applied sciences

MDPI

Article

Security Requirement Recommendation Method Using Case-Based Reasoning to Prevent Advanced Persistent Threats

Ji-Wook Jung [1] and Seok-Won Lee [1,2,*]

[1] Department of Artificial Intelligence, Ajou University, Suwon 16499, Republic of Korea
[2] Department of Software and Computer Engineering, Ajou University, Suwon 16499, Republic of Korea
* Correspondence: leesw@ajou.ac.kr; Tel.: +82-31-219-3548

Abstract: As the world becomes digitized and connected, cyberattacks and security issues have been steadily increasing. In particular, advanced persistent threats (APTs) are actors who perform various complex attacks over the long term to achieve their purpose. These attacks involve more planning and intelligence than typical cyberattacks. Many studies have investigated APT detection and defense methods; however, studies on security requirements that focus on non-technical factors and prevention are relatively few. Therefore, this study aims to provide attack information to users obtained by analyzing attack scenarios as well as security requirements to help the users understand and make decisions. To this end, we propose a method for extracting attack elements by providing users with templates for attack scenarios with different levels of abstraction. In addition, we use a problem domain ontology that is based on the concept of a case to provide users with attack analysis results and recommended security requirements. Our method uses case-based reasoning to retrieve similar cases, recommend reusable security requirements, and propose revision directions. The ontology can be improved by adding the solution to the problem as a new case. We conducted case studies and surveys to evaluate our methods and showed that they help specify security requirements.

Keywords: advanced persistent threat; security requirement; problem domain ontology; case-based reasoning; artificial intelligence; recommendation system

check for updates

Citation: Jung, J.-W.; Lee, S.-W. Security Requirement Recommendation Method Using Case-Based Reasoning to Prevent Advanced Persistent Threats. *Appl. Sci.* **2023**, *13*, 1505. https://doi.org/10.3390/app13031505

Academic Editor: Arcangelo Castiglione

Received: 15 November 2022
Revised: 14 January 2023
Accepted: 19 January 2023
Published: 23 January 2023

1. Introduction

The Internet has become an essential technology for human activities such as work, transactions, communication, community interaction, and information collection. These activities may require personal information, accounts, and company assets. However, cyberattacks and security breaches by unspecified threat groups to steal vital information and resources are steadily increasing. According to a report by Verizon, 5258 data breaches occurred in 16 industries and 4 regions worldwide in 2021 [1], which is an increase of approximately 33% compared with the 3950 cases that occurred in 2020. In particular, there have been active cyberattacks that have taken advantage of the changes in society caused by the COVID-19 pandemic, and security experts have found that these incidents were based on phishing scams or violations of security policies [2]. Such attacks employ social engineering, which is difficult to defend against even if technical security is perfect because social engineering is a non-technical method. This is one of the characteristics of advanced persistent threats (APTs), which have long been established as a critical security issue [3].

However, although methods to detect and defend against APTs have been studied, security issues continue to arise in industry and in governments. The causes of such security issues are as follows. First, most research on defending against cyberattacks focuses on a technical perspective. Such research is about techniques, such as network intrusion and malware detection, that are deployed in a system. Although this approach is common, it can be weak when faced with a new malicious code. Because it is difficult to develop a perfect defense method, precise and effective security requirement specifications are needed. However, the requirement specification process is time-consuming and expensive, and hence a method to efficiently derive security requirements is needed. Second, APTs also utilize social engineering elements, and there is no structure for analysis structures that includes non-technical elements. As mentioned before, most research focusing on technical solutions considers attack methods and mechanisms but does not consider non-system factors such as attack targets, attack goals, and security policies. Therefore, this study proposes a method to recommend security requirements by building an integrated knowledge base called the case-based problem domain ontology (CB-PDO) to address this problem. To recommend security requirements, we define all elements associated with APT characteristics in the APT component and analyze their relationships to build a knowledge base that is integrated into one abstract level. Furthermore, to derive recommended specifications, we use a case-based reasoning (CBR) method, which is a classical artificial intelligence technique that imitates human problem solving by solving new problems based on previous similar cases. We retain past problems that have already been solved, using them as "cases" to solve current problems, and continuously accumulate a knowledge base to represent learning. We used two methods to evaluate the proposed method: first, we conducted a case study to evaluate whether the proposed method met our research objectives. Second, we performed security expert verification and validation to evaluate whether our research was practically effective and helpful. Each quality was evaluated using a questionnaire. The results show that the proposed method effectively recommends security requirements to security experts. Furthermore, we show that our method can become a more sophisticated recommendation system with the continuous accumulation of cases.

The remainder of this paper is organized as follows: Section 2 describes related work on APT attack analysis, security, and CBR. Our proposed method is presented in Section 3, and the verification and evaluation results are presented in Section 4. Finally, the conclusions, limitations, and future work are presented in Section 5.

2. Related Work

2.1. APT Analysis

APTs were first described in 2006 by United States Air Force analysts [4], who distinguished between the characteristics of an APT and existing cyber threats. An APT has a clear goal for a specific target instead of an unspecified number of victims. The characteristics of an attack are as follows [5]:

- Advanced: Attackers can evade detection and access the network, adapt well to specific network environments, and have many attack tools.
- Persistent: It is difficult for defenders to protect themselves against persistent threats. After an attacker gains access to the system network, it becomes very difficult for the user to interrupt or remove that access.
- Threat: An attacker has the ability to access stored sensitive information.

The APT attack phase has been studied in the literature. Lockheed Martin, a USA munitions company, coined the security term "kill chain," and their framework consists of a seven-phase model to identify and prevent cyber intrusion activities [6,7]. The phases are "reconnaissance," "weaponization," "delivery," "exploitation," "installation," "command and control," and "actions on objects," depending on what the attack group needs to accomplish for its goals. This cyber-kill-chain model can identify the purpose of the attacker or the characteristics of the attack step-by-step and enables the sequence of each phase to be tracked. However, it is difficult to prepare detailed countermeasures against internal attacks because this model does not contain specific concepts for such attacks. Attack techniques related to persistence and defense evasion may be included in internal attacks. Therefore, there is a limit to presenting these within each phase [8].

Unlike the cyber-kill-chain model, the MITRE ATT&CK framework analyzes real attacks and defines 12 attack categories [8,9]. The attack elements defined by the ATT&CK framework are not confined to a specific order of operations and have the advantage of being able to specifically represent behaviors. For example, techniques that can be implemented after intrusions into a network can be classified into categories such as discovery, persistence, privilege escalation, and defense evasion. APTs can be represented by several models depending on the perspective, and each model focuses on a different aspect. Furthermore, because the methods for representing APTs are diverse and complex, rather than adopting one model, a model based on an integrated knowledge system that considers several models is a better approach.

2.2. Security Requirements for APT

The specification of security requirements is a preventive solution to minimize the damage caused by cyberattacks. To facilitate these specifications, it is necessary to have a good understanding of the concepts of the attack elements and their relationship. However, it is difficult to understand and manage security requirements because they are represented by multiple documents with different levels of abstraction. To address this issue, a method was proposed for building a problem domain ontology (PDO) which facilitates understanding of the problems in a target domain and systematically represents the relevant elements [10]. This method was originally used to build a PDO for regulatory documents created by the Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP). The aim was to extract and express the characteristics and constraints of distributed security requirements from multiple documents. Furthermore, the authors demonstrated the possibility of supporting decision-making activities using a PDO.

A method for recommending security requirements using a PDO was also proposed by Kim et al. [11]. In this study, a three-layer PDO was designed, where the layers represented the steps needed to effectively build diverse complex attacks and distribute resources. Scattered concepts and resources were classified and refined using the three layers and then integrated into one PDO as understandable and usable knowledge. The PDO was integrated into the general-purpose knowledge (GPK) base, which constitutes general knowledge, and the domain specific knowledge (DSK) base, including domain business models and objectives. Kim et al.'s method analyzes the vulnerabilities of common attacks from PDOs and performs a risk assessment to recommend derived security requirements.

A method was proposed to recommend effective security requirements for APTs by adding an APT knowledge (APTK) base to the proposed PDO structure [12]. It defines the elements that constitute the APT and represents their relationships. An APTK base can be used to analyze the attack pattern of an APT and derive security requirements suitable for defending against the attack. The knowledge bases proposed in this paper are based on the structure of these knowledge bases. We further propose a new revised base and process that can be managed and evolved to incorporate many cases.

2.3. CBR

CBR was proposed by Schank et al., and it led to significant advances in the field of artificial intelligence [13]. CBR imitates an individual's approach to solving new problems based on previously experienced problems. Each experience is represented as a unit called a "case" and its solution is stored and used to solve current problems [14]. Thus, CBR recognizes knowledge as experience, and that knowledge consists of encapsulated case libraries [15].

The CBR cycle consists of four stages: case retrieval, case reuse, case revision, and case retention (Figure 1). In the case retrieval stage, the most similar previous cases are retrieved and compared with the current problem. To find similar cases, we must extract the attributes or features that can be used to measure similarity and have a good understanding of the domain of the problem. Therefore, appropriate similarity measurement and matching algorithms are used for the features of each domain and the attributes of the cases. For example, in an ontology, matching algorithms can be applied to the features of instances in the domains to measure the similarities between the instances. In addition, similarity can be calculated using various measurements such as rule-based or concept similarity measurements [16,17]. To solve the current problem, CBR enters the case reuse stage, which reuses the solutions of retrieved cases. If necessary, the solutions are revised by modifying the retrieved cases or their solutions to develop a more optimized solution. After the case revision stage, the validated solution is stored as a case that can be used for future problems.

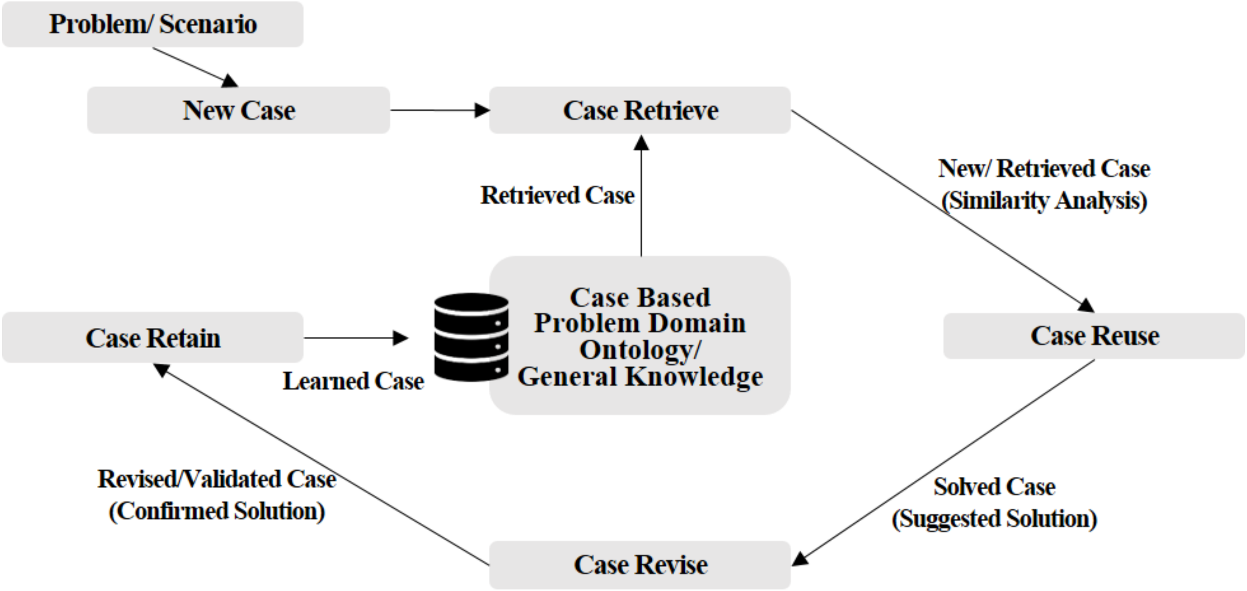


Figure 1. Cycle of CBR (adapted from [18]).

The CBR method is relatively simple and can be applied in fields in which the problem area is not standardized. One such field is recommendation systems, in which CBR is used because human interests and tastes are diverse and cannot be expressed by rules. Hernández-Nieves et al. proposed an application that recommends financial products using CBR methods. A method for measuring similarity according to each element was proposed by establishing the necessary assets and information according to the target area of finance [19]. Matching algorithms and heuristic experiments were used to set thresholds and measure similarities according to the data characteristics. Furthermore, the K-nearest neighbor method was used to reuse the solutions of the retrieved cases and solution revisions were based on the user's subjective judgment. Bokolo introduced a recommendation system using CBR for sustainable smart city development [20]. They selected features to use in the search for specified development plan cases, adopted comparative and Levenstein distance algorithms for keyword search [21], and proposed a method to train case bases using the measured satisfaction regarding the proposed solution. Lee et al. proposed a recommendation system that uses an ontology and CBR [15]. Services were represented using an ontology, and CBR was applied to recommend services, because symptoms, diseases, related departments, and physician-related data types are intertwined in complex ways because of the characteristics of healthcare systems and data.

Research using CBR requires different components to build a system depending on the domain, and an appropriate similarity metric is required depending on the characteristics of the elements. Alternatively, various artificial intelligence techniques can be combined. Furthermore, if knowledge or data are sufficiently described, solutions can be derived without special reasoning if a given problem is similar to or equal to the previous case. This also ensures the problem is solved quickly. Revision methods for optimizing reused solutions are among the main challenges. In addition, it is necessary to propose a case-retention method that enables efficient maintenance and management.

3. Proposed Method

In this study, we propose a security requirement recommendation method using the CBR method by combining the concept of cases with PDO based on extensions of the methods in [11,12]. We describe the construction of the CB-PDO, in which case elements are added to the ontological structure and validated through existing case studies. We then employ the CB-PDO to analyze the attack elements in input scenarios and describe procedures and methods according to the CBR process. Furthermore, we propose a similarity metric that is suitable for our domain. It consists of a basic similarity algorithm (baseline) and an APT property called a weight. This section then describes the process of recommending security requirements to provide detailed information to users and help them make decisions according to the subsequent processes. We also define a study question to validate the method and evaluate whether the answer is explainable and justifiable.

Figure 2 provides an overview of the security requirement recommendation method proposed in this study. This method consists of four steps of CBR and one step of preprocessing, and the process performed in each step is shown in the figure. The results generated at each step are used as the input for the next step and can be reviewed and analyzed by security experts to specify security requirements. In addition, our proposed CB-PDO was utilized to provide the knowledge and information necessary to derive the required results. The scenario preprocessing step before the conventional CBR step is converted into an attack template to make it easier to analyze the given attack scenario and provide expert-analyzed attack information. In the case retrieval step, the previous cases that are most similar to the attack components are determined using the similarity metrics. The previous cases included information about attacks performed by threat groups in the past and the security requirements used to prevent them. Therefore, in the case reuse step, we perform an adaptation process to reuse requirements specified in the past to solve the current problem. This step is not necessary if the solution can be used as is; however, if there is a difference in the cases, the solution needs to be optimized so that it is suitable for the current problem. Accordingly, information for optimization is provided to the security experts in the case revision stage. Finally, if an expert determines that the problem was correctly solved, it is then stored in the CB-PDO as a case. The processes and components of each step are described in detail in this subsection.

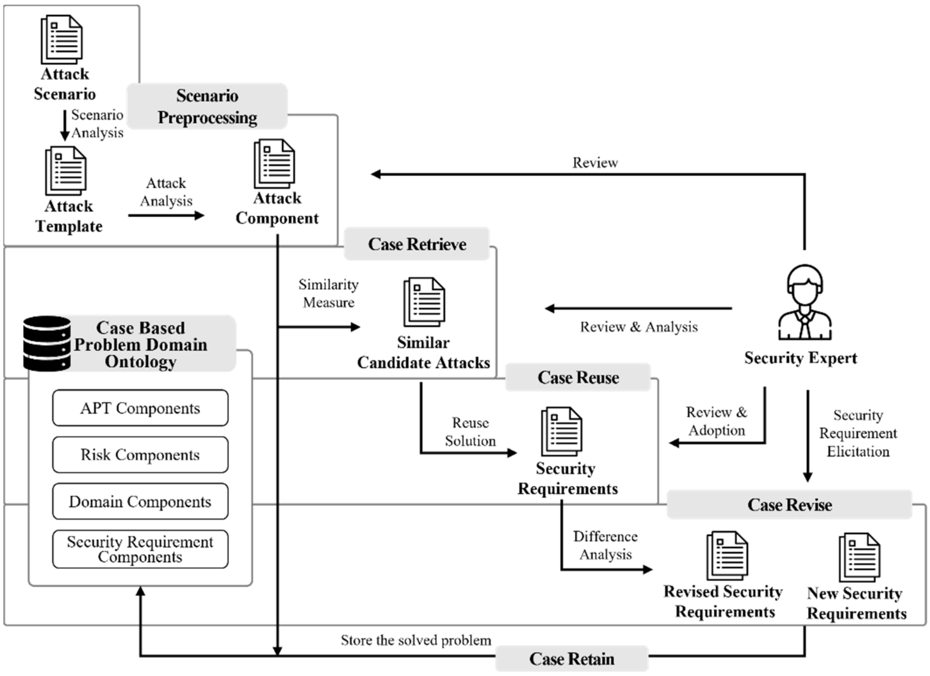
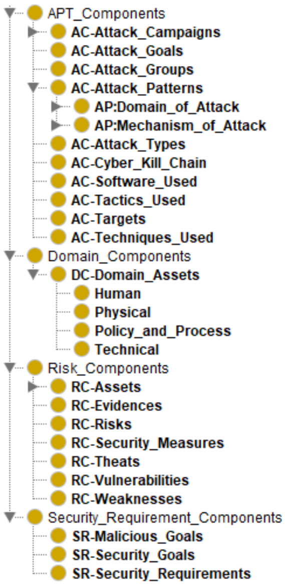


Figure 2. Overview of our proposed method.

3.1. CB-PDO

The ontology proposed in this paper was constructed by partially redefining the structure of an existing PDO [10–12]. Unlike previous ontologies, it was constructed based on four components [22]. The components of the proposed ontology consider not only the security requirement recommendations for APT attacks, but also the scalability for future risk assessments. The CB-PDO consists of APT components, security requirements components, risk components, and domain components. Protégé was used to implement the ontology, and the structure is shown in Figure 3 [23].



- APT_Components
  - AC-Attack_Campaigns
  - AC-Attack_Goals
  - AC-Attack_Groups
  - AC-Attack_Patterns
    - AP:Domain_of_Attack
    - AP:Mechanism_of_Attack
  - AC-Attack_Types
  - AC-Cyber_Kill_Chain
  - AC-Software_Used
  - AC-Tactics_Used
  - AC-Targets
  - AC-Techniques_Used
- Domain_Components
  - DC-Domain_Assets
    - Human
    - Physical
    - Policy_and_Process
    - Technical
- Risk_Components
  - RC-Assets
  - RC-Evidences
  - RC-Risks
  - RC-Security_Measures
  - RC-Theats
  - RC-Vulnerabilities
  - RC-Weaknesses
- Security_Requirement_Components
  - SR-Malicious_Goals
  - SR-Security_Goals
  - SR-Security_Requirements

Figure 3. Components of CB-PDO.

3.1.1. APT Component

The APT component is the part of the knowledge base that is used to analyze previous attack cases and includes all elements related to the APT attacks. The items constituting the APT component are expressed as classes, and the elements corresponding to each class are composed of instances. The classes constituting the APT component and examples of the corresponding instances are listed in Table 1. These classes include the software and techniques used by the attack group to achieve attack goals as well as attack targets. The cyber-kill chain is used to represent a series of techniques, and a tactic is used to distinguish the types of techniques. To ensure the reliability of the data used in this study, classes and instances were implemented based on MITRE ATT&CK [9]. Attack patterns were constructed based on the Common Attack Pattern Enumeration and Classification (CAPEC) [24] and can be classified as attack domain patterns or attack mechanism patterns according to the characteristics of the techniques used in the patterns.

3.1.2. Risk Component

The risk component is a part of the knowledge base that includes the elements required to assess the risk of attack and the risk of exposure to assets, and its classes are listed in Table 2. Vulnerabilities represent flaws that an attack group can directly use to access a system or network. A weakness is a broader concept that includes errors that can lead to software vulnerabilities. Vulnerabilities and weaknesses can also expose an asset to risk, which threat groups can exploit. Therefore, the elements defined in the CWE (Common Weakness Enumeration) [25] and CVE (Common Vulnerabilities and Exposures) [26] are included in the proposed risk component. To perform a risk assessment, evidence is constructed to express the degree to which a set of assets satisfies the security requirements. Threats are a class that are used to infer security requirements by identifying security goals and malicious goals. This class consists of attack elements included in the attack component and vulnerabilities or weaknesses.

Table 1. Description of the APT component.

Class	Description	Instances
Attack group	Name of the threat group	APT 18, Carbanak, and Gorgon Group
Attack type	Type of attack	DDoS and ransomware
Attack campaign	A case that represents an example of a group's behavior	-
Attack goal	The goal that the group wants to achieve	Financial gain, political gain, and production damage
Target	The target threatened by the threat group	Banks and factories
Software	Software used in the attack	S0001: Trojan Mebromi, S0002: Mimikatz
Technique	Technique used in the attack	T1001: data obfuscation
Tactic	Tactic used by a technique of the threat group	TA0001: initial access and TA0002: execution
Cyber-kill chain	Steps of a technique used by the threat group	CKC01: Reconnaissance and CKC02: Delivery
Attack pattern	Pattern of the attack	CAPEC-127 and CAPEC-132

Table 2. Description of the risk component.

Class	Description	Instances
Asset	Asset affected by an attack	AS1, AS2, and AS3
Evidence	The degree to which an asset meets security requirements	-
Risks	Risk of an attacks (the evaluation result of evidence)	-
Threats	Security threat given attack information	-
Vulnerabilities	A mistake in software that a threat group can directly use to access a system or network	CVE-2019-9670 and CVE-2020-10189
Weakness	An error that can lead to software vulnerabilities	CWE-73 and CWE-836

3.1.3. Security Requirements and Domain Components

The security requirement component consists of security goals, malicious goals, and security requirements. Security goals include three elements: confidence, integrity, and availability, and these goals include goals to counter the threats presented in the risk component. In addition, the three elements of security, exposure, modification, and destruction are included. The goals of the threat groups are included in the malicious goals. Security requirements include existing security requirements specified by the Security Technical Implementation Guide [27] and newly specified security requirements depending on the elements derived from the risk component.

The domain component is an extended component for future risk assessment, and the assets identified in the risk component are mapped to four layers. Domain assets are considered in terms of business processes and human, technical, and physical aspects. Table 3 describes these two components.

Table 3. Description of the security requirement and domain components.

Component	Class	Description	Instances
Security requirement component	Security goal	Three elements of security or asset security goals to protect against threats	Confidentiality, integrity, availability, SG01, SG02, and SG03
	Malicious goal	Opposing security goal to expose an asset to risk	Exposure, modification, destroy, MG01, MG02, and MG03
	Security requirement	Requirement elicited from the security requirements list or the risk component	SR0001 and SR0002
Domain component	Domain asset	Secured domain assets classified by four-layered perspective	-

3.1.4. Case-Base integrated Knowledge Base: CB-PDO

CB-PDO was developed by integrating the components described above, and security requirements are recommended based on analysis of each case. The relationships among the elements used for reuse and review are shown in the conceptual model in Figure 4. An integrated knowledge base provides users with information about each component, enabling them to further understand the interrelationships among each component and identify the impact of one element on the other elements.

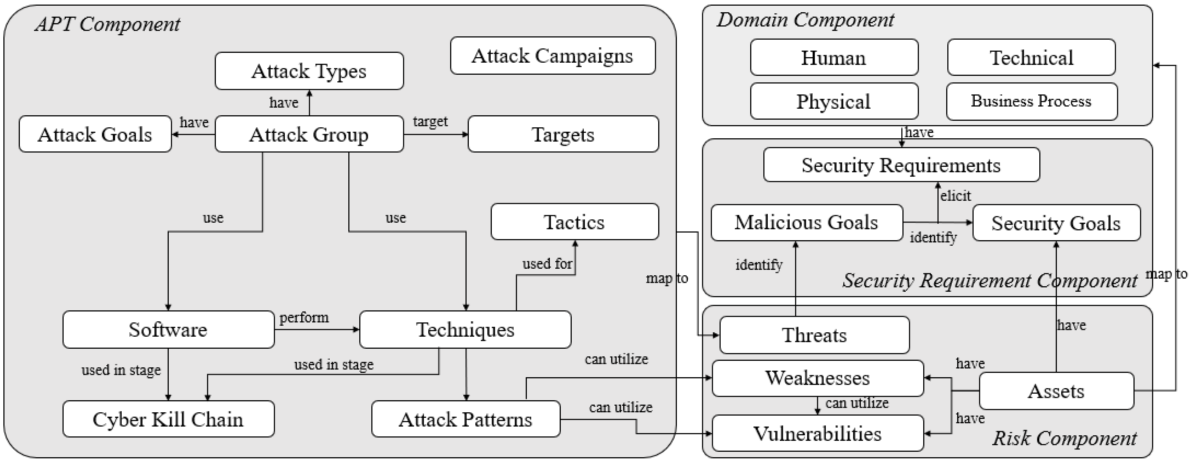


Figure 4. Conceptual model of CB-PDO.

Attack campaigns were added to the ontology to represent the structure of each case. The elements included in attack campaigns are a combination of information about the APT and security requirement components, as shown in Table 4. In each case, it is assumed that past events are true. A case consists of the attack information (group, target, goal, software, techniques, and cyber-kill chain) and security requirements. Hence, if there is a past case that is similar to the new problem, it can be used to solve the new problem. This saves time and money when deriving new solutions.

Table 4. Elements of an attack case.

Campaign Element	Description
Attack goal	Goal of the threat group
Attack group	Name of the threat group
Target	Target (e.g., organization, enterprise, or government) of the attack
Software	Software or tool used by the threat group
Techniques	Techniques that the threat group perform during the campaign
Cyber-kill chain	Phases of the attack using the techniques performed by the threat group
Security requirements	Security requirements required as the result of a previous APT

3.2. CBR Process

This section describes the implementation of each step of the CBR process. The outputs obtained for each step are summarized in Table 5. Detailed implementation methods are described, and the evaluation of the results is explained in the case study in the next section.

Table 5. Artefacts of each process.

Category	Input	Process	Artefact
Scenario preprocessing	Scenario Attack template	Scenario analysis Attack analysis	Attack template Attack component
Case retrieval	Attack component	Similarity metric	Similar attack candidates
Case reuse	Similar attack candidates	Reuse solution	Security requirements
Case revision	Attack component Similar attack candidates Security requirements Security expert information	Difference analysis	Revised security requirements
	Security expert information	Security requirements elicitation	New security requirements
Case retention	Attack component Revised or new security requirements	Storage of the solved problem	Improved CB-PDO

3.2.1. Scenario Preprocessing

Scenarios that are used as inputs may contain attack techniques and attack information provided by threat groups as past attack reports [28,29] as well as attack simulations to derive the security requirements required for system design. However, because the reports and documents reported by each security company have different levels of abstraction, it is difficult to extract elements at an appropriate level. Therefore, to easily extract the necessary elements from a scenario, an attack template is provided to the user. The user can identify and add appropriate attack elements to the scenario through the categories provided in the attack template. The template consists of guidelines for inputting the attacker's basic attack information and detailed attack technique elements. Basic information consists of the goals, targets of attack, and software, and attack techniques can be identified from the CKC and MITRE Tactic perspectives [8,9]. The results of applying the template are presented in Section 4.

A user of the proposed method is a security expert who needs to be provided with specific, correct, and understandable information about the input elements. Therefore, before retrieving a case, we provide understandable and analytical results to the users in the attack component. The attack component consists of the results of the following five questions, which are based on the relationship between the elements inputted to the template and those defined by the CB-PDO.

Q1 Which tactics does the technique employ?
Q2 What platform does the technique work on?
Q3 What kind of attack pattern does the technique use?
Q4 What weaknesses does the attack pattern use?
Q5 What vulnerabilities does the weakness use?

To represent the results from these questions, we use SPARQL [30]. The results of these questions help the user to make decisions and increase their understanding.

3.2.2. Case Retrieval

The purpose of this study was to develop a system that finds the previous APT cases that are the most similar to the current APT case. In this study, we used Jaccard similarity as a basic baseline metric to measure the similarity between the current problem and previous cases. Jaccard similarity is a method for measuring the similarity of two objects. The similarity between the two sets is defined as the ratio of the intersection to the union, as follows.

$$\mathrm{J}(\mathrm{A},\ \mathrm{B}) = \frac{|\mathrm{A} \cap \mathrm{B}|}{|\mathrm{A} \cup \mathrm{B}|} = \frac{|\mathrm{A} \cap \mathrm{B}|}{|\mathrm{A}| + |\mathrm{B}| - |\mathrm{A} \cap \mathrm{B}|} \tag{1}$$

Using Jaccard similarity, the overall similarity of the software and techniques in the previous case and current problem is calculated and represented as a value between 0 and 1. Equation (2) is used to calculate the case-to-case similarity measure, which is represented as the sum of the matching software and skill sets, where C represents the current problem and C'_n represents the nth previous case.

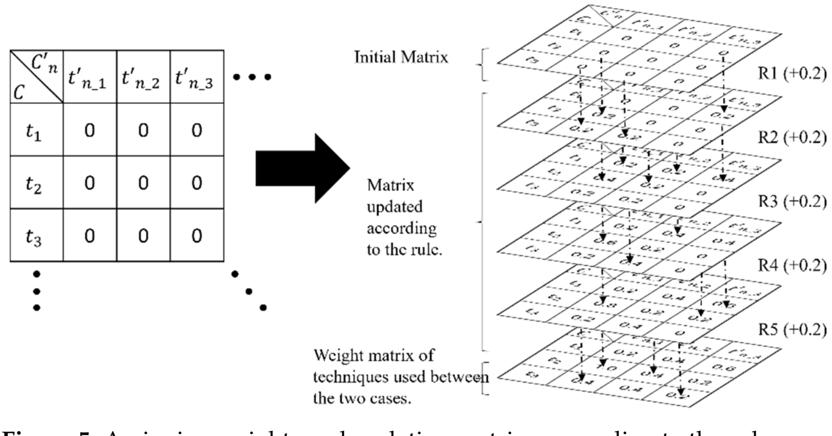
$$\mathrm{J}(\mathrm{C}, \mathrm{C}'_\mathrm{n}) = \mathrm{J}(\mathrm{C}_\mathrm{SW}, \mathrm{C}'_{\mathrm{n}_\mathrm{sw}}) + \mathrm{J}(\mathrm{C}_\mathrm{Tech}, \mathrm{C}'_{\mathrm{n}_\mathrm{Tech}}) \tag{2}$$

However, ontology includes numerous characteristics that can represent the characteristics of an APT in addition to the software and technologies used by the threat group. Therefore, we propose a method to measure the similarity between techniques by considering the similarity of the characteristics in addition to Jaccard similarity, which simply considers whether the same elements are used. To measure similar elements, we compared the class of elements in the ontology or the relationship between instances belonging to that class. We defined weights for computing the APT features and reflect them in the similarity measurements. The questions that can be selected to measure the similarity between technique instances stored in an ontology are as follows:

R1 Do the two elements have a parent–child/sibling relationship?
R2 Can two elements be performed on the same platform?
R3 Do the two elements use the same tactics?
R4 Are the attack patterns utilized by the two elements the same?
R5 Were the two elements performed in the same attack stage (in the cyber-kill chain)?

The answers to each of these five questions have an equal initial weight, and hence, 0.2 is assigned to each satisfied condition. If all are satisfied, the total is 1. To assign weights between the current problem C and past case C'_n, the weights for features satisfying the conditions are calculated by formulating the problem for the two cases as a matrix, as shown in Figure 5. The size of the matrix is the number of techniques performed in C × the number of techniques performed in C'_n. Equation (3) shows that, according to the rule, 0.2 or 0 is assigned to conditions that are satisfied.

$$\mathrm{sim}(t, t')\begin{cases} 0.2 \text{ (satisfied with the rule)} \\ 0 \text{ (not satisfied with the rule)} \end{cases} \tag{3}$$



C'_n / C	t'_{n_1}	t'_{n_2}	t'_{n_3}	• • •
t_1	0	0	0	
t_2	0	0	0	
t_3	0	0	0	
⋮				⋱

Initial Matrix
Matrix updated according to the rule.
Weight matrix of techniques used between the two cases.
R1 (+0.2)
R2 (+0.2)
R3 (+0.2)
R4 (+0.2)
R5 (+0.2)

Figure 5. Assigning weights and updating matrices according to the rules.

Equation (4) represents the overall average of the weight matrix. Finally, the similarity between the new problem and the existing cases is defined by Equation (5) as the sum of the Jaccard similarity and weight.

$$\mathrm{weight}(\mathrm{C}, \mathrm{C}') = \frac{1}{\mathrm{m}}\frac{1}{\mathrm{n}}\sum_{\mathrm{m}=1}^{\mathrm{m}}\sum_{\mathrm{n}=1}^{\mathrm{n}} \mathrm{sim}(t_\mathrm{m}, t'_\mathrm{n}) \tag{4}$$

$$\mathrm{Sim}(\mathrm{C},\ \mathrm{C}') = \mathrm{J}(\mathrm{C},\ \mathrm{C}'_\mathrm{n}) + \mathrm{weight}(\mathrm{C}, \mathrm{C}') \tag{5}$$

3.2.3. Case Reuse

Case reuse provides the user with a solution that has been used to resolve the most similar case identified in case retrieval. The security requirements found in previous cases have already been used as solutions for previous APTs, indicating the degree of their reliability. Therefore, this step provides the security requirements to address the attack methods and tactical characteristics of the identified similar cases. To recommend the security requirements, we answer the following questions:

Q1 What kind of techniques did the case use?
Q2 What is the attack method in this case?
Q3 What are the security requirements for this case?

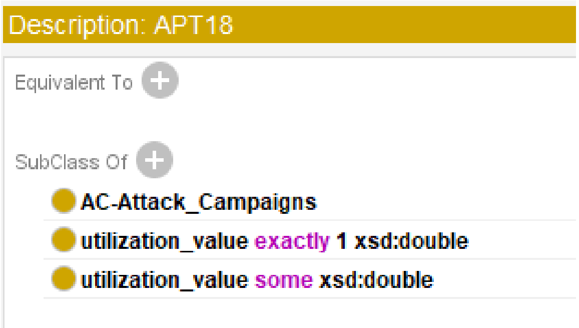
By providing users with the results of these questions, we enable them to identify the characteristics and examples of attacks used by threat groups in similar cases and review the security requirements. These requirement recommendations are provided in descending order according to the number of times they were required in previous attacks. Therefore, the user can select recyclable security requirements from among the proposed security requirements. This reduces the time and labor costs required to derive new security requirements.

3.2.4. Case Revision

The differences between the current problem and previous similar cases mean that simply reusing a previous solution does not identify all the necessary security requirements. Therefore, case revision is required to correct the security requirements that fit the current problem. The goal of our study is to provide users with information and recommend requirements to help them make decisions about eliciting security requirements. Therefore, this step provides information that can be referenced and will be helpful to users when modifying security requirements. There are two ways to correct a case plan. The first method recommends additional security requirements that might be needed. Using the answer to Q1 for case reuse, we analyze the differences in the attack characteristics of current problems and similar cases and recommend security requirements that can reduce these differences. For example, it is possible to analyze tactical differences in attack techniques that appear in cases that are similar to the scenario. Visualizing this difference and providing it to users can elicit the requirements for attack tactics that have not yet been addressed by the security requirements. In addition, the relationship between attack elements, software, and techniques is a 1:N relationship; thus, there are additional attack techniques that can be performed in the software and be used to identify additional recommended security requirements. In the second method, users manually create or revise new security requirements by inferring the relationships among the elements in the CB-PDO. This method enables the user to derive security requirements manually by utilizing a knowledge framework. To elicit security requirements, malicious goals and security goals can be extracted from threats using the risk assessment process proposed in [22]. Thus, the extracted goals can be derived through a goal-based approach, and the process can be found in a case study in [22]. This process is performed, and then the user verifies that the solution obtained to solve the current problem is appropriate. Finally, it is stored in CB-PDO as a subclass of the attack campaigns.

3.2.5. Case Retention

If it is verified that the problem has been completely solved, it is stored in the ontology as a new case. To convert the problem and solution to a case, the information is rearranged according to the components listed in Table 4. This includes examples of the attack case and has elements that can represent and explain the case well. As such problems are continuously being solved, the number of cases containing new attack information increases, increasing the problem-solving ability. In addition, a utilization value (UV) is assigned to each case depending on whether the case is retrieved and utilized for solving subsequent problems. This value evaluates the case in which the user helps solve the problem, and for each case, the UV is expressed as a property in the ontology. Figure 6 shows that the UV of the APT18 group was stored using the ontology editor Protégé. Continuous problem solving in the future will lead to high- and low-use cases, and users will be able to maintain the ontology, for example, by eliminating useless cases.



Description: APT18

Equivalent To

SubClass Of

- AC-Attack_Campaigns
- utilization_value exactly 1 xsd:double
- utilization_value some xsd:double

Figure 6. Utilization value of the retained case.

4. Evaluation

In this section, the results obtained by implementing the proposed method are discussed. We employed two verification methods. First, a case study was conducted using the study questions and a case study proposal. Second, our proposed method and each process were evaluated by security experts to determine if they were substantially helpful. We provided a half-day tutoring session about the proposed approach to the security experts with a prototype of the tool [31] so that they could understand and be familiar with the approach. To this end, a questionnaire was prepared and administered such that the contents could be evaluated. Finally, using the responses of the questionnaire, an evaluation of the process and results according to the quality attributes is presented.

4.1. Case Study

We defined the study question using “how” and “why” questions and defined the case study proposals that must be evaluated to satisfy the study question [32], which is as follows.

Study Question: How can and why does the proposed method recommend and help users meet security requirements?

To answer the study question, we subdivided it into general proposals (GPs) and specific proposals (SPs).

GP1: Through the proposed ontology and process, the results can be used to analyze and understand APT characteristics, and security requirements can be recommended by retrieving cases that are similar to the scenario.

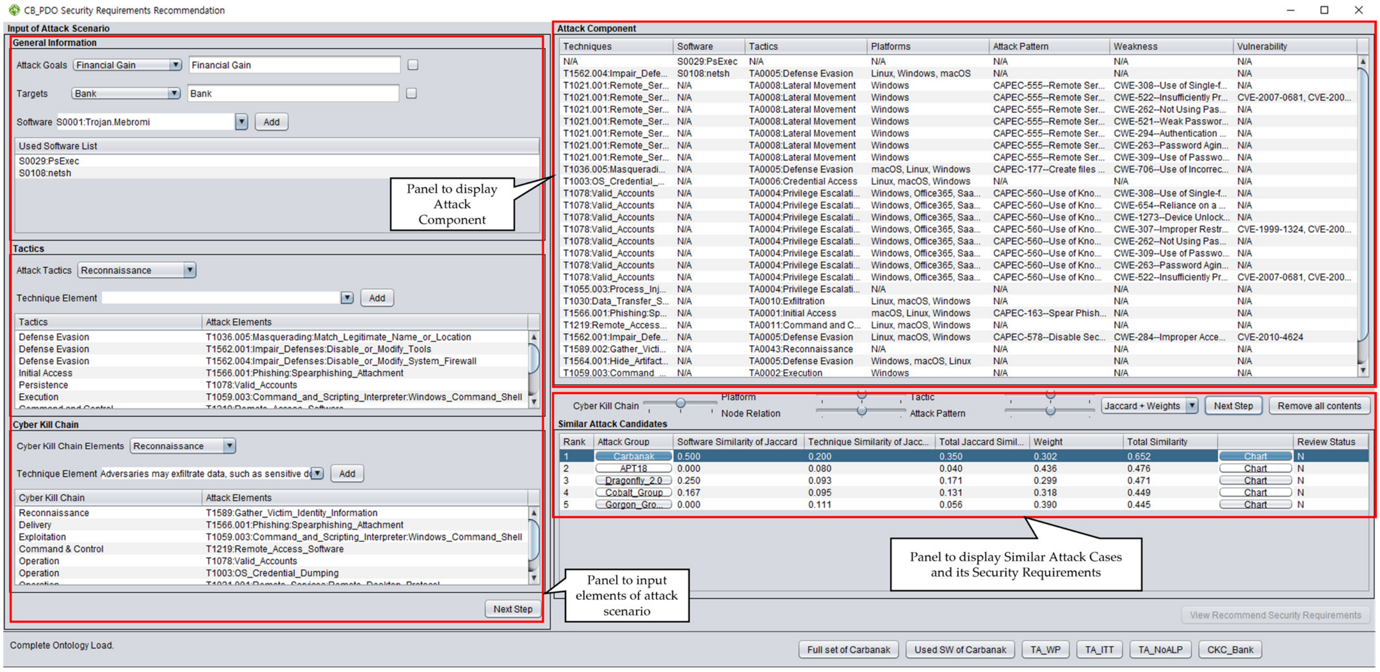
GP2: To recommend useful security requirements, we can retrieve cases similar to the scenario and reuse the security requirements contained in these similar cases.

SP1.1: To recommend useful security requirements, we can retrieve cases similar to the scenario and reuse the security requirements contained in these similar cases.
SP1.2: The user may receive the result of analyzing the APT characteristics through the relationship of elements configured in the CB-PDO.
SP2.1: Similar cases can be retrieved and explained using similarity metrics that reflect the characteristics of the APT.
SP2.2: Depending on the given environment and constraints, users can retrieve cases in which a particular characteristic stands out by strongly weighting one particular characteristic.
SP2.3: Users can review and manage the recommended security requirements to solve current problems using similar cases.

To evaluate whether the defined study goal was satisfied, several analyses were performed, as summarized in Table 6. We defined the evidence needed to meet each study proposal and study criteria to prove they are met, as described in the process of the proposed method. For the analysis, we designed a prototype using Java, and a screenshot of the program is shown in Figure 7. Other processes were described through an analysis of the study proposition.

Table 6. Study propositions and analyses.

General Proposition	Specific Proposition	Evidence	Study Criteria (Process)
GP1	SP1.1	It is possible to identify characteristics that meet the conditions in the scenario using the provided template.	Scenario analysis
	SP1.2	It is possible to reason about each related attack feature from the relationships among the elements in the CB-PDO.	Attack analysis
GP2	SP2.1	It is possible to reason about similar attacks using the similarity of the features used to represent the APT attacks.	Similarity measure
	SP2.2	If there are environments or restrictions that the user considers, the priority of features may be different, and a result that considers these priorities is presented.	Similarity measure
	SP2.3	Users are provided with functions for reusing and adopting requirements.	Solution reuse



CB_PDO Security Requirements Recommendation

Input of Attack Scenario

General Information

Attack Goals: Financial Gain ▾ | Financial Gain ☐

Targets: Bank ▾ | Bank ☐

Software: S0001:Trojan.Mebromi ▾ Add

Used Software List
S0029:PsExec
S0108:netsh

Tactics

Attack Tactics: Reconnaissance ▾

Technique Element: ▾ Add

Tactics	Attack Elements
Defense Evasion	T1036.005:Masquerading:Match_Legitimate_Name_or_Location
Defense Evasion	T1562.001:Impair_Defenses:Disable_or_Modify_Tools
Defense Evasion	T1562.004:Impair_Defenses:Disable_or_Modify_System_Firewall
Initial Access	T1566.001:Phishing:Spearphishing_Attachment
Persistence	T1078:Valid_Accounts
Execution	T1059.003:Command_and_Scripting_Interpreter:Windows_Command_Shell
Command and Control	T1219:Remote_Access_Software

Cyber Kill Chain

Cyber Kill Chain Elements: Reconnaissance ▾

Technique Element: Adversaries may exfiltrate data, such as sensitive d ▾ Add

Cyber Kill Chain	Attack Elements
Reconnaissance	T1589:Gather_Victim_Identity_Information
Delivery	T1566.001:Phishing:Spearphishing_Attachment
Exploitation	T1059.003:Command_and_Scripting_Interpreter:Windows_Command_Shell
Command & Control	T1219:Remote_Access_Software
Operation	T1078:Valid_Accounts
Operation	T1003:OS_Credential_Dumping
Operation	T1021.001:Remote_Services:Remote_Desktop_Protocol

Next Step

Complete Ontology Load.

Attack Component

Techniques	Software	Tactics	Platforms	Attack Pattern	Weakness	Vulnerability
N/A	S0029:PsExec	N/A	N/A	N/A	N/A	N/A
T1562.004:Impair_Defe...	S0108:netsh	TA0005:Defense Evasion	Linux, Windows, macOS	N/A	N/A	N/A
T1021.001:Remote_Ser...	N/A	TA0008:Lateral Movement	Windows	CAPEC-555--Remote Ser...	CWE-308--Use of Single-f...	N/A
T1021.001:Remote_Ser...	N/A	TA0008:Lateral Movement	Windows	CAPEC-555--Remote Ser...	CWE-522--Insufficiently Pr...	CVE-2007-0681, CVE-200...
T1021.001:Remote_Ser...	N/A	TA0008:Lateral Movement	Windows	CAPEC-555--Remote Ser...	CWE-262--Not Using Pas...	N/A
T1021.001:Remote_Ser...	N/A	TA0008:Lateral Movement	Windows	CAPEC-555--Remote Ser...	CWE-521--Weak Passwor...	N/A
T1021.001:Remote_Ser...	N/A	TA0008:Lateral Movement	Windows	CAPEC-555--Remote Ser...	CWE-294--Authentication ...	N/A
T1021.001:Remote_Ser...	N/A	TA0008:Lateral Movement	Windows	CAPEC-555--Remote Ser...	CWE-263--Password Agin...	N/A
T1021.001:Remote_Ser...	N/A	TA0008:Lateral Movement	Windows	CAPEC-555--Remote Ser...	CWE-309--Use of Passwo...	N/A
T1036.005:Masqueradi...	N/A	TA0005:Defense Evasion	macOS, Linux, Windows	CAPEC-177--Create files ...	CWE-706--Use of Incorrec...	N/A
T1003:OS_Credential_...	N/A	TA0006:Credential Access	Linux, macOS, Windows	N/A	N/A	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, Saa...	CAPEC-560--Use of Kno...	CWE-308--Use of Single-f...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, Saa...	CAPEC-560--Use of Kno...	CWE-654--Reliance on a ...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, Saa...	CAPEC-560--Use of Kno...	CWE-1273--Device Unlock...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, Saa...	CAPEC-560--Use of Kno...	CWE-307--Improper Restr...	CVE-1999-1324, CVE-200...
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, Saa...	CAPEC-560--Use of Kno...	CWE-262--Not Using Pas...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, Saa...	CAPEC-560--Use of Kno...	CWE-309--Use of Passwo...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, Saa...	CAPEC-560--Use of Kno...	CWE-263--Password Agin...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, Saa...	CAPEC-560--Use of Kno...	CWE-522--Insufficiently Pr...	CVE-2007-0681, CVE-200...
T1055.003:Process_Inj...	N/A	TA0004:Privilege Escalati...	N/A	N/A	N/A	N/A
T1030:Data_Transfer_S...	N/A	TA0010:Exfiltration	Linux, macOS, Windows	N/A	N/A	N/A
T1566.001:Phishing:Sp...	N/A	TA0001:Initial Access	macOS, Linux, Windows	CAPEC-163--Spear Phish...	N/A	N/A
T1219:Remote_Access...	N/A	TA0011:Command and C...	Linux, macOS, Windows	N/A	N/A	N/A
T1562.001:Impair_Defe...	N/A	TA0005:Defense Evasion	Linux, macOS, Windows	CAPEC-578--Disable Sec...	CWE-284--Improper Acce...	CVE-2010-4624
T1589.002:Gather_Victi...	N/A	TA0043:Reconnaissance	N/A	N/A	N/A	N/A
T1564.001:Hide_Artifact...	N/A	TA0005:Defense Evasion	Windows, macOS, Linux	N/A	N/A	N/A
T1059.003:Command ...	N/A	TA0002:Execution	Windows	N/A	N/A	N/A

Jaccard + Weights ▾ Next Step Remove all contents

Similar Attack Candidates

Rank	Attack Group	Software Similarity of Jaccard	Technique Similarity of Jacc...	Total Jaccard Simil...	Weight	Total Similarity		Review Status
1	Carbanak	0.500	0.200	0.350	0.302	0.652	Chart	N
2	APT18	0.000	0.080	0.040	0.436	0.476	Chart	N
3	Dragonfly_2.0	0.250	0.093	0.171	0.299	0.471	Chart	N
4	Cobalt_Group	0.167	0.095	0.131	0.318	0.449	Chart	N
5	Gorgon_Gro...	0.000	0.111	0.056	0.390	0.445	Chart	N

View Recommend Security Requirements

Full set of Carbanak | Used SW of Carbanak | TA_WP | TA_ITT | TA_NoALP | CKC_Bank

Panel to display Attack Component

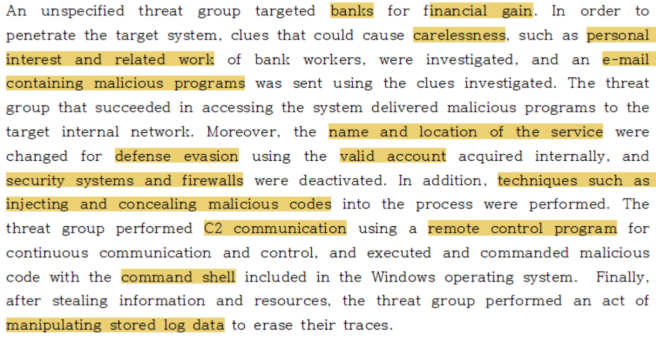
Panel to display Similar Attack Cases and its Security Requirements

Panel to input elements of attack scenario

Figure 7. Initial screen of the prototype.

4.1.1. SP1.1

A virtual scenario was created to evaluate the study proposition. The scenario was written by combining the attack techniques used in several previous cases [29,33]. In this implementation, only the elements of the attack technique are reflected to compare the explainable results according to the application of weights to the attack technique information. The example scenario is shown in Figure 8, where the user has highlighted the elements to be extracted from the scenario. The user converts the identified elements according to the template, as presented in Table 7. A screenshot of the results entered into the prototype is shown in Figure 9, demonstrating that the user can determine the elements to be extracted and organize them using the template.



An unspecified threat group targeted banks for financial gain. In order to penetrate the target system, clues that could cause carelessness, such as personal interest and related work of bank workers, were investigated, and an e-mail containing malicious programs was sent using the clues investigated. The threat group that succeeded in accessing the system delivered malicious programs to the target internal network. Moreover, the name and location of the service were changed for defense evasion using the valid account acquired internally, and security systems and firewalls were deactivated. In addition, techniques such as injecting and concealing malicious codes into the process were performed. The threat group performed C2 communication using a remote control program for continuous communication and control, and executed and commanded malicious code with the command shell included in the Windows operating system. Finally, after stealing information and resources, the threat group performed an act of manipulating stored log data to erase their traces.

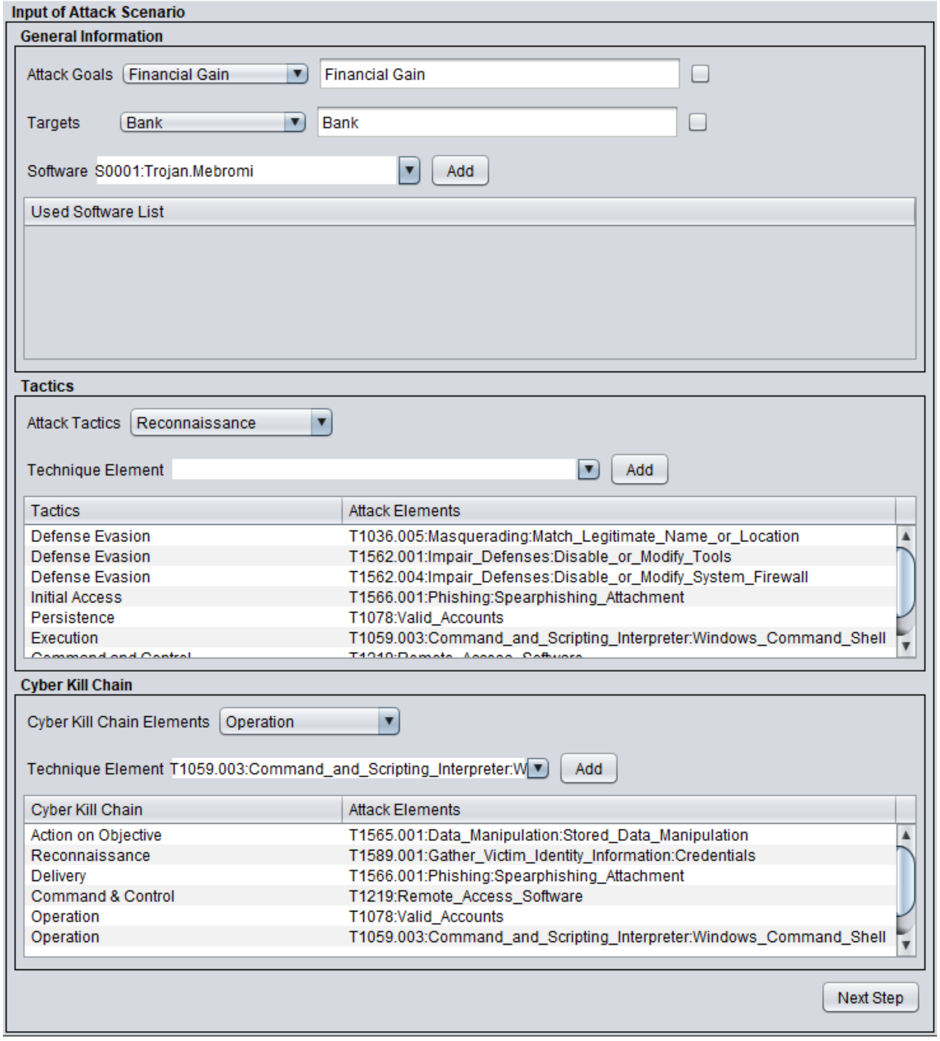
Figure 8. Example scenario for evaluation.

Table 7. Result of filling in the attack template.

Attack Elements		Instances
Objective		Financial Goal
Target		Bank
Software		-
Cyber-kill Chain	Reconnaissance	T1589:Gather_Victim_Identity_Information
	Delivery	T1566.001:Phishing:Spearphishing_Attachment
	Command and Control	T1219:Remote_Access_Software
	Operation	T1078:Valid_Accounts T1059.003:Command_and_Scripting_Interpreter:Windows_Command_Shell
	Action on the Objective	T1030:Data_Transfer_Size_Limits
Tactic	Initial Access	T1566.001:Phishing:Spearphishing_Attachment
	Command and Control	T1219:Remote_Access_Software
	Execution	T1059.003:Command_and_Scripting_Interpreter:Windows_Command_Shell
	Persistence	T1078:Valid_Accounts
	Defense Evasion	T1036.005:Masquerading:Match_Legitimate_Name_or_Location T1562.001:Impair_Defenses:Disable_or_Modify_Tools T1562.004:Impair_Defenses:Disable_or_Modify_System_Firewall T1055.003:Process_Injection:Thread_Execution_Hijacking T1564.001:Hide_Artifacts:Hidden_Files_and_Directories

4.1.2. SP1.2

Figure 10 shows the attack component results, which provide the user with detailed information regarding the attack and the elements entered to provide the answer to this proposition. As a result of the previously mentioned five questions, the tactical characteristics, operating platforms, security weaknesses, and vulnerabilities that can be exploited as attack patterns are derived. Elements that match the query appear and elements that have not been fully analyzed or have not yet been updated are marked as "N/A." This process will continue to improve as the amount of security-related knowledge and data increases and is analyzed in the future. The attack component makes it easier for users to access the entered attack information and perform case retrieval using the attack element information.



Input of Attack Scenario

General Information

Attack Goals: Financial Gain ▼ Financial Gain □

Targets: Bank ▼ Bank □

Software: S0001:Trojan.Mebromi ▼ Add

Used Software List

Tactics

Attack Tactics: Reconnaissance ▼

Technique Element: ▼ Add

Tactics	Attack Elements
Defense Evasion	T1036.005:Masquerading:Match_Legitimate_Name_or_Location
Defense Evasion	T1562.001:Impair_Defenses:Disable_or_Modify_Tools
Defense Evasion	T1562.004:Impair_Defenses:Disable_or_Modify_System_Firewall
Initial Access	T1566.001:Phishing:Spearphishing_Attachment
Persistence	T1078:Valid_Accounts
Execution	T1059.003:Command_and_Scripting_Interpreter:Windows_Command_Shell
Command and Control	T1219:Remote_Access_Software

Cyber Kill Chain

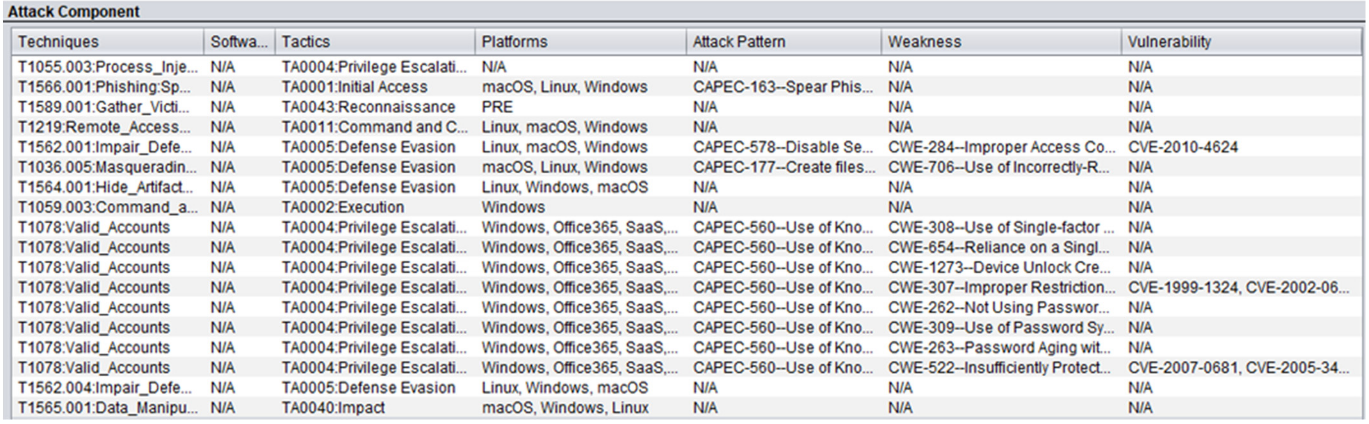
Cyber Kill Chain Elements: Operation ▼

Technique Element: T1059.003:Command_and_Scripting_Interpreter:W ▼ Add

Cyber Kill Chain	Attack Elements
Action on Objective	T1565.001:Data_Manipulation:Stored_Data_Manipulation
Reconnaissance	T1589.001:Gather_Victim_Identity_Information:Credentials
Delivery	T1566.001:Phishing:Spearphishing_Attachment
Command & Control	T1219:Remote_Access_Software
Operation	T1078:Valid_Accounts
Operation	T1059.003:Command_and_Scripting_Interpreter:Windows_Command_Shell

Next Step

Figure 9. Attack scenario input screen.



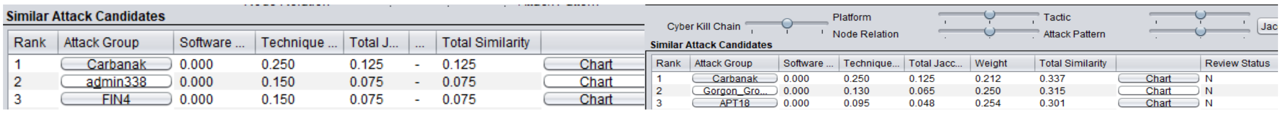
Attack Component

Techniques	Softwa...	Tactics	Platforms	Attack Pattern	Weakness	Vulnerability
T1055.003:Process_Inje...	N/A	TA0004:Privilege Escalati...	N/A	N/A	N/A	N/A
T1566.001:Phishing:Sp...	N/A	TA0001:Initial Access	macOS, Linux, Windows	CAPEC-163--Spear Phis...	N/A	N/A
T1589.001:Gather_Victi...	N/A	TA0043:Reconnaissance	PRE	N/A	N/A	N/A
T1219:Remote_Access...	N/A	TA0011:Command and C...	Linux, macOS, Windows	N/A	N/A	N/A
T1562.001:Impair_Defe...	N/A	TA0005:Defense Evasion	Linux, macOS, Windows	CAPEC-578--Disable Se...	CWE-284--Improper Access Co...	CVE-2010-4624
T1036.005:Masqueradin...	N/A	TA0005:Defense Evasion	macOS, Linux, Windows	CAPEC-177--Create files...	CWE-706--Use of Incorrectly-R...	N/A
T1564.001:Hide_Artifact...	N/A	TA0005:Defense Evasion	Linux, Windows, macOS	N/A	N/A	N/A
T1059.003:Command_a...	N/A	TA0002:Execution	Windows	N/A	N/A	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, SaaS,...	CAPEC-560--Use of Kno...	CWE-308--Use of Single-factor ...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, SaaS,...	CAPEC-560--Use of Kno...	CWE-654--Reliance on a Singl...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, SaaS,...	CAPEC-560--Use of Kno...	CWE-1273--Device Unlock Cre...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, SaaS,...	CAPEC-560--Use of Kno...	CWE-307--Improper Restriction...	CVE-1999-1324, CVE-2002-06...
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, SaaS,...	CAPEC-560--Use of Kno...	CWE-262--Not Using Passwor...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, SaaS,...	CAPEC-560--Use of Kno...	CWE-309--Use of Password Sy...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, SaaS,...	CAPEC-560--Use of Kno...	CWE-263--Password Aging wit...	N/A
T1078:Valid_Accounts	N/A	TA0004:Privilege Escalati...	Windows, Office365, SaaS,...	CAPEC-560--Use of Kno...	CWE-522--Insufficiently Protect...	CVE-2007-0681, CVE-2005-34...
T1562.004:Impair_Defe...	N/A	TA0005:Defense Evasion	Linux, Windows, macOS	N/A	N/A	N/A
T1565.001:Data_Manipu...	N/A	TA0040:Impact	macOS, Windows, Linux	N/A	N/A	N/A

Figure 10. Attack component results.

4.1.3. SP2.1

To answer proposition SP2.1, we compare the results of applying the baseline and applying the baseline plus additional weights. Figure 11 shows the results for the two similarities. The Carbanak, admin338, and FIN4 groups are retrieved when Jaccard similarity is used. The results have relatively low values because this similarity compares only the presence of the same elements in two sets. Figure 12 shows that more common elements increase the overlapping area in the scenarios and cases. In contrast, the results of applying the weights are the Carbanak, Gorgon, and APT18 groups. In particular, the Gorgon Group and APT18 appear in the second and third rankings, respectively, because of the similarity of the APT characteristics. They have lower Jaccard similarity values than Carbanak, but higher weight values. Figure 13 shows the similarity relationship between the APT features of the scenario and the retrieved results to evaluate whether these results have a relationship satisfying the previously defined rules. The shaded elements represent matching elements, and each line shows elements that satisfy the rule and content. The Carbanak and Gorgon group have mechanisms for manipulating system resources that are consistent with the input scenario, and APT18 has a commonality in disabling access control. In addition, the results show that among the attack elements, the techniques can only be performed on the Windows operating system; thus, the elements used by the resulting groups also reflect those performed only on Windows. The results demonstrate retrieval by influencing the similarity measurement by the features defined in this study, as well as the consistency of Jaccard similarity.



Similar Attack Candidates

Rank	Attack Group	Software ...	Technique ...	Total J...	...	Total Similarity	
1	Carbanak	0.000	0.250	0.125	-	0.125	Chart
2	admin338	0.000	0.150	0.075	-	0.075	Chart
3	FIN4	0.000	0.150	0.075	-	0.075	Chart

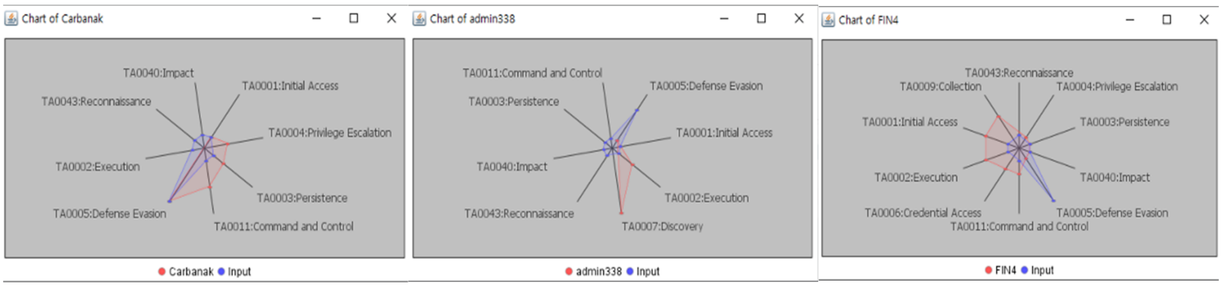
(**a**)

Similar Attack Candidates

Rank	Attack Group	Software ...	Technique...	Total Jacc...	Weight	Total Similarity		Review Status
1	Carbanak	0.000	0.250	0.125	0.212	0.337	Chart	N
2	Gorgon_Gro...	0.000	0.130	0.065	0.250	0.315	Chart	N
3	APT18	0.000	0.095	0.048	0.254	0.301	Chart	N

(**b**)

Figure 11. Result of similarity measurement. (**a**) Baseline (Jaccard similarity) and (**b**) baseline + weights.

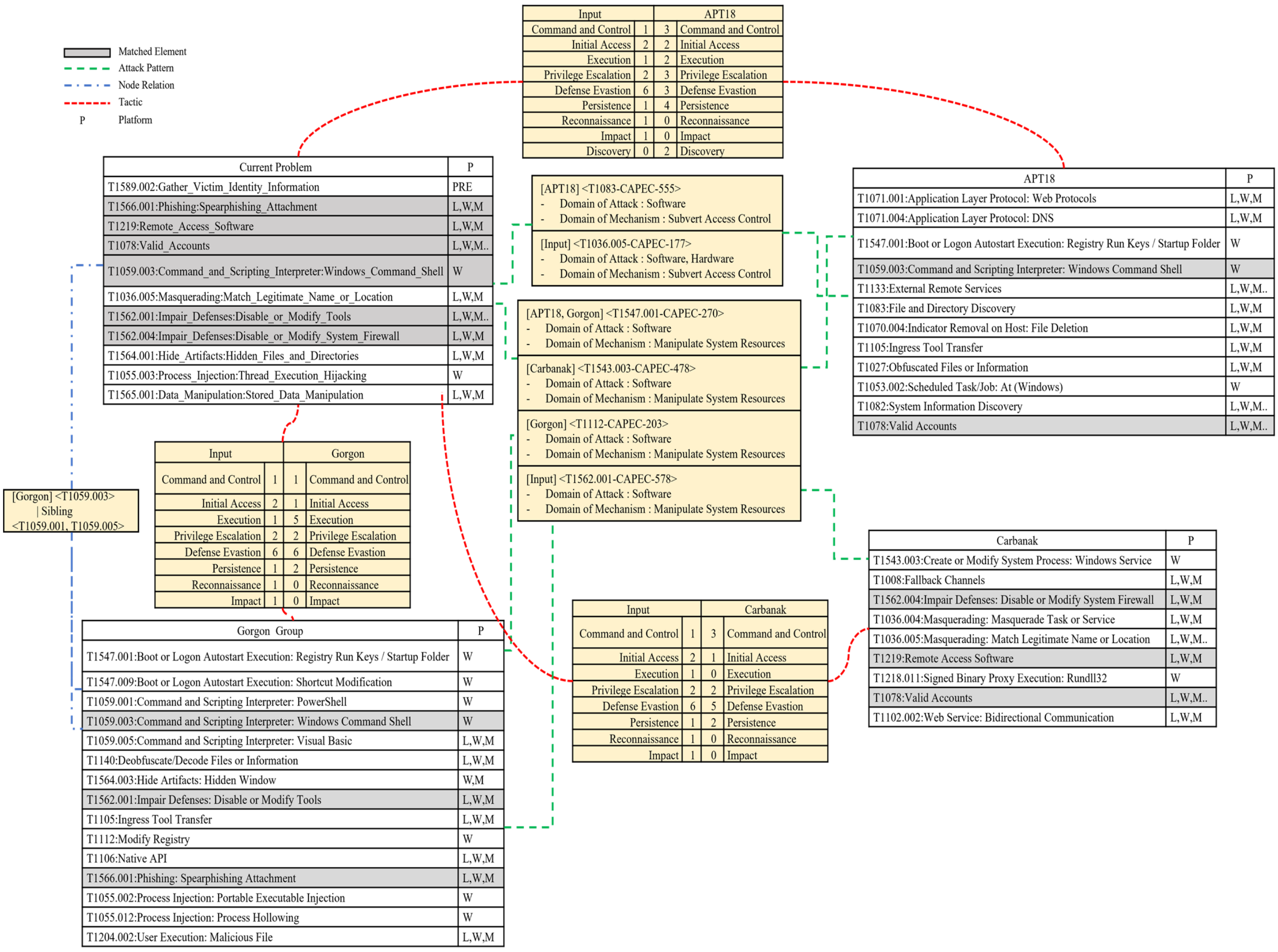


(**a**) (**b**) (**c**)

Figure 12. Radar chart of the difference between the tactics used by the group and the new input tactics. (**a**) Carbanak, (**b**) admin338, and (**c**) FIN4.

4.1.4. SP2.2

In proposition SP2.2, we consider allowing users to assign higher weights to one feature, as they may consider a particular APT feature particularly important. This allows them to retrieve similar cases. To answer the corresponding research proposition, a slider was added to the prototype to give more weight to the abovementioned APT features. Figure 14 shows the execution results after adding more weight to the aforementioned features using a slider. The weight of each feature was reassigned using a softmax function according to the value of the slider. Using this process, APT18 was found to be the most similar. The results in Figures 14 and 15 show that the number and type of tactics used by the APT18 group and the threat group in the example scenario are the most similar. Accordingly, it was confirmed that APT18 is assigned a higher weight because of the above characteristics, unlike other groups.



Input		APT18	
Command and Control	1	3	Command and Control
Initial Access	2	2	Initial Access
Execution	1	2	Execution
Privilege Escalation	2	3	Privilege Escalation
Defense Evastion	6	3	Defense Evastion
Persistence	1	4	Persistence
Reconnaissance	1	0	Reconnaissance
Impact	1	0	Impact
Discovery	0	2	Discovery

Current Problem	P
T1589.002:Gather_Victim_Identity_Information	PRE
T1566.001:Phishing:Spearphishing_Attachment	L,W,M
T1219:Remote_Access_Software	L,W,M
T1078:Valid_Accounts	L,W,M..
T1059.003:Command_and_Scripting_Interpreter:Windows_Command_Shell	W
T1036.005:Masquerading:Match_Legitimate_Name_or_Location	L,W,M
T1562.001:Impair_Defenses:Disable_or_Modify_Tools	L,W,M..
T1562.004:Impair_Defenses:Disable_or_Modify_System_Firewall	L,W,M
T1564.001:Hide_Artifacts:Hidden_Files_and_Directories	L,W,M
T1055.003:Process_Injection:Thread_Execution_Hijacking	W
T1565.001:Data_Manipulation:Stored_Data_Manipulation	L,W,M

[APT18] <T1083-CAPEC-555>
- Domain of Attack : Software
- Domain of Mechanism : Subvert Access Control

[Input] <T1036.005-CAPEC-177>
- Domain of Attack : Software, Hardware
- Domain of Mechanism : Subvert Access Control

[APT18, Gorgon] <T1547.001-CAPEC-270>
- Domain of Attack : Software
- Domain of Mechanism : Manipulate System Resources

[Carbanak] <T1543.003-CAPEC-478>
- Domain of Attack : Software
- Domain of Mechanism : Manipulate System Resources

[Gorgon] <T1112-CAPEC-203>
- Domain of Attack : Software
- Domain of Mechanism : Manipulate System Resources

[Input] <T1562.001-CAPEC-578>
- Domain of Attack : Software
- Domain of Mechanism : Manipulate System Resources

APT18	P
T1071.001:Application Layer Protocol: Web Protocols	L,W,M
T1071.004:Application Layer Protocol: DNS	L,W,M
T1547.001:Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	W
T1059.003:Command and Scripting Interpreter: Windows Command Shell	W
T1133:External Remote Services	L,W,M..
T1083:File and Directory Discovery	L,W,M
T1070.004:Indicator Removal on Host: File Deletion	L,W,M
T1105:Ingress Tool Transfer	L,W,M
T1027:Obfuscated Files or Information	L,W,M
T1053.002:Scheduled Task/Job: At (Windows)	W
T1082:System Information Discovery	L,W,M..
T1078:Valid Accounts	L,W,M..

[Gorgon] <T1059.003>
| Sibling
<T1059.001, T1059.005>

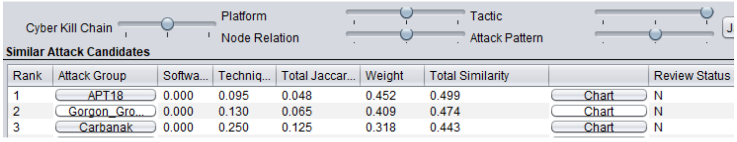
Input		Gorgon	
Command and Control	1	1	Command and Control
Initial Access	2	1	Initial Access
Execution	1	5	Execution
Privilege Escalation	2	2	Privilege Escalation
Defense Evastion	6	6	Defense Evastion
Persistence	1	2	Persistence
Reconnaissance	1	0	Reconnaissance
Impact	1	0	Impact

Gorgon Group	P
T1547.001:Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	W
T1547.009:Boot or Logon Autostart Execution: Shortcut Modification	W
T1059.001:Command and Scripting Interpreter: PowerShell	W
T1059.003:Command and Scripting Interpreter: Windows Command Shell	W
T1059.005:Command and Scripting Interpreter: Visual Basic	L,W,M
T1140:Deobfuscate/Decode Files or Information	L,W,M
T1564.003:Hide Artifacts: Hidden Window	W,M
T1562.001:Impair Defenses: Disable or Modify Tools	L,W,M
T1105:Ingress Tool Transfer	L,W,M
T1112:Modify Registry	W
T1106:Native API	L,W,M
T1566.001:Phishing: Spearphishing Attachment	L,W,M
T1055.002:Process Injection: Portable Executable Injection	W
T1055.012:Process Injection: Process Hollowing	W
T1204.002:User Execution: Malicious File	L,W,M

Input		Carbanak	
Command and Control	1	3	Command and Control
Initial Access	2	1	Initial Access
Execution	1	0	Execution
Privilege Escalation	2	2	Privilege Escalation
Defense Evastion	6	5	Defense Evastion
Persistence	1	2	Persistence
Reconnaissance	1	0	Reconnaissance
Impact	1	0	Impact

Carbanak	P
T1543.003:Create or Modify System Process: Windows Service	W
T1008:Fallback Channels	L,W,M
T1562.004:Impair Defenses: Disable or Modify System Firewall	L,W,M
T1036.004:Masquerading: Masquerade Task or Service	L,W,M
T1036.005:Masquerading: Match Legitimate Name or Location	L,W,M..
T1219:Remote Access Software	L,W,M
T1218.011:Signed Binary Proxy Execution: Rundll32	W
T1078:Valid Accounts	L,W,M..
T1102.002:Web Service: Bidirectional Communication	L,W,M

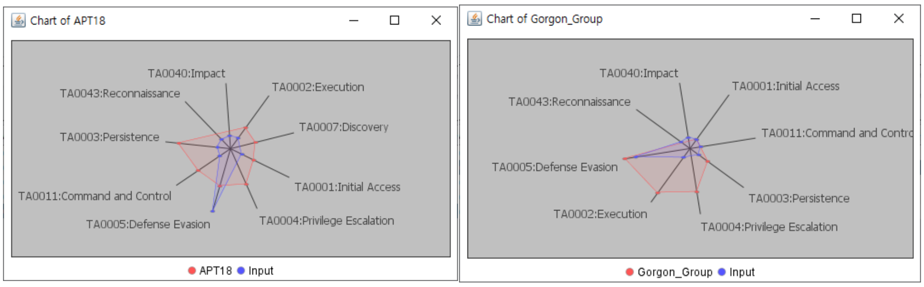
Figure 13. APT feature similarity relationship for the input scenario and each result.



Similar Attack Candidates

Rank	Attack Group	Softwa...	Techniq...	Total Jaccar...	Weight	Total Similarity		Review Status
1	APT18	0.000	0.095	0.048	0.452	0.499	Chart	N
2	Gorgon_Gro...	0.000	0.130	0.065	0.409	0.474	Chart	N
3	Carbanak	0.000	0.250	0.125	0.318	0.443	Chart	N

Figure 14. Result of the similarity metric (with one feature weighted more than the others).

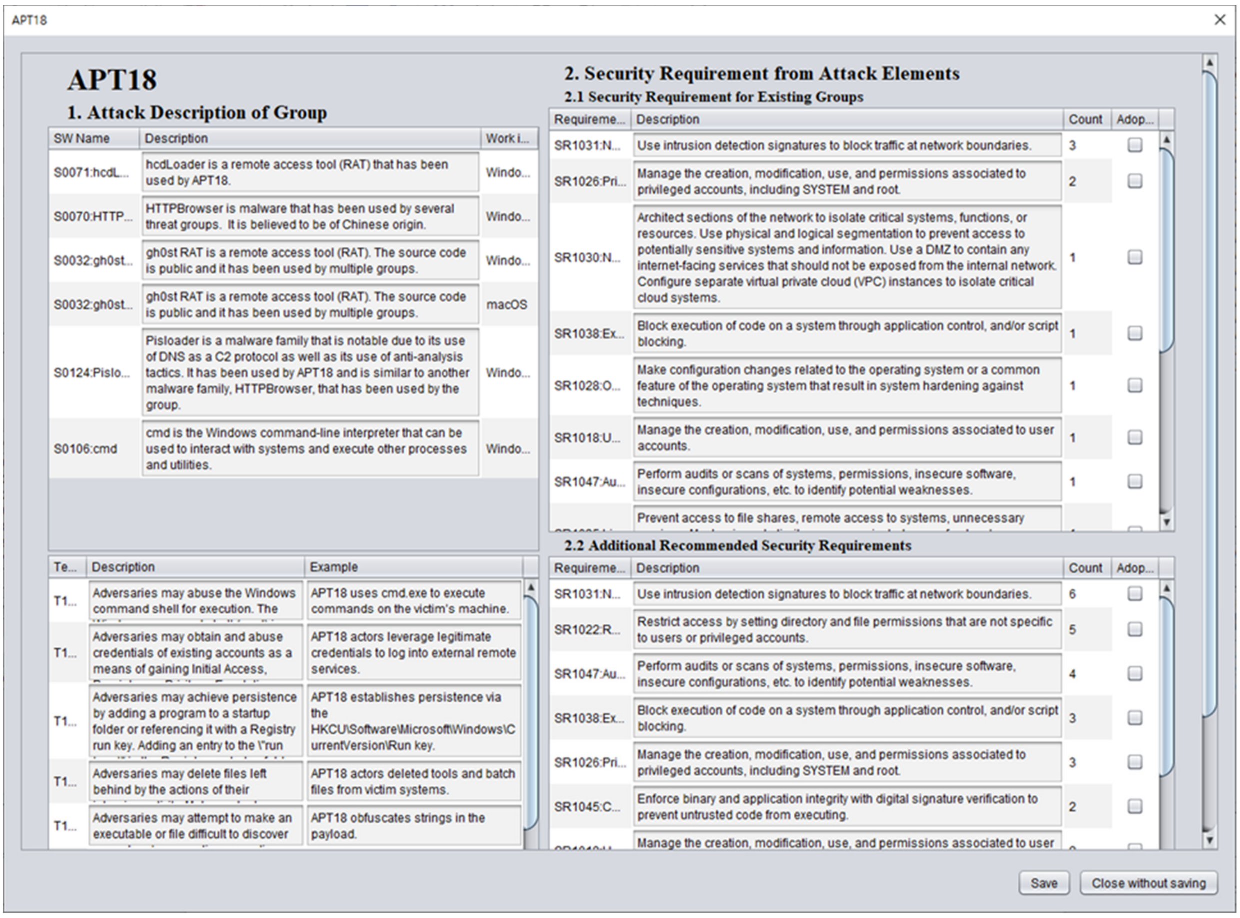


(**a**) (**b**)

Figure 15. Tactical visualization of the top-two groups shown in Figure 14. (**a**) APT18 and (**b**) the Gorgon group.

4.1.5. SP2.3

Proposition SP2.3 provides the user with a detailed description of the software and techniques used in similar cases as well as a list of security requirements needed to prevent attacks. The group name button is clicked to display the information used by the group and the recommended security requirements in a pop-up window. Figure 16 shows the information and security requirements for APT18. The left-hand side shows the software tools and techniques used by the attack group. The software section displays a description of the tool and the operating platform environment. The technique section displays information about the attack and a description of how the group used the attack. The right-hand side shows a list of security requirements recommended for the attack case. The column shows the security requirements index and the description of each requirement, with a count indicating the requirements recommended by the attack elements in descending order. The user may check the security requirements for reuse by reviewing them using the check box. This is a function to identify the security requirements that were adopted if the case is retrieved in the future. In addition, because different requirements can be adopted from each retrieved case, the user shows whether each case has been reviewed and the requirements have been adopted as a check box, as shown in Figure 17. For insufficient or additional requirements that have not been provided, security professionals can perform the case revision step using decision-making and subjective judgment. The solved problem is stored as a case in CB-PDO, and the UVs of the retrieved cases utilized during the process are increased. To evaluate whether our proposed method is valid, we decomposed the study question into propositions, and showed whether each proposition was met by the prototype. Finally, the answers to each proposition are presented, and the proposed method was verified through case studies.



APT18

1. Attack Description of Group

SW Name	Description	Work i...
S0071:hcdL...	hcdLoader is a remote access tool (RAT) that has been used by APT18.	Windo...
S0070:HTTP...	HTTPBrowser is malware that has been used by several threat groups. It is believed to be of Chinese origin.	Windo...
S0032:gh0st...	gh0st RAT is a remote access tool (RAT). The source code is public and it has been used by multiple groups.	Windo...
S0032:gh0st...	gh0st RAT is a remote access tool (RAT). The source code is public and it has been used by multiple groups.	macOS
S0124:Pislo...	Pisloader is a malware family that is notable due to its use of DNS as a C2 protocol as well as its use of anti-analysis tactics. It has been used by APT18 and is similar to another malware family, HTTPBrowser, that has been used by the group.	Windo...
S0106:cmd	cmd is the Windows command-line interpreter that can be used to interact with systems and execute other processes and utilities.	Windo...

Te...	Description	Example
T1...	Adversaries may abuse the Windows command shell for execution. The	APT18 uses cmd.exe to execute commands on the victim's machine.
T1...	Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access,	APT18 actors leverage legitimate credentials to log into external remote services.
T1...	Adversaries may achieve persistence by adding a program to a startup folder or referencing it with a Registry run key. Adding an entry to the \"run	APT18 establishes persistence via the HKCU\Software\Microsoft\Windows\CurrentVersion\Run key.
T1...	Adversaries may delete files left behind by the actions of their	APT18 actors deleted tools and batch files from victim systems.
T1...	Adversaries may attempt to make an executable or file difficult to discover	APT18 obfuscates strings in the payload.

2. Security Requirement from Attack Elements

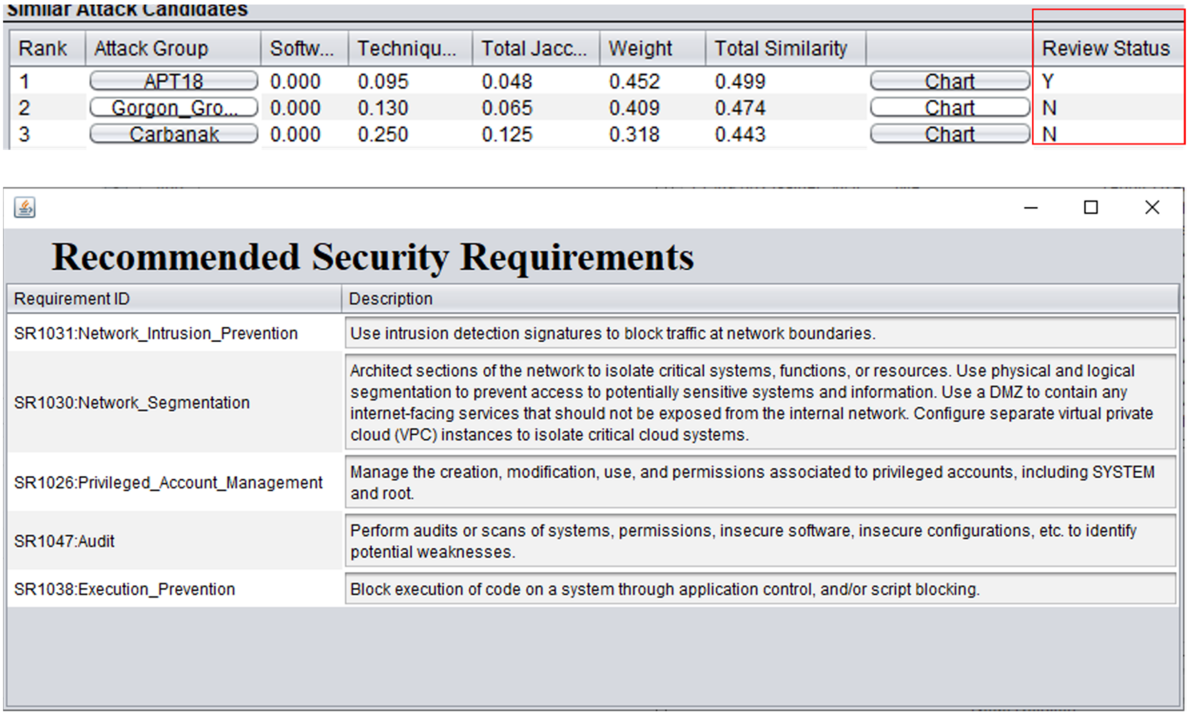
2.1 Security Requirement for Existing Groups

Requireme...	Description	Count	Adop...
SR1031:N...	Use intrusion detection signatures to block traffic at network boundaries.	3	☐
SR1026:Pri...	Manage the creation, modification, use, and permissions associated to privileged accounts, including SYSTEM and root.	2	☐
SR1030:N...	Architect sections of the network to isolate critical systems, functions, or resources. Use physical and logical segmentation to prevent access to potentially sensitive systems and information. Use a DMZ to contain any internet-facing services that should not be exposed from the internal network. Configure separate virtual private cloud (VPC) instances to isolate critical cloud systems.	1	☐
SR1038:Ex...	Block execution of code on a system through application control, and/or script blocking.	1	☐
SR1028:O...	Make configuration changes related to the operating system or a common feature of the operating system that result in system hardening against techniques.	1	☐
SR1018:U...	Manage the creation, modification, use, and permissions associated to user accounts.	1	☐
SR1047:Au...	Perform audits or scans of systems, permissions, insecure software, insecure configurations, etc. to identify potential weaknesses.	1	☐
	Prevent access to file shares, remote access to systems, unnecessary		

2.2 Additional Recommended Security Requirements

Requireme...	Description	Count	Adop...
SR1031:N...	Use intrusion detection signatures to block traffic at network boundaries.	6	☐
SR1022:R...	Restrict access by setting directory and file permissions that are not specific to users or privileged accounts.	5	☐
SR1047:Au...	Perform audits or scans of systems, permissions, insecure software, insecure configurations, etc. to identify potential weaknesses.	4	☐
SR1038:Ex...	Block execution of code on a system through application control, and/or script blocking.	3	☐
SR1026:Pri...	Manage the creation, modification, use, and permissions associated to privileged accounts, including SYSTEM and root.	3	☐
SR1045:C...	Enforce binary and application integrity with digital signature verification to prevent untrusted code from executing.	2	☐
	Manage the creation, modification, use, and permissions associated to user		

Figure 16. Group information and recommended security requirements.



Similar Attack Candidates

Rank	Attack Group	Softw...	Techniqu...	Total Jacc...	Weight	Total Similarity		Review Status
1	APT18	0.000	0.095	0.048	0.452	0.499	Chart	Y
2	Gorgon_Gro...	0.000	0.130	0.065	0.409	0.474	Chart	N
3	Carbanak	0.000	0.250	0.125	0.318	0.443	Chart	N

Recommended Security Requirements

Requirement ID	Description
SR1031:Network_Intrusion_Prevention	Use intrusion detection signatures to block traffic at network boundaries.
SR1030:Network_Segmentation	Architect sections of the network to isolate critical systems, functions, or resources. Use physical and logical segmentation to prevent access to potentially sensitive systems and information. Use a DMZ to contain any internet-facing services that should not be exposed from the internal network. Configure separate virtual private cloud (VPC) instances to isolate critical cloud systems.
SR1026:Privileged_Account_Management	Manage the creation, modification, use, and permissions associated to privileged accounts, including SYSTEM and root.
SR1047:Audit	Perform audits or scans of systems, permissions, insecure software, insecure configurations, etc. to identify potential weaknesses.
SR1038:Execution_Prevention	Block execution of code on a system through application control, and/or script blocking.

Figure 17. Reviewed and adopted security requirements.

4.2. Questionnaire

In this subsection, we present the results of a survey of security experts to verify whether the proposed method would be valid in the field of security. Security experts were targeted as survey participants. Two people with more than 20 years of experience and one person with more than 10 years of experience participated in the survey. The performance tasks were product planning and security consulting. The questionnaire consisted of 25 questions and was based on the ISO9126 standard, which defines the characteristics of software quality and the methods of quality evaluation, reflecting the characteristics that can be evaluated for each question [34,35]. The original questionnaire can be found in [36], and Table 8 categorizes the contents of the questionnaire according to categories and processes. It also shows the quality characteristics and responses for each question. Each response was scored using a value from 0 to 10, and the table shows the minimum, maximum, average, and process-specific averages for the results of each question.

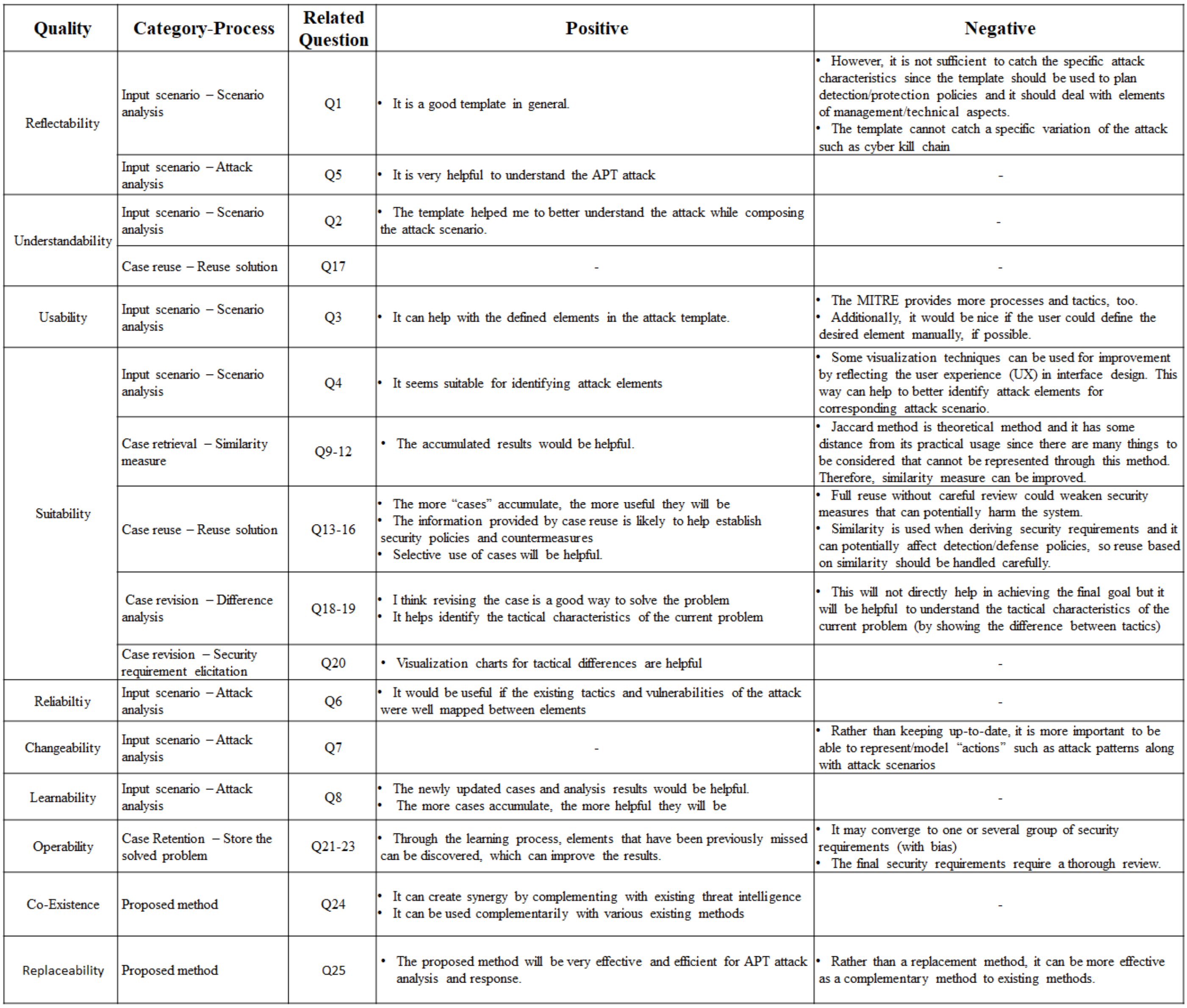
Table 8. Components of and responses to the questionnaire.

Category	Process	Quality	Question	Min	Max	Average	Process Average
Input scenario	Scenario analysis	Reflectability	Q1. Do you think the provided attack template can help reflect APT features?	6	8	6.66	7.91
		Understandability	Q2. Do you think providing these attack templates can help users understand an APT when inputting scenarios?	8	9	8.33	
		Usability	Q3. Do you think providing these attack templates can help users input the attack element into the system?	8	10	8.66	
		Suitability	Q4. Do you think this approach to provide an attack template is a good way for users to understand attacks, identify attack elements, and input attack elements into the system?	7	9	8	
	Attack analysis	Reflectability	Q5. CB-PDO consists of and is interconnected with tactics, platform, attack pattern, weakness, and vulnerability for analysis of the inputted APT elements. Do you think this can help you analyze APT features?	6	9	7.66	7.99

Table 8. *Cont.*

Category	Process	Quality	Question	Min	Max	Average	Process Average
		Reliability	Q6. CB-PDO is based on MITRE ATT&CK, CAPEC, CWE, and CVE, a globally accessible knowledge base for cyberattack tactics and techniques. Do you think this method can help you with the reliability of the APT element analysis results?	8	10	9	
		Changeability	Q7. MITRE ATT&CK, CAPEC, CWE, and CVE, the knowledge bases for cyberattacks, are constantly updated with new attack techniques, tactics, and more. Do you think this can help CB-PDO stay current?	7	8	7.66	
		Learnability	Q8. Do you think this is helpful for APT case analysis?	7	8	7.66	
Case retrieval	Similarity measure	Suitability	Q9. Do you think this will help you recommend security requirements for specific attacks?	6	9	7.33	6.74
			Q10. Do you think using Jaccard similarity as the default baseline helps you measure the similarities between cases and reason similar cases?	5	8	6.33	
			Q11. In addition to Jaccard similarity, review the five rules to consider the similarity between attack techniques. Do you think the five rules help to differentiate between cases using the similarities of the cases?	5	8	6.33	
			Q12. Can selecting goals and targets as elements for case matching help differentiate between cases?	7	7	7	
Case reuse	Reuse solution	Suitability	Q13. This proposed method provides a process to help you decide whether to reuse your security requirements. Can this help you choose the right security requirements?	5	8	6.66	6.86
			Q14. Does this help you decide whether to reuse your security requirements?	7	7	7	
			Q15. The solution is based on MITRE and STIGs guidelines and previous research. Can this help you deliver reliable and appropriate results?	7	9	7.66	
			Q16. The proposed method provides users with the ability to choose to reuse only a subset of the recommended security requirements list. Can this help you reuse your security requirements?	5	7	6.33	
		Understandability	Q17. Determine whether to reuse security requirements based on the security requirements information provided. Can this help you make decisions about reusing security requirements?	5	8	6.66	
Case revision	Difference analysis	Suitability	Q18. The proposed method provides the user with relevant materials to assist in making decisions. Do you think this will help you determine the optimal security requirements?	7	7	7	7.16
			Q19. Can the user understand the tactical differences between cases and help correct security requirements?	6	9	7.33	
	Security requirements elicitation		Q20. The method enables you to create security requirements based on attack components obtained through APT element analysis based on differences in information between cases. Can this help users understand case-to-case tactical differences and create security requirements?	8	8	8	8
Case retention	Storage pf the solved problem	Operability	Q21. Case retention expands the CB-PDO knowledge base and can provide solutions for new attacks. Can these operational methods help to derive optimized security requirements that reflect the APT characteristics for this study?	7	9	8.33	8.22
			Q22. Do you think the attack campaign component is appropriate as a configuration for saving new cases?	8	9	8.33	
			Q23. Repeated CBR improves analysis and prevention of different APTs. Can this process help CB-PDO learn?	7	9	8	
Proposed method	-	Co-existence	Q24. Can our proposed method be used as a complement to existing requirement derivation methods?	8	9	8.66	7.83
		Replaceability	Q25. Can our proposed method be used to replace existing requirement derivation methods?	6	8	7	

The results of the response range between a minimum score of 5 to a maximum score of 10, and the response results for each process show that the scores are evenly distributed in the range of 6 to 8. However, because of the small number of samples collected, opinions reflected in the survey in Figure 18 were collected to specifically evaluate the survey results, and positive/negative opinions were organized for each quality characteristic.



Quality	Category-Process	Related Question	Positive	Negative
Reflectability	Input scenario – Scenario analysis	Q1	• It is a good template in general.	• However, it is not sufficient to catch the specific attack characteristics since the template should be used to plan detection/protection policies and it should deal with elements of management/technical aspects. • The template cannot catch a specific variation of the attack such as cyber kill chain
	Input scenario – Attack analysis	Q5	• It is very helpful to understand the APT attack	-
Understandability	Input scenario – Scenario analysis	Q2	• The template helped me to better understand the attack while composing the attack scenario.	-
	Case reuse – Reuse solution	Q17	-	-
Usability	Input scenario – Scenario analysis	Q3	• It can help with the defined elements in the attack template.	• The MITRE provides more processes and tactics, too. • Additionally, it would be nice if the user could define the desired element manually, if possible.
Suitability	Input scenario – Scenario analysis	Q4	• It seems suitable for identifying attack elements	• Some visualization techniques can be used for improvement by reflecting the user experience (UX) in interface design. This way can help to better identify attack elements for corresponding attack scenario.
	Case retrieval – Similarity measure	Q9-12	• The accumulated results would be helpful.	• Jaccard method is theoretical method and it has some distance from its practical usage since there are many things to be considered that cannot be represented through this method. Therefore, similarity measure can be improved.
	Case reuse – Reuse solution	Q13-16	• The more "cases" accumulate, the more useful they will be • The information provided by case reuse is likely to help establish security policies and countermeasures • Selective use of cases will be helpful.	• Full reuse without careful review could weaken security measures that can potentially harm the system. • Similarity is used when deriving security requirements and it can potentially affect detection/defense policies, so reuse based on similarity should be handled carefully.
	Case revision – Difference analysis	Q18-19	• I think revising the case is a good way to solve the problem • It helps identify the tactical characteristics of the current problem	• This will not directly help in achieving the final goal but it will be helpful to understand the tactical characteristics of the current problem (by showing the difference between tactics)
	Case revision – Security requirement elicitation	Q20	• Visualization charts for tactical differences are helpful	-
Reliabiltiy	Input scenario – Attack analysis	Q6	• It would be useful if the existing tactics and vulnerabilities of the attack were well mapped between elements	-
Changeability	Input scenario – Attack analysis	Q7	-	• Rather than keeping up-to-date, it is more important to be able to represent/model "actions" such as attack patterns along with attack scenarios
Learnability	Input scenario – Attack analysis	Q8	• The newly updated cases and analysis results would be helpful. • The more cases accumulate, the more helpful they will be	-
Operability	Case Retention – Store the solved problem	Q21-23	• Through the learning process, elements that have been previously missed can be discovered, which can improve the results.	• It may converge to one or several group of security requirements (with bias) • The final security requirements require a thorough review.
Co-Existence	Proposed method	Q24	• It can create synergy by complementing with existing threat intelligence • It can be used complementarily with various existing methods	-
Replaceability	Proposed method	Q25	• The proposed method will be very effective and efficient for APT attack analysis and response.	• Rather than a replacement method, it can be more effective as a complementary method to existing methods.

Figure 18. Summary of qualitative evaluation of the proposed method by the experts.

As shown in Figure 18, on the positive side, the experts believe the proposed method is potentially useful for efficient and effective APT attack analysis and security requirements recommendation. However, on the negative side, they address the difficulty of the sophisticated nature of APT attacks, the deficiency of the proposed method, and the need for improvement as a complementary method to existing methods.

Figure 19 shows the average of each software quality characteristic. The reliability, coexistence, and usability characteristics received high scores. This demonstrates that the scenario analysis method is beneficial to the user, employs real defined elements and relationship definitions, and is a positive way to leverage them with existing intelligence. In contrast, replaceability received low scores, although research possibilities for advanced intelligence remain.

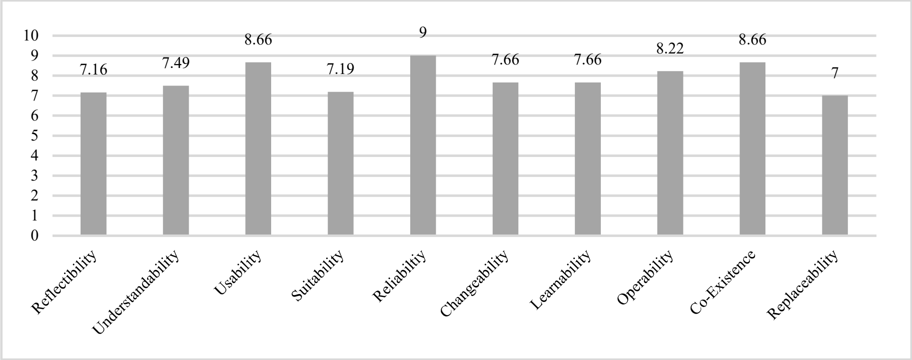


Figure 19. Average of each software quality characteristic.

Through the evaluation performed in this study, we confirmed through a case study that we achieved our research goals. In addition, we identified that our proposed method helps security experts to specify security measures and plans. This is consistent with the research objective of proposing a method to assist security experts in decision-making.

5. Limitations and Discussion

In contrast to the goals of methods in previous security-related studies, the goal of our method is prevention, and we hence propose a novel approach for extracting attack data from previously built knowledge units. We also proposed a method for recommending requirements based on the relevant security measures. However, because these measures all continue to evolve and improve, the following limitations have been identified and should be considered in further research.

First, a full understanding of the domain knowledge of the field is required, and the knowledge base should be updated and maintained throughout its life cycle. Although it is assumed that the provided ontology-based knowledge framework includes attack and security requirements, domain information, and associated risk areas, and these are assumed to be enough to generate some meaningful results, this may not be the case since the problem-related data, information, and knowledge are continuously changing in the real world. Therefore, it is necessary to develop a mechanism to measure the quality of the case-based knowledge base as well as the way to continuously revise and maintain the knowledge base to retain the benefits of reuse. Otherwise, the generated outcomes are no longer trustworthy and eventually may interfere with and harm the related decision-making process.

Second, the number of survey samples was insufficient and more thoroughly designed experiments are needed. To evaluate a newly developed method, the subject matter experts should be fully familiar with it. In this regard, although a half-day tutoring session was provided with the tool demo, we believe the pool of experts was not sufficient. For this reason, although the preliminary results show the potential of the method, the objectivity and reliability of the overall response may be insufficient. In addition, we plan to improve the tools and their user interfaces to help users obtain the required knowledge of basic ontology usage and the analysis process used in the method.

Finally, to perform some of the processes in our proposed method, expert opinions and subjective judgments are required. Appropriate accommodation of experts' opinions is required for case reuse, and a revision plan should be considered for the weakening of security measures due to reuse. Therefore, subjective opinions from security experts are required to reflect and retrieve the final security requirements. To support and facilitate the cycle of CBR, and most importantly assure the correctness of the cases, completeness and consistency of the system as well as management of the updates should be considered in the further development of the method and tools.

On the basis of the feedback provided in the questionnaire, we also confirmed the improvement in the system, which reflects the representation of surrounding environmental factors that take into account future policies or security measures.

6. Summary, Conclusions, and Future Work

In general, the data used in attack detection studies are numerical and quantitative. In contrast, because we include the non-technical elements of APTs as well as the assets and domains, there is a need for a method to include multifarious elements and express them at a single level of abstraction. Moreover, because an APT attack is not performed using a constant and formal form every time, a conceptual integrated knowledge base to understand APTs was built using ontology. To solve the problem using the ontology we built, we adopted an artificial intelligence technique called CBR that can address knowledge units, define case concepts, and propose each step. In this process, we provided an APT attack template that can extract elements to suit the characteristics of the APT attack obtained from the scenario and devise a similarity metric that is suitable for our domain. Thus, equal or unequal weights can be assigned according to the user's subjective judgment, and the results of similar cases are retrieved according to the defined weights. Then, security requirements from the retrieved cases are recommended, and users can make decisions based on APT analysis information and the security requirements provided to adopt solutions for reuse and, if necessary, revise them to address the current problem. In addition, as a function of maintenance and management, the solved problems can then be stored in the CB-PDO, and highly utilized cases can be distinguished according to the UV.

For validation, we defined a study question to evaluate our proposed method, decomposing the study question into general propositions and specific propositions to answer it. Consequently, each proposition was evaluated, and an explainable result was obtained to answer the study question.

In addition, we employed a questionnaire survey of security experts working in security companies to evaluate our method (after a half-day teaching session of the proposed approach with the use of the tools and a prototype demonstration [31]) and analyzed the responses according to process and quality perspectives. Furthermore, we organized the overall positive/negative opinions for specific analysis and demonstrated that our method has the potential to contribute to decision-making as complementary intelligence through the evaluation of the usability and co-existence qualities.

Further studies should be conducted to overcome the limitations identified in Section 5 and enhance the research. The first task will be to build a knowledge base in a completely standardized security field. An ontology similar to WordNet [37], which is a representative language ontology, a system that reflects all the elements involved in the security domain, will be established to reduce the effort needed by users to manipulate or manage the ontology. Second, the sample size will be increased by continuously evaluating the system using students in related majors and security company employees to ensure the reliability of the research results. Finally, the constructed knowledge can be used as a resource for analyzing the correlations and patterns of elements and combining them with the field of deep learning, which is currently being actively studied. For example, using CB-PDO and a generative neural network [38], we could study how weights can be learned according to the combinations and patterns of the techniques used by an attack group. Furthermore, we plan to combine natural language processing and natural language generation methods [39] that address text to automatically generate and recommend security requirements and extend the system to a fully automated method.

Author Contributions: Conceptualization, J.-W.J. and S.-W.L.; methodology, J.-W.J. and S.-W.L.; software, J.-W.J.; validation, J.-W.J. and S.-W.L.; investigation, J.-W.J. and S.-W.L.; resources, S.-W.L.; data curation, J.-W.J.; writing—original draft preparation, J.-W.J.; writing—review and editing, S.-W.L.; visualization, J.-W.J.; supervision, S.-W.L.; project administration, S.-W.L.; funding acquisition, S.-W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Science and ICT (NRF-2020R1F1A1075605), and the BK21 FOUR program of the National Research Foundation of Korea, funded by the Ministry of Education (NRF5199991014091).

Data Availability Statement: Survey Questionnaire Form for the Case Study Designed Methodology and the results are available here (in Korean). Available online: https://docs.google.com/forms/d/1eVdHJ_eJ8wHF_7q3oaZY2NmnEGeQP7svfYJqPI53WbE/edit?usp=sharing (accessed on 11 November 2021).

Conflicts of Interest: The authors have no conflict of interest to declare. The funders had no role in the study design; the collection, analysis, or interpretation of data; the writing of the manuscript; or decision to publish the results.

References

1. 2021 Data Breach Investigations Report (DBIR) by Verizon. Available online: https://vigitrust.com/wp-content/uploads/2021/08/John-Grim-2021-DBIR-Patterns-in-Data-Breaches.pdf?msclkid=5d208ed5cea911ec9606a7306c2c091d (accessed on 11 November 2021).
2. Security Priority Studies 2022 by Foundry, an IDG Incorporated. Available online: https://www.idg.com/tools-for-marketers/research-security-priorities/ (accessed on 11 November 2021).
3. APT Trends Report Q3 2021 by Global Research & Analysis Team. Kaspersky. Available online: https://securelist.com/apt-trends-report-q3-2021/104708/ (accessed on 11 November 2021).
4. Chen, P.; Desmet, L.; Huygens, C. A study on advanced persistent threats. In Proceedings of the IFIP International Conference on Communications and Multimedia Security, Aveiro, Portugal, 25–26 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 63–72.
5. Singh, S.; Sharma, P.K.; Moon, S.Y.; Moon, D.; Park, J.H. A comprehensive study on APT attacks and countermeasures for future networks and communications: Challenges and solutions. *J. Supercomput.* **2019**, *75*, 4543–4574. [CrossRef]
6. Proactively Detect Persistent Threats—Cyber Kill Chain by Lockheed Martin. Available online: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html (accessed on 11 November 2021).
7. Yadav, T.; Mallari, R.A. Technical Aspect of Cyber Kill Chain. *arXiv* **2016**, arXiv:1606.03184.
8. MITRE ATT&CK vs. Cyber Kill Chain vs. Diamond Model, CyCraft Technology Corp. CyCraft Blog. Available online: https://medium.com/cycraft/cycraft-classroom-mitre-att-ck-vs-cyber-kill-chain-vs-diamond-model-1cc8fa49a20f (accessed on 11 November 2021).
9. MITRE. ATT&CK. Available online: https://attack.mitre.org/ (accessed on 11 November 2021).
10. Lee, S.-W.; Gandhi, R.; Muthurajan, D.; Yavagal, D.; Ahn, G.-J. Building problem domain ontology from security requirements in regulatory documents. In Proceedings of the 2006 International Workshop on Software Engineering for Secure Systems, Shanghai, China, 20–22 May 2006; pp. 43–50.
11. Kim, B.-J.; Lee, S.-W. Understanding and recommending security requirements from problem domain ontology: A cognitive three-layered approach. *J. Syst. Softw.* **2020**, *169*, 110695. [CrossRef]
12. Kim, M.; Park, S.-H.; Lee, S.-W. A Security Requirements Recommendation Framework Based on APT Attack Case. *J. KIISE* **2021**, *48*, 1014–1026. [CrossRef]
13. Roger, S.; Robert, A. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*; Psychology Press: London, UK, 2013.
14. Kolodner Janet, L. An introduction to case-based reasoning. *Artif. Intell. Rev.* **1992**, *6*, 3–34. [CrossRef]
15. Lee, J.H.; Kim, H.S. eHealth Recommendation service system using ontology and case-based reasoning. In Proceedings of the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015; IEEE: New York, NY, USA, 2015; pp. 1108–1113.
16. Sebag, M.; Schoenauer, M. A rule-based similarity measure. In *European Workshop on Case-Based Reasoning*; Springer: Berlin/Heidelberg, Germany, 1993; pp. 119–131.
17. Formica, A. Ontology-based concept similarity in formal concept analysis. *Inf. Sci.* **2006**, *176*, 2624–2641. [CrossRef]
18. Aamodt, A.; Plaza, E. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun.* **1994**, *7*, 39–59. [CrossRef]
19. Hernandez-Nieves, E.; Hernández, G.; Gil-Gonzalez, A.B.; Rodríguez-González, S.; Corchado, J.M. CEBRA: A CasE-Based Reasoning Application to recommend banking products. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104327. [CrossRef]
20. Bokolo, A., Jr. A case-based reasoning recommender system for sustainable smart city development. *AI Soc.* **2021**, *36*, 159–183.
21. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.
22. Park, S.-H.; Jung, J.-W.; Lee, S.-W. Multi-perspective APT Attack Risk Assessment Framework using Risk-Aware Problem Domain Ontology. In Proceedings of the 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW), Notre Dame, IN, USA, 20–24 September 2021; IEEE: New York, NY, USA, 2021; pp. 400–405.
23. Protégé. Available online: http://protege.stanford.edu/ (accessed on 11 November 2021).

24. Common Attack Pattern Enumerations and Classifications (CAPEC). Available online: https://capec.mitre.org/ (accessed on 11 November 2021).
25. Common Weakness Enumeration (CWE). Available online: https://cwe.mitre.org/ (accessed on 11 November 2021).
26. Common Vulnerabilities and Exposures (CVE). Available online: https://cve.mitre.org/ (accessed on 11 November 2021).
27. Security Technical Implementation Guide (STIGs). Available online: https://public.cyber.mil/stigs/srg-stig-tools/ (accessed on 11 November 2021).
28. Operation Wilted Tulip—Exposing a Cyber Espionage Apparatus, Trend Micro. ClearSky Cyber Security. Available online: https://www.clearskysec.com/wp-content/uploads/2017/07/Operation_Wilted_Tulip.pdf?msclkid=6c7b3063cebb11ecbc76539ac6d132d4/ (accessed on 11 November 2021).
29. Carbanak APT The Great Bank Robbery—Kaspersky. Available online: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/08064518/Carbanak_APT_eng.pdf (accessed on 11 November 2021).
30. SPARQL 1.1 Overview. W3C. Available online: https://www.w3.org/TR/sparql11-overview/ (accessed on 11 November 2021).
31. Jung, J.-W.; Park, S.-H.; Lee, S.-W. A Tool for Security Requirements Recommendation using Case-Based Problem Domain Ontology, Tools and Demo track RE 2021. In Proceedings of the 29th IEEE International Requirements Engineering Conference 2021, Notre Dame, IN, USA, 20–24 September 2021.
32. Lee, S.-W.; Rine, D.C. Case Study Methodology Designed Research in Software Engineering Methodology Validation. In Proceedings of the Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE), Banff, AB, Canada, 20–24 June 2004; pp. 117–122.
33. The Gorgon Group: Slithering Between Nation State and Cybercrime, Paloalto Networks. Available online: https://unit42.paloaltonetworks.com/unit42-gorgon-group-slithering-nation-state-cybercrime/ (accessed on 11 November 2021).
34. ISO/IEC 9126 in Software Engineering, Geeks for Geeks. Available online: https://www.geeksforgeeks.org/iso-iec-9126-in-software-engineering/ (accessed on 11 November 2021).
35. Bevan, N. Quality in Use: Meeting User Needs for Quality. *J. Syst. Softw.* **1999**, *49*, 89–96. [CrossRef]
36. Survey Questionnaire Form for the Case Study Designed Methodology (in Korean). Available online: https://docs.google.com/forms/d/1eVdHJ_eJ8wHF_7q3oaZY2NmnEGeQP7svfYJqPI53WbE/edit?usp=sharing (accessed on 11 November 2021).
37. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
38. Sanchez-Lengeling, B.; Reif, E.; Pearce, A.; Wiltschko, A.B. A Gentle Introduction to Graph Neural Networks. *Distill* **2021**, *6*, e33. [CrossRef]
39. Gatt, A.; Krahmer, E. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *J. Artif. Intell. Res.* **2018**, *61*, 65–170. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.