threat event | ✓ | ✓ | - | - | - |
| R. Khan et al.[50] | Comprehensive threat identification | ✓ | - | - | - | - |
| P. Johnson et al.[51] | Automatic threat identification | ✓ | ✓ | - | - | - |
| A. Ekelhart et al.[52] | Attack simulation | ✓ | ✓ | - | - | - |
| M. Mohsin et al.[48] | Identify likelihood of attack scenarios | ✓ | ✓ | ✓ | - | ✓ |

the ontological representation of attack scenarios on ICS. Drawing upon the Diamond model, it models both attackers and target systems (victims). On the modeled target system, attacker actions are structurally represented as transitions within a state model. The capabilities of the attacker are defined based on knowledge bases such as ATT&CK. For ICS, based on a graph-oriented attack path and the success or failure of attacks at each point, we determine the feasibility of the said attacker reaching the final goal, thereby deciding the priority of attack scenarios.

In relation to the proposed method, there have been reports on techniques aimed at revealing the ontology of threat representations for systems, encompassing not only ICS but also IT systems and IoT and mechanically deriving attack scenarios or attack trees. A common approach in these studies involves defining relational models of various elements based on standard vulnerabilities and attack techniques, such as ATT&CK, CVE, and CAPEC. These models then serve as the foundation to represent attack scenarios as a sequence of discrete events logically articulated [12][43][44]. In terms of knowledge modeling, proposals have emerged that define linguistic models for cyber-attack events and represent cyber-attacks using these language models [45]. Moreover, efforts to simulate attacker behaviors, as seen in penetration tests, aim to clarify the ontology of intrusion actions onto systems, subsequently simulating the movements of the attacker based on this understanding [46]. There are also endeavors to quantitatively represent risks utilizing standardized metrics like CVSS [46]. Additionally, some research reports introduce probabilistic quantitative metrics to elements of attack scenarios, expressing them using structural models like Bayesian networks or Markov models [48][49].

TABLE IX describes benchmarks of the proposed method in comparison with several existing studies. These prior research endeavors universally aim to mechanically identify threat scenarios or attack scenarios and have reported on the logical representation of threat events. While the objectives they seek to achieve differ, each method aims to logically represent threat events and scientifically derive rational attack scenarios. A common feature among these methods is ensuring the reusability of threat events by logically modeling threat or attack scenarios.

The extent to which each method models the target system varies. In our approach, we require detailed modeling of the target system, representing it as a graph model with various attributions. Additionally, our method determines the success or failure of an attack based on the attacker's capabilities. This approach is akin to the method proposed by M. Mohsin et al. [48], which probabilistically represents the attacker's capabilities, attempting to provide a quantitative risk representation.

In contrast to their method, which emphasizes modeling discrete and independent attack actions, our approach integrates the state changes induced by each attack on the modeled target system, reflecting the dynamic nature of the attacker's behavior. Our approach distinguishes itself by identifying and evaluating attack scenarios based on this. As a result, it is better suited to structurally represent more realistic attacker behaviors and eliminate personal judgement as well.

## IV. CONCLUSION

In this study, we have established a method for representing threat events in a reusable format and distinguishing these events while eliminating differences of personal judgment from threats that must be dealt with preferentially. To achieve this aim, we have studied a structural representation model of threat events. This proposed model represents attack scenarios based on the Diamond model and the attack intermediate state model.

In this proposed model, we have defined a system model and a threat actor model for the ICS system subject to security assessment, and the possibility of reaching the goal of the threat actor is evaluated based on the defined contents. This model provides capabilities to represent attack scenarios structurally in a reusable form, and the attack scenarios in a detailed and objective form. This approach enables us to identify the attack scenarios that should be dealt with preferentially while eliminating personal judgement.

We have evaluated and discussed the usefulness of the proposed model in terms of the reusability of threat events and the prioritization of threats. In terms of the reusability of threat events, we have shown that reusability can be ensured by creating templates of threat events and attack conditions based on past cases and on standards defined by ATT&CK and the like. Consequently, it enables the determination of attack scenarios and prioritizing qualitatively eliminating personal judgement.

Aiming to eliminate personal judgement in assessing the likelihood of reaching the goal of the $ThreatActor$, we have attempted to quantify security risks based on the proposed model. As a result, we have shown that the proposed model enables a certain degree of personal judgment to be eliminated. However, the parameters namely $\boldsymbol{P}_{\boldsymbol{ThreatActor}}$, $\varphi_{C,k}$ and $\varphi_{I,k}$ are remain ambiguous and can still be subject to personal judgment. To fully eliminate personal judgment in prioritizing threat events quantitatively, it is essential to determine these elements in an objective manner. Therefore, it is necessary to determine their values through empirical methods. Establishing a quantitative method for determining these parameters remains a challenge.

In addition, when considering more complex adversarial models or system models, the impact of computational requirements cannot be overlooked. Especially under such premises, reducing these computational complexities and devising mechanisms to prevent combinatorial explosion are subjects for future research.

In future research, our focus only be on eliminating factors of personal judgment in quantitative methodologies but also on refining the logic used to assess adversarial behavior concerning attack cost-efficiency. Moreover, we intend to model potential adversary actions to prevent combinatorial explosions.

## REFERENCES

[1] J. Slowik, "Evolution of ICS Attacks and the Prospects for Future Disruptive Events," Dragos Writepaper, Feb. 2019. Available: https://www.dragos.com/wp-content/uploads/Evolution -of-ICS-Attacks-and-the-Prospects-for-Future-Disruptive-Events-Joseph-Slowik-1.pdf

[2] D. Bhamarea, M. Zolanvaric, A. Erbadb, R. Jainc, K. Khanb and N. Meskind, " Cybersecurity for industrial control systems: A survey," Computers & Security, Vol. 89, 101677, Feb. 2020.

[3] K. Sawada, "Model-based cybersecurity for control systems: Modeling, design and control," Proceedings of 2017 56th Annual Conference of Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), IEEE, Nov. 2017.

[4] A. Shostack, "Threat modeling: Designing for security," Wiley Feb. 2014.

[5] IPA, "Security Risk Assessment Guide for Industrial Control Systems 2nd Edition," Mar. 2020. Available: https://www.ipa.go.jp/security/controlsystem/riskanalysis.html (in Japanese)

[6] N. Matsumoto, J. Fujita, H. Endoh, T. Yamada, K. Sawada and O. Kaneko, "Asset Management Method of Industrial IoT Systems for Cyber-Security Countermeasures," MDPI Information 2021, Vol. 12, pp. 460, Nov. 2021. Available: https://doi.org/10.3390/info12110460

[7] Y. Hashimoto, T. Toyoshima, S. Yogo, M. Koike, T. Hamaguchi, S. Jing and I. Koshijima, "Safety securing approach against cyber-attacks for process control system," Computers & Chemical Engineering, Vol. 57, pp. 181–186, Oct. 2013.

[8] C. Haley, R. Laney, J. Moffett and B. Nuseibeh, "Security requirements engineering: A framework for representation and analysis," IEEE Transactions on Software Engineering, Vol. 34, Issue 1, pp. 133–153, Jan. 2018

[9] Lockheed Martin Corp., "Cyber Kill Chain®," https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html, Accessed on Nov. 2021.

[10] D. Miessl, "Securing the Internet of Things: Mapping Attack Surface Areas Using the OWASP IoT Top 10," RSA Conference 2015, ASD-T10, Jul. 2015.

[11] MITRE Corp., CAPEC™, https://capec.mitre.org/, Accessed on Nov. 2021.

[12] G. Falco, A. Viswanathan, C. Caldera and H. Shrobe, "A Master Attack Methodology for an AI-Based Automated Attack Planner for Smart Cities," IEEE Access, Vol. 6, pp. 48360–48373, Aug. 2018.

[13] JASE, "JASO, TP15002: Guideline for Automotive Information Security Analysis," Jan. 2016.

[14] IEC: "Industrial communication networks - Network and system security - Part 2-1: Establishing an industrial automation and control system security program," IEC 62443-2-1:2010, Nov. 2010.

[15] L. Gallon and J.J. Bascou, "Using CVSS in attack graph," 2011 Sixth International Conference on Availability, Reliability and Security, pp. 59–66, Aug. 2011.

[16] Y. Kawanishi, H. Nishihara, D. Souma, H. Yoshida and Y. Hata, " A Comparative Study of JASO TP15002-Based Security Risk Assessment Methods for Connected Vehicle System Design," Hindawi Security and Communication Networks, Vol. 2019, Article ID 4614721, Feb. 2019.

[17] Y. Kawanishi, H. Nishihara, D. Souma, H. Yoshida and Y. Hata, "A Study on Quantitative Risk Assessment Methods in Security Design for Industrial Control Systems," Proceedings of 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, pp. 62–69, Aug. 2018.

[18] P. Mell, K. Scarfone and S. Romanosky, "A Complete Guide to the Common Vulnerability Scoring System Version 2.0," Jan. 2007. Available: https://www.first.org/cvss/v2/cvss-v2-guide.pdf

[19] MITRE Corp. "Common Weakness Scoring System (CWSS™) Version 1.0.1," Sep.2014. Available: https://cwe.mitre.org/cwss/cwss_v1.0.1.html, Accessed on Nov. 2021.

[20] C., Sergio P. Andrew and B. Christopher, "The Diamond Model of Intrusion Analysis," DTIC Document, Technical Report, Jul. 2013.

[21] A.J. van der Schaft, "Achievable behaviors of general systems," Systems & Control Letters, Vol. 49, pp. 141–149, Jun. 2003.

[22] MITRE Corp., ATT&CK®, https://attack.mitre.org/, Accessed on Nov. 2021.

[23] MITRE Corp., ATT&CK® for Industrial Control Systems, https://collaborate.mitre.org/attackics/index.php/Main_Page, Accessed on Nov. 2021.

[24] MITRE Corp., "Dragonfly," ATT&CK®, https://attack.mitre.org/groups/G0035/, Accessed on Nov. 2021.

[25] MITRE Corp., "Group: Dragonfly 2.0, Berserk Bear, DYMALLOY, "ATT&CK® for Industrial Control Systems, https://collaborate.mitre.org/attackics/index.php/Group/G0006, Accessed on Nov. 2021.

[26] MITRE Corp., "Lazarus Group," ATT&CK®, https://attack.mitre.org/ groups/G0032/, Accessed on Nov. 2021.

[27] CISA: "Alert (AA21-131A) DarkSide Ransomware: Best Practices for Preventing Business Disruption from Ransomware Attacks," https://www.cisa.gov/uscert/ncas/ alerts/aa21-131a, May 2021

[28] T. D. Wagner, K. Mahbub, E. Palomar and A. E. Abdallah: "Cyber threat intelligence sharing: Survey and research directions," Computers & Security, Vol. 87, Nov. 2019.

[29] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby and K. Stoddart: "A review of cyber security risk assessment methods for SCADA systems," Computers & Security, Vol. 56, pp. 1-27, Feb. 2016.

[30] ISO 31000:2018 "Risk management – Guidelines," Feb. 2018.

[31] S. Ali and R. W. Anwar, "Trused: a trust-based security evaluation scheme for a distributed control system," Computers, Materials & Continua 2023, Vol. 74, No.2, pp. 4381–4398, Oct. 2022.

[32] M. Battaglioni, G. Rafaiani, F. Chiaraluce and M. Baldi, "MAGIC: A Method for Assessing Cyber Incidents Occurrence," in IEEE Access, Vol. 10, pp. 73458-73473, Jul. 2022.

[33] FIRST, "Common Vulnerability Scoring System," https://www.first.org/cvss/, Accessed on Nov. 2021.

[34] I. Agrafiotis, J. R. C. Nurse, M. Goldsmith, S. Creese and D. Upton, "A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate," Journal of Cybersecurity, Vol. 4, Issue 1, Oct. 2018.

[35] E. Haapamäki and J. Sihvonen, " Cybersecurity in accounting research," The Managerial Auditing Journal, Vol. 34, Issue 7, Jul. 2019.

[36] D. Woods and L. Walter, "Reviewing Estimates of Cybercrime Victimisation and Cyber Risk Likelihood," in 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 150-162, Jun. 2022.

[37] F. Cremer, B. Sheehan, M. Fortmann, A. N. Kia, M. Mullins, F. Murphy and S. Materne, " Cyber risk and cybersecurity: a systematic review of data availability," Geneva Pap Risk Insur Issues Pract 47, pp. 698-736, Feb. 2022.

[38] T. Cody, "A Layered Reference Model for Penetration Testing with Reinforcement Learning and Attack Graphs," in 2022 IEEE 29th Annual Software Technology Conference (STC), pp. 41-50, Oct. 2022.

[39] C. Phillips and L. P. Swiler, "A graph-based system for network vulnerability analysis," in Proceedings of the 1998 Workshop on New Security Paradigms, pp. 71–79, Jan. 1998.

[40] O. Sheyner, J. Haines, S. Jha, R. Lippmann and J. M. Wing, "Automated generation and analysis of attack graphs," in

Proceedings 2002 IEEE Symposium on Security and Privacy. IEEE, pp. 273–284, May 2002.

[41] K. Ingols, R. Lippmann, and K. Piwowarski, "Practical attack graph generation for network defense," in 2006 22nd Annual Computer Security Applications Conference (ACSAC'06). IEEE, pp. 121–130, Dec. 2006.

[42] S. Shiva, S. Roy and D. Dasgupta: "Game theory for cyber security," Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research, Apr. 2010.

[43] L. Singels, C. Biebuyck and L. Malukele, "A Formal Concept Analysis Driven Ontology for ICS Cyberthreats," in Southern African Conference for Artificial Intelligence Research (SACAIR) 2020 Proceedings: Knowledge Representation and Reasoning, pp.247-263, Dec. 2020.

[44] M. A. Jarwar, J. Watson, U. D. Ani and S. Chalmers, "Industrial Internet of Things Security Modelling using Ontological Methods," in Proceedings of the 12th International Conference on the Internet of Things (IoT'22), pp.163-170, Nov. 2022.

[45] P. Johnson, R. Lagerström and M. Ekstedt, "A Meta Language for Threat Modeling and Attack Simulations," Proceedings of the 13th International Conference on Availability, Reliability and Security (ARIS) No.38, pp.1–8, Aug. 2018.

[46] G. Chu and A. Lisitsa, "Ontology-based Automation of Penetration Testing," In Proceedings of the 6th International Conference on Information Systems Security and Privacy (ICISSP 2020), pp.713–720, Feb. 2020.

[47] A. I. Newaz, A. Aris, A. K. Sikder and A. S. Uluagac, "Systematic Threat Analysis of Modern Unified Healthcare Communication Systems," GLOBECOM 2022-2022 IEEE Global Communications Conference, pp.1404-1410, Dec. 2022.

[48] M. Mohsin, M. U. Sardar, O. Hasan and Z. Anwar, "IoTRiskAnalyzer: A Probabilistic Model Checking Based Framework for Formal Risk Analytics of the Internet of Things," in IEEE Access, Vol.5, pp.5494-5505, Apr. 2017.

[49] M. A. Bode, S. A. Oluwadare, B. K. Alese and A. F. -B. Thompson, "Risk analysis in cyber situation awareness using Bayesian approach," 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), pp.1-12, Jun. 2015.

[50] R. Khan, K. McLaughlin, D. Laverty and S. Sezer, "STRIDE-based threat modeling for cyber-physical systems," 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), pp.1-6, Sep. 2017.

[51] P. Johnson, A. Vernotte, M. Ekstedt and R. Lagerström, "pwnPr3d: An Attack-Graph-Driven Probabilistic Threat-Modeling Approach," 2016 11th International Conference on Availability, Reliability and Security (ARES), pp.278-283, Aug. 2016.

[52] A. Ekelhart, E. Kiesling, B. Grill, C. Strauss and C. Stummer, "Integrating attacker behavior in IT security analysis: a discrete-event simulation approach" Information Technology and Management, Vol.16, pp.221–233, Jun. 2015.

**JUNYA FUJITA** (M'19) received his Master of Engineering from the University of Tokyo in 2011. He is a senior researcher at Hitachi, ltd. and has experience in R&D related to embedded systems and cybersecurity for industrial automation control applications over 10 years. He is a member of SICE, ISA and IEEE. He is also a member of ISA99 workgroup and an expert member of IEC/TC 65/WG 10 as standardization bodies. He holds world-recognized cybersecurity certifications, such as CISSP, CISA, GICSP and OSCP.

**TAKASHI OGURA** received his M.S. and Ph.D. degrees from Nagoya University, Japan, in 2016 and 2019. Since 2019, he has been working at Research & Development Group in Hitachi Ltd. His research interests are in security, especially for industrial control systems.

**KAZUYA OKOCHI** received his Master's degree from the Graduate School of Engineering, Okayama University in 1999. He joined Hitachi, Ltd. at the same year. After studying security technology and architecture for critical infrastructure systems, security design and development for the systems, he is currently engaged in security-related business development in Security Business Incubation Division in Hitachi. CISSP.

**NORITAKA MATSUMOTO** received his B.E. and M.E. degrees in mechanical engineering from Waseda University, Japan, in 1999 and 2001. He currently works at the R&D group of Hitachi Ltd. He has experience in R&D on industrial control system technologies, particularly embedded systems and cybersecurity. He is a member of the national committee of IEC TC65/WG10 and WG20. He is also a member of IEICE and IPSJ.

**KENJI SAWADA** (M'11) received his Ph.D. degree in engineering in 2009 from Osaka University. He is an Associate Professor in Info-Powered Energy System Research Center, The University of Electro-Communications, Japan. He is also an advisor of Control System Security Center since 2016. He received Outstanding Paper Awards from FA Foundation (2015 and 2019), Fluid Power Technology Promotion Foundation (2018), and JSME (2018). His research interests include the control theory of cyber-physical systems and control system security. He is a member of SICE, ISCIE, IEICE, IEEJ, JSME, and IEEE.

**OSAMU KANEKO** (M'97) received his Ph.D. degree in engineering in 2005 from Osaka University. He is a Professor in the Department of Mechanical and Intelligent Systems Engineering, The University of Electro-Communications, Japan. He received Outstanding Pioneer Technology Award from Control Division, SICE (2015) etc. His research interests include systems control theory and its applications. He is a member of SICE, ISCIE, IEEJ, JSME, and IEEE.