applied sciences

MDPI

Article

GENICS: A Framework for Generating Attack Scenarios for Cybersecurity Exercises on Industrial Control Systems

InSung Song [1], Seungho Jeon [2], Donghyun Kim [1], Min Gyu Lee [1] and Jung Taek Seo [3,*]

[1] Department of Information Security, Gachon University, Seongnam-daero 1342, Seongnam-si 13119, Republic of Korea; chris0@gachon.ac.kr (I.S.); 202240222@gachon.ac.kr (D.K.); cpsmgyu@gmail.com (M.G.L.)
[2] Department of Computer Engineering (Smart Security), Gachon University, Seongnam-daero 1342, Seongnam-si 13119, Republic of Korea; shjeon90@gachon.ac.kr
[3] Department of Computer Engineering, Gachon University, Seongnam-daero 1342, Seongnam-si 13119, Republic of Korea
* Correspondence: seojt@gachon.ac.kr

Abstract: Due to the nature of the industrial control systems (ICS) environment, where process continuity is essential, intentionally initiating a cyberattack to check security controls can cause severe financial and human damage to the organization. Therefore, most organizations operating ICS environments check their level of security through simulated cybersecurity exercises. For these exercises to be effective, high-quality cyberattack scenarios that are likely to occur in the ICS environment must be assumed. Unfortunately, many organizations use limited attack scenarios targeting essential digital assets, leading to ineffective response preparedness. To derive high-quality scenarios, there is a need for relevant attack and vulnerability information, and standardized methods for creating and evaluating attack scenarios in the ICS context. To meet these challenges, we propose GENICS, an attack scenario generation framework for cybersecurity training in ICS. GENICS consists of five phases: threat analysis, attack information identification, modeling cyberattack scenarios, quantifying cyberattacks, and generating scenarios. The validity of GENICS was verified through a qualitative study and case studies on current attack scenario-generating methods. GENICS ensures a systematic approach to generate quantified, realistic attack scenarios, thereby significantly enhancing cybersecurity training in ICS environments.

Keywords: cybersecurity exercise; industrial control systems; cyberattack scenarios; cyber physical system

Citation: Song, I.; Jeon, S.; Kim, D.; Lee, M.G.; Seo, J.T. GENICS: A Framework for Generating Attack Scenarios for Cybersecurity Exercises on Industrial Control Systems. *Appl. Sci.* **2024**, *14*, 768. https://doi.org/10.3390/app14020768

Academic Editor: Anyang Lu

Received: 22 November 2023
Revised: 12 January 2024
Accepted: 13 January 2024
Published: 16 January 2024

1. Introduction

Operators of industrial control systems (ICS), such as factories or power plants, should establish appropriate security controls to protect the assets contained in the facility [1]. In addition, it should be confirmed that the security controls operate correctly through appropriate means. However, as many incidents in the past have shown [2,3], due to the nature of the ICS environment where process continuity is essential, checking security controls by intentionally inducing cyberattacks can cause severe financial and human damage to organizations. Therefore, organizations operating ICS environments check their security level through simulated cybersecurity exercises [4,5]. The cybersecurity exercise is an educational process that strengthens the ability of an agency or organization's employees to respond to, prevent, and protect information assets from cyber threats. These exercises include training in the basic principles of cybersecurity, managing security risks, and strategies to deal with emerging threats. In 2002, the U.S. federal government enacted the Federal Information Security Management Act (FISMA) [6] to strengthen the security of federal information systems. Under FISMA, all federal agencies are required to ensure the security of information systems, and the law requires organizations to provide security

awareness training to all users of information systems. In addition, the Cybersecurity and Infrastructure Security Agency (CISA), a primary U.S. agency in charge of cybersecurity, published "Cybersecurity Incident and Vulnerability Response Playbooks" [7]. This publication helps to plan to respond to cybersecurity incidents and vulnerabilities in federal civilian executive branch (FCEB) information security systems. Other countries enact laws to protect their businesses and assets.

For simulated cybersecurity exercises to be effectively conducted in the ICS environment, high-quality cyberattack scenarios that are highly likely to occur in the ICS environment should be assumed. The cyberattack scenarios are mainly written from the attacker's point of view and describe in detail the procedures they go through to achieve the attacker's objective. In order to write such a cyberattack scenario, information such as assets and network topology included in the ICS environment should be collected. Unfortunately, most organizations repeatedly use limited attack scenarios for training against some essential digital assets. If cyber exercises are conducted with limited attack scenarios, the adequacy of the organization's security controls to protect its assets cannot be verified appropriately. Furthermore, the security controls cannot be revised to suit the organization. Therefore, a standardized framework that can derive various expected attack scenarios that can be applied to facilities is required for practical cyberattack response exercises in the ICS environment.

There are several challenges to writing a high-quality cyberattack scenario. First, information on vulnerabilities for attacks that can be applied to assets within the ICS environment needs to be systematized. This not only reduces the understanding of the system but also makes it challenging to derive realistic scenarios for training. Furthermore, it is difficult to establish a response strategy for the hypothesized cyberattack scenario. Second, there is a lack of standardized methods for generating attack scenarios. This can lead to incomplete threat modeling by missing critical attack steps in the attack scenario. In addition, writing attack scenarios without a systematic methodology leads to inefficient training. Third, there is a lack of methods to evaluate the scenarios' quality. Measuring and evaluating the quality of the scenarios ensures the effectiveness of training and enhances participants' learning experience. If the quantitative or qualitative evaluation of the attack scenario is omitted, the effectiveness of the training cannot be measured, and it cannot be judged whether the participants acquired the knowledge and skills to counter cyber threats through the training.

Our insights to overcome the above challenges are as follows. First, we introduce the adversarial tactics, techniques, and common knowledge (ATT&CK) framework [8] developed by MITRE to identify possible attacks or inherent vulnerabilities on assets. MITRE ATT&CK is a knowledge base on cybersecurity attacks, defining the tactics, techniques, and procedures attackers use to infiltrate systems, move within networks, extract valuable information, or destroy systems. Next, attack trees [9,10] are used to generate cyberattack scenarios systematically. An attack tree is a graphical tool used to analyze vulnerabilities in security systems. An attack tree is a visual representation of the different ways to achieve an attack goal, which helps evaluate and improve a system's security state. Finally, we utilize the common vulnerability scoring system (CVSS) [11] and DREAD [12] to evaluate the quality of generated cyberattack scenarios. CVSS is a scoring system widely used to evaluate the severity of security vulnerabilities. DREAD stands for damage, reproducibility, exploitability, affected users, and discoverability and evaluates the risk of vulnerabilities by scoring each factor. However, since CVSS and DREAD were designed to evaluate only the severity of a single vulnerability, it is difficult to use them to evaluate attack scenarios directly. Therefore, we extend CVSS and DREAD to evaluate attack scenarios.

In this paper, we propose an attack scenario generation framework for cybersecurity exercises in ICS. This framework consists of a preparation phase and four phases for attack scenario generation and evaluation: preparation, threat analysis (step 1), attack information identification (step 2), modeling cyberattack (step 3), quantitative evaluation (step 4), and attack scenario generation (step 5). The preparation phase collects the schematic

diagram for the ICS environment and identifies the assets included in it. These assets may include personal computers (PCs), engineering workstations (EWS), programmable logic controllers (PLCs), and field devices. Since each attack procedure that constitutes a cyberattack scenario mostly targets these assets, the assets should be identified prior to writing the attack scenario. The threat analysis step establishes an outline of cyberattack scenarios for training. This step defines the objective of the attacker's attack, the ultimate target within the facility, and the attack's impact. The attack information identification stage collects information such as attack tactics, techniques, and vulnerabilities that can be applied to the assets in the facility identified in the preparation stage. In the attack scenario generation step, attack scenarios for exercises are generated based on the previously collected information. We propose a method to configure a modified attack tree to achieve the previously defined attack goal and generate an attack scenario. Lastly, we evaluate the attack scenario generated in the previous step in the quantitative evaluation step. We use CVSS and DREAD to score each attack procedure that constitutes an attack scenario and evaluate the level of the attack scenario by integrating the scores of all attack procedures.

Recognizing the challenges inherent in assessing security controls within the ICS environment, GENICS provides a structured approach to validate the current security posture of facilities through practical exercises. GENICS stands out by integrating established knowledge bases, including MITRE ATT&CK and CVE databases, to construct detailed attack scenarios tailored to ICS facilities. The framework distinguishes itself by systematically mapping out attack procedures using an attack tree model, which not only elucidates potential threat pathways but also enables the quantification of each step within the attack vector. This methodical representation ensures that each asset is accounted for, from the initial entry point to the final target, facilitating the development of a solid defense strategy. Applying GENICS in cybersecurity exercises promises to significantly enhance training outcomes, ensuring that participants are exposed to realistic attack vectors and are equipped with the knowledge to respond effectively. The quantified approach that GENICS introduces guarantees a more measured and insightful experience for trainees, enabling them to understand the intricacies of cyber threats within the ICS landscape. In addition to its efficiency in creating cyber-attack scenarios for training purposes, the GENICS framework also demonstrates a proactive capability in identifying potential cyber threats within systems, enabling pre-emptive response strategies. This dual functionality enhances both the quality of training and the overall cybersecurity readiness of ICS.

The contributions of our paper are as follows:

- We propose an attack scenario generation framework for cybersecurity exercises in ICS.
- We propose a method for deriving attack scenarios on ICS facilities by utilizing well-known knowledge bases such as MITRE ATT&CK and CVE.
- Systematic representation of attack scenarios using attack trees and the quantification of each procedure within these scenarios.

The remainder of this paper is organized as follows. Section 2 presents the background for describing studies on cyberattack scenario generation and the proposed framework. In Section 3, we describe a cyberattack scenario generation framework for exercises in an ICS environment. Section 4 performs a qualitative study and case study on the proposed framework. Section 5 discusses the limitations of the framework, and in Section 6, we present the conclusions of this study and future work.

2. Background and Related Work

2.1. Background

2.1.1. MITRE Adversarial Tactics, Techniques and Common Knowledge

MITRE ATT&CK [8] is an open-access knowledge base that subdivides and structures attacks from the attacker's point of view based on data derived from actual security incidents. MITRE ATT&CK provides integrated information on tactics used by attackers to achieve goals and techniques for achieving tactics and is classified into enterprise, mobile, and ICS depending on the field. It is mainly used for defining attacks with attack trees,

graphs, paths, and sequences in threat modeling [13–17]. In this paper, GENICS refers to tactics/techniques of MITRE ATT&CK for enterprise and ICS for attack identification in cyberattack scenarios. MITRE ATT&CK for enterprise provides 14 tactics and 227 techniques, and MITRE ATT&CK for ICS provides 12 tactics and 92 techniques. MITRE ATT&CK for enterprise/ICS includes a total of 16 tactics, and detailed explanations are as follows.

Reconnaissance, such as active scanning, is a method by which an attacker collects information about a target organization or system. Resource development is a method by which an attacker develops or acquires tools, malware, and vulnerabilities necessary for an attack, such as establishing accounts. The initial access is how an attacker gains access to a system or network, such as hardware additions or supply chain compromise. Execution is a method that adversely affects the system by executing malicious codes or actions, such as execution using native application programming interfaces (APIs). Persistence is a method by which an attacker exists in the system or activates a specific function to maintain access permissions, such as account manipulation or activation of external remote services. Privilege escalation is a method of obtaining higher privileges within the system by extending the initial privileges acquired by the attacker, such as securing valid accounts. Defense evasion (evasion), like debugger evasion, is a method by which attackers circumvent security systems and tools to avoid detection and response. Credential access, like brute force, is a method by which an attacker obtains valid credentials within the system. The discovery, like network sniffing, is a method in which an attacker performs exploration within the system to obtain information about the architecture and environment of the target system. The lateral movement, like replication through removable media, is a method in which an attacker moves within a system to expand access permissions or find vulnerable systems. The collection is a way to collect information targeted by an attacker, such as screen capture. The command-and-control method is when an attacker remotely transmits and controls commands to a malicious code or system, such as proxy communication. The inhibit response function, like denial of service, is a method by which an attacker hinders or interferes with the target organization's ability to respond to security. The exfiltration, like exfiltration over alternative protocol, is a method of leaking data or information an attacker steals to the outside. The impair process control, such as spoof reporting messages, is a method by which an attacker manipulates a process or control system in a system to disrupt or damage its operation. Lastly, the impact, like data manipulation, is a method by which an attacker exerts a destructive influence on the system to achieve the final goal.

2.1.2. Attack Tree

An attack tree [9] is a well-known threat modeling technique. The attack tree expresses possible security threats to the system through a tree and sequentially connects a series of actions to achieve the goal. The attack tree generally consists of root, leaf, and intermediate nodes [10]. The root node is at the top of the tree, meaning the attack's ultimate goal. The leaf node (or the terminal node) is located at the bottom of the attack tree, consisting of initial attacks to reach the upper node. The intermediate node represents the condition to direct the upper node to achieve the goal. These nodes are generally connected to upper nodes by gates, such as AND and OR gates. The AND gate indicates that all lower nodes must be successfully executed to reach the upper node (subgoal or goal). On the other hand, the OR gate indicates that the upper node can be reached even if only one of the lower nodes is successfully executed. The attack tree is used in various threat modeling studies [18,19]. Recently, an augmented attack tree with notions, such as sequential AND (SAND) gate and priority, has been employed by extending the conventional attack tree [20–23].

2.1.3. Common Vulnerability Scoring System

The CVSS [11] is a framework for evaluating and quantifying the vulnerabilities' impact. The CVSS score is expressed on a scale of 0 to 10, allowing us to rate the severity of a given vulnerability to prioritize a response. Currently, CVSS v3.1 classifies various evalua-

tion factors into three groups (the base metric, the temporal metric, and the environmental metric) to evaluate vulnerabilities.

The base metric consists of the access vector, access complexity, privileges required, user interaction, scope, and impact. The access vector represents the path (the network, adjacent network, and local, physical) an attacker uses to gain access to exploit a vulnerability. The attack complexity indicates how complex an attacker should go through to exploit a vulnerability and is rated as 'low' or 'high'. The privileges required indicate what level of privilege an attacker should have to exploit the vulnerability and evaluates to 'none', 'low', and 'high'. The user interaction indicates whether an attacker needs to interact with the user to exploit the vulnerability and evaluates it to 'none' and 'require'. The scope indicates the scope the vulnerability affects and is evaluated as 'unchanged' or 'changed'. The impact is divided into confidentiality, integrity, and availability, evaluated as 'none', 'low', and 'high', respectively. The temporal metrics include exploit code maturity, remediation level, and report confidence. The exploit code maturity indicates how mature the exploit code for a vulnerability is, rated as 'not defined', 'unproven that exploit exists', 'proof of concept code', 'functional exploit exists', and 'high'. The remediation level represents the organization's response to vulnerabilities, evaluated as 'not defined', 'official fix', 'temporary fix', 'workaround', and 'unavailable'. The report confidence indicates the reliability of reporting on vulnerabilities and is evaluated as 'not defined unknown', 'reasonable', and 'confirmed'. The environmental metric is a category of the base metric and consists of confidentiality, integrity, and availability requirements. The confidentiality requirement represents the required level of confidentiality/integrity/availability of the system or data affected by the vulnerability and is evaluated as 'not defined', 'low', 'medium', and 'high'. The CVSS score is calculated using the base metric, and the temporary and environmental metrics are used to provide additional information.

2.1.4. DREAD

DREAD [12] is one of the threat modeling techniques prioritizing threats. The priority is determined by assigning scores from 1 to 10 to five factors (Damage, Reproducibility, Exploitability, Affected Users, and Discoverability). The damage indicates the damage that could occur if the vulnerability were exploited. The reproducibility represents the degree of difficulty of the conditions necessary to reproduce the vulnerability, and the lower the reproducibility, the more difficult it is to exploit. The exploitability indicates the degree of technical ability or resources required to exploit a vulnerability, and the higher the exploitability, the easier the vulnerability can be exploited. The affected users indicate the number of users affected by the vulnerability. The discoverability indicates the probability that a vulnerability can be discovered; the higher it is, the faster it can be discovered. DREAD has been used to evaluate attacks in various threat modeling or risk assessment studies [24,25].

2.2. Related Work

Nguyen et al. [15] interfaced the attack planner's knowledge base with both MITRE common vulnerabilities and exposures (CVE) and the National Institute of Standards and Technology (NIST) national vulnerability database (NVD) using the BRON framework [26] to create a detailed attack tree. Kern et al. [27] proposed a model-based semi-automated attack tree generation method considering attack motive and functional dependency to solve the labor-intensive attack path generation problem required for vehicle cyber security risk analysis.

Ferda et al. [16] proposed Attack Dynamics, an automated attack graph generation framework finding attack paths that can identify attack types using MITRE common attack pattern enumeration and classification (CAPEC) and MITRE common weakness enumeration (CWE). Ibrahim et al. [28] presented attack graphs causing plant shutdowns in nuclear power plants, vehicular network systems, and ICS using models considering system and security attributes, implementing cyberattack scenarios from the graphs.

Serru et al. [13] experimented with analyzing the impact of multi-level cyberattacks on the safety of cyber-physical systems (CPS) using discrete event simulations (DES). In this experiment, the navigation system of an autonomous ship is expressed with the modeling language AltaRica, and attack sequences are automatically extracted to indicate all events affecting the system. Choi et al. [14] proposed an automatic generation method of attack sequences based on MITRE ATT&CK's tactics/techniques to add various and realistic attack sequences to the ICS dataset. Takahashi et al. [17] proposed APTGen for generating a targeted attack dataset to solve the problem of accelerating incident response and lack of attack sequence information. APTgen generated an attack sequence using MITRE ATT&CK's tactics/techniques and created a dataset based on eight real-world cases. In Dutta et al. [29], CAgen, a framework generating attack sequences for cyber-physical energy systems such as power grids, is proposed. CAgen was based on an electric high-risk failure (EHFS) scenario; CAgen started with generating the attack tree, mapped CWE/CVE, and added details to the attack sequence.

Polatidis et al. [30] proposed an attack graph-based risk management framework developed for maritime supply chains. This framework used parameters such as attacker location and maximum propagation length in the attack path search process and adopted a priority algorithm to identify non-recursive attack paths. Islam et al. [31] proposed a seven-stepped attack path generation method considering the characteristics of medical information infrastructure. The attack vectors were created by adding parameters to the MITIGATE method. This framework used existing proven frameworks such as CVE/CVSS in the attack path generation process. It developed a knowledge base of nine parameters through rule-based reasoning to identify the attack path. Kavallieratos et al. [32] proposed a method of searching and analyzing attack paths targeting the CPS environment. The proposed method identified the importance of the attack path by considering the risk estimation of each component of CPS and the importance to the system through DREAD. Polatidis et al. [33] proposed a method for predicting attacks from attack vectors identified through a rule-based inference approach. The attack path consisted of four steps, and the attack path was created as a vulnerability chain. The generated attack path was used as an input value for the collaborative filtering recommendation system, and a predicted attack was derived.

3. GENICS Framework: A Strategy for Generating Attack Scenarios in Industrial Control Systems Cybersecurity Exercises

In this section, we propose GENICS, a framework to generate cyberattack scenarios. This framework systematically derives attack scenarios that can occur in the ICS environment by using the existing cybersecurity-related knowledge bases. In addition, the generated scenario and its suitability for cyber-incident response exercises in the ICS environment are evaluated quantitatively. Here and after, we consider an attack scenario to consist of several attack procedures that should be executed to achieve an attacker's goal.

Figure 1 shows an overview of the proposed framework. The proposed framework is divided into a total of five steps, that are performed after the preparation: threat analysis (step 1), attack information identification (step 2), modeling cyberattack scenarios (step 3), quantifying cyberattack scenarios (step 4), and generating cyberattack scenario (step 5). In the preparation step, all assets attackers may be interested in deployed in the ICS. The threat analysis step outlines the cyberattack scenario by defining the attacker's attack goal, the attack target to achieve the goal, and the attack's impact. The attack information identification step configures the attack procedure, and MITRE ATT&CK's tactics are adopted to assign the attack technique to the corresponding attack procedure. In the attack scenario modeling step, the attack tree is constructed with the analyzed information produced in the previous steps; we augment and modify the standard attack tree to present analyzed information appropriate to the ICS environment. In the attack scenario generation step, attack scenarios are derived from the attack tree. In the quantification step, the scenarios generated are evaluated through the CVSS and DREAD techniques.

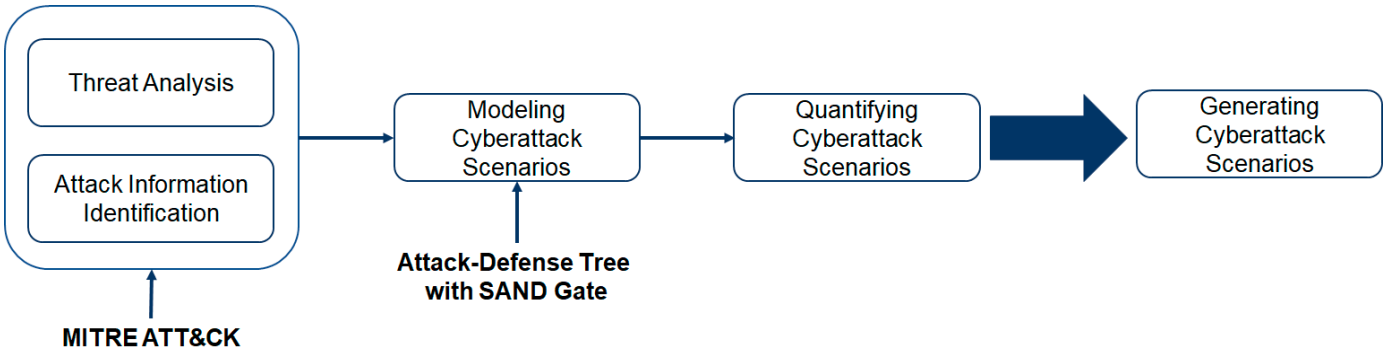


Figure 1. Overview of GENICS Framework for Generating Attack Scenario.

Preparation. The preparation step identifies the assets deployed in the ICS. The assets included in the ICS environment are largely categorized into physical assets, network assets, and software assets. The physical assets are responsible for process control or monitoring and may include assets such as PLCs, EWS, human-machine interfaces (HMIs), and field devices. The network assets include devices for segmenting or connecting multiple sub-networks in the industrial networks and protocols for each communication segment. This can include devices such as switches and routers. Software assets are programs that control physical assets or manage the entire process. We should utilize the ICS environment's schematic diagram or network architecture to identify these assets. Unidentified assets will be excluded from threat modeling or scenario generation if this step is not performed accurately. This may result in deriving an incomplete attack scenario.

3.1. Step 1: Threat Analysis

An outline of cyberattacks for exercise is established in the threat identification step. To outline a cyberattack, the attack's objective, target, and impact should be defined. The attack's objective is to describe the human, economic, and social events that may result from an attack. The attack's target is an asset in the ICS that should be neutralized to achieve the purpose of the attack in the last stage. For the attack's target, the asset's name and the system in which the asset is installed should be specified together to represent the asset accurately. To specify the impact, the cyberattack's effect on the ICS should be written. More specifically, we state the attack's impact from the point of view the impact on the target asset, the system, and the entire ICS.

3.2. Step 2: Attack Information Identification

In the attack information identification step, the assets identified in the previous step are mapped to MITRE ATT&CK tactics. We should determine whether a given asset can be utilized in the specific tactic of MITRE ATT&CK. Then, one should assign the attack techniques and vulnerabilities to the asset according to the tactics. The attack information (assets, attack techniques, vulnerabilities) for each asset defined in this step is used for attack tree modeling (i.e., nodes) in the next step.

Table 1 is an example of mapping attack tactics of MITRE ATT&CK for ICS to representative assets used in ICS. The attack tactics mapped with each asset can identify the attacker's action on each asset. MITRE ATT&CK provides cybersecurity incidents observed in the past. Therefore, if there is an incident case similar to the scenarios' overview, the attack techniques to be used in the scenario can be easily identified based on the case.

Table 1. Mapping between assets and MITRE ATT&CK's tactics.

Asset	Attack Tactic
Internal network PC	Initial Access Execution Persistence Privilege Escalation Evasion Collection Command and Control Discovery Lateral Movement
HMI	Initial Access Lateral Movement Execution Persistence Privilege Escalation Evasion Collection Command and Control Inhibit Response Function Impair Process Control Impact
EWS	Initial Access Lateral Movement Execution Persistence Privilege Escalation Evasion Collection Command and Control Discovery Lateral Movement Inhibit Response Function Impair Process Control Impact
PLC	Initial Access Lateral Movement Execution Persistence Privilege Escalation Evasion Collection Inhibit Response Function Impair Process Control Impact

Using an internal network PC as an example, Initial Access can collect virtual private network (VPN) credentials and use them to gain access to the internal network for an attack. Execution may involve using a command-line interface. An attacker can use xp_cmdshel and Powershell on an MS-SQL server to execute commands. Persistence may involve hardcoding a username or password. A hardcoded password on a database server can be used to propagate an attack to other systems. Privilege Escalation can involve exploiting a vulnerability in software to gain elevated privileges. Vulnerabilities such as [34] can be used to perform privilege escalation on Windows systems. Evasion can involve disguising a malicious file as a legitimate file to avoid raising suspicion. Evasion may involve disguising a malicious file as a legitimate file in order to avoid suspicion. Collection can be the act of gathering information to prepare for a subsequent action, such as sniffing a network

through a man-in-the-middle. Command and Control could be Ingress Tool Transfer. This is when an attacker sends a tool or a file to be used in an attack from an external source, which can propagate a malicious file to the attack vector. Discovery can include the act of learning about a user. An attacker can obtain a valid account from the target's system or through compromised account information. Lateral Movement involves the propagation of an attack to other networks. An attack can move from the external network to the internal network by hijacking a session that allows the attacker to access the internal network from the external network.

Table 2 is an example of assets that can be used in a scenario causing damage to the PLC's control process and information on attack techniques/vulnerabilities accordingly. Note that not all assets are exploitable or vulnerable to any particular attack technique. For example, when attacking the Triconex safety instrumented system (SIS) developed by Schneider Electric, the malware used in Triton [3], an attack on an oil and gas facility, uses the SIS's application programming interfaces (APIs) to execute malicious shellcode. MITRE ATT&CK supports performing an attack by utilizing normal functions of a service or system through attack techniques such as native API. Otherwise, if a specific attack technique, such as firmware modification, is mapped to an asset, the CVE enabling the attack should also be assigned to the asset. By allocating vulnerability information to assets, MITRE ATT&CK's weakness can be supplemented, which does not provide detailed information on attack techniques. If multiple vulnerabilities are identified for one asset, to minimize the complexity of the scenario, the single most appropriate vulnerability can be selected by considering the attack purpose, attack target, and impact.

Table 2. An example of the attack techniques causing damage to the PLC's control process.

Asset	Attack Tactic	Attack Technique and Vulnerability
Internal network PC	Initial Access	Hardware additions
	Execution	Exploitation for client execution (CVE-2012-0158)
	Discovery	Remote system discovery
	Lateral Movement	Exploitation of remote services
EWS	Initial Access	Exploitation of remote services (CVE-2017-0144)
	Execution	Exploitation for client execution (CVE-2020-7315)
		API for Memory Allocation and Code Execution/Injection
		Program download (CVE-2015-7937)
	Lateral Movement	Program download (CVE-2015-7937)
PLC	Lateral Movement	Program download (CVE-2015-7937)
	Impair Process Control	Unauthorized command message

Table 3 is an example of an event message that can detect an attack technique used in a scenario causing damage to the PLC's control process. To use cyberattack scenarios in exercises, information enabling to identify or respond against cyberattacks should be included. For example, if an attacker attacks an ICS asset remotely over the network, events associated with the attack may be observed in the network packets. In this case, by monitoring the network traffic, the attack can be detected. To this end, we identify logs and event messages that can be observed by the attack techniques previously assigned to assets as information for recognizing and responding to cyberattacks. Since the proposed method utilizes MITRE ATT&CK, log or event messages necessary for detecting cyberattacks can be adopted through detection items provided by ATT&CK.

Table 3. An example event message for attack causing damage to the PLC's control process.

Attack Technique	Event Message
Hardware additions	Drive creation events
Exploitation of remote services	Network traffic log
Program download	Application log
Unauthorized command message	Process history log

3.3. Step 3: Modeling Cyberattack Scenarios

An attack tree is constructed based on the information analyzed for cyberattack scenarios in the attack scenario modeling step. At this step, the scenario developer assumes all attacks that can occur in the target system and expresses the paths to achieve the attacker's goal defined above as an attack tree. To this end, we augment and modify the standard attack tree to present the previously analyzed information.

Figure 2 is an example of an attack tree for an ICS; the nodes have a hierarchical structure. The attack tree recursively comprises several partial attack trees under a single root node. A partial attack tree is a part of the overall attack tree and comprises assets to be targeted for a specific attack goal and nodes containing attack information. By this definition, each node represents an attack procedure, which is a single step of the attack scenario. In the attack tree, the attack's ultimate target is located at the root node. In nodes below the root, techniques of MITRE ATT&CK for Enterprise/ICS required to achieve the attack's goal are expressed along with assets. There may be multiple ways to achieve a single goal, which is why there may be multiple partial attack trees under a single root node. The attack tree, starting from the leaf node to the root node, each node represents an attack that an attacker should carry out.

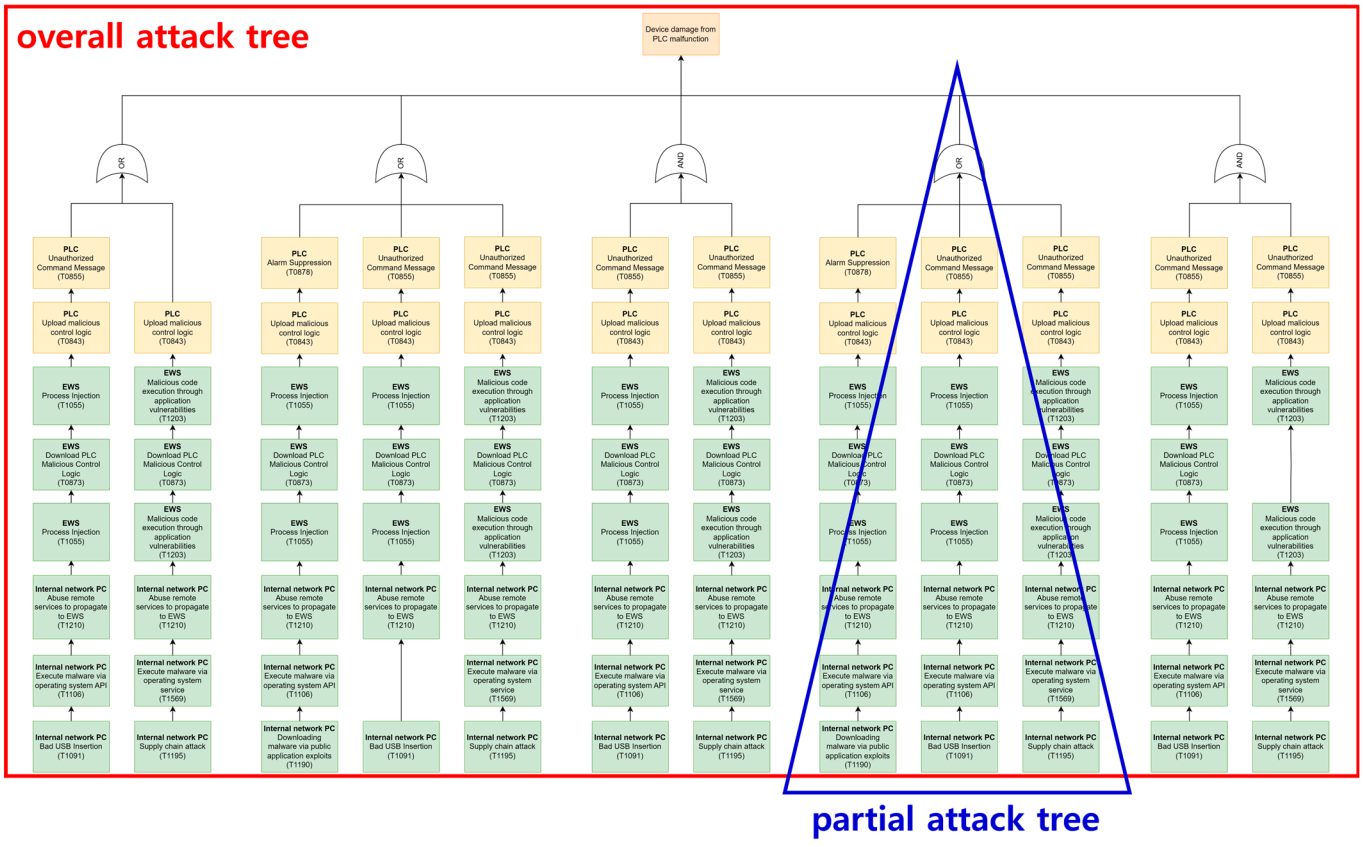


Figure 2. An example of the attack tree for cyberattack.

As shown in Figure 3, each node except the root node includes a target (information on the asset where the attack is conducted), action (malicious action on the asset), and attack information (attack technique, CVE vulnerability, CWE weaknesses, and mitigation) are specified.

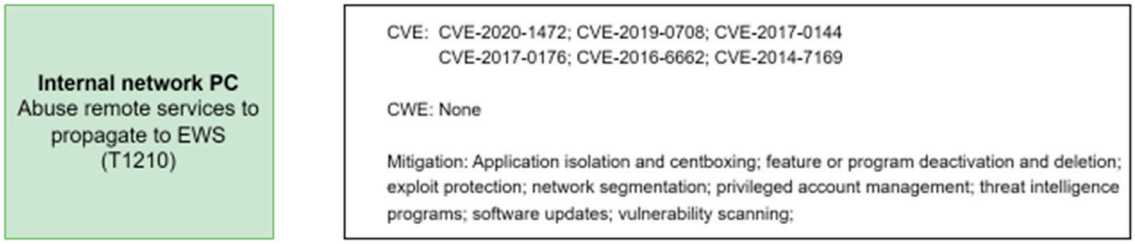


Figure 3. An example of the node in the attack tree.

The attack tree is constructed from the root node to the leaf node. We introduce the Purdue model to hierarchically configure the attack tree according to the structure of ICS. Since the attacker's ultimate target is located at the root node, we take devices at a lower level (close to level 0) as the root node in the Purdue model. Then, according to the Purdue model, the attack tree is formed by sequentially arranging the upper layer devices. Table 4 shows the devices according to the hierarchy of the Purdue model.

Table 4. Examples of hierarchical levels and their devices according to Purdue model.

Purdue Model Level	Component Example
Level 0	Field device (Sensors, Actuators, Robots, etc.)
Level 1	PLC
Level 2	HMI
Level 3	EWS, Internal network PC

For example, assume the following attack sequence: PC in internal network -> EWS -> PLC -> field device. For this scenario, the attack tree is layered from level 0 to level 3 as follows, according to the Purdue model. Also, if the initial attack starts from the HMI, the attack tree is composed only from level 0 to level 2.

Progressing from the leaf node to the root node in the attack tree means the cyberattack proceeds from the initial access to the attack target. When an attack moves from a lower node to an upper node, there may be optional attack techniques to improve or enhance the attack's success rate. We introduce three operators to systematically express these attack techniques in the attack tree (see Table 5). The enter operator makes an essential attack technique to enter an attack target (asset) in a partial attack tree and is represented as a leaf node of the partial attack tree. The Subsequent operator lists subsequent attack techniques that should be performed to achieve the attack goal after the attacker enters the attack asset. The Subsequent operator places these attack techniques in order between the leaf node and the upper node of the partial attack tree. Lastly, the select operator does not have to be performed, but attack techniques that can be selectively performed to increase the success rate or advance the attack are listed. By these operators, several nodes can be located between the leaf node and the root node of the subtree. We call a path from any single leaf node to the root node as an attack sequence (see Figure 4). An attack sequence is necessarily generated by at least one entry operator and subsequent operators.

Table 5. Operators for constructing attack sequence.

Operator	Description
Enter	A node at the end of a partial attack tree that represents an attack vector for entering the target asset.
Subsequent	Nodes containing techniques that may occur after entry and are essential to the attack scenario. Can consist of 1 to N nodes
Select	Nodes representing incidental techniques that may occur after entry and are not essential to the attack scenario. Can consist of 0 to N nodes

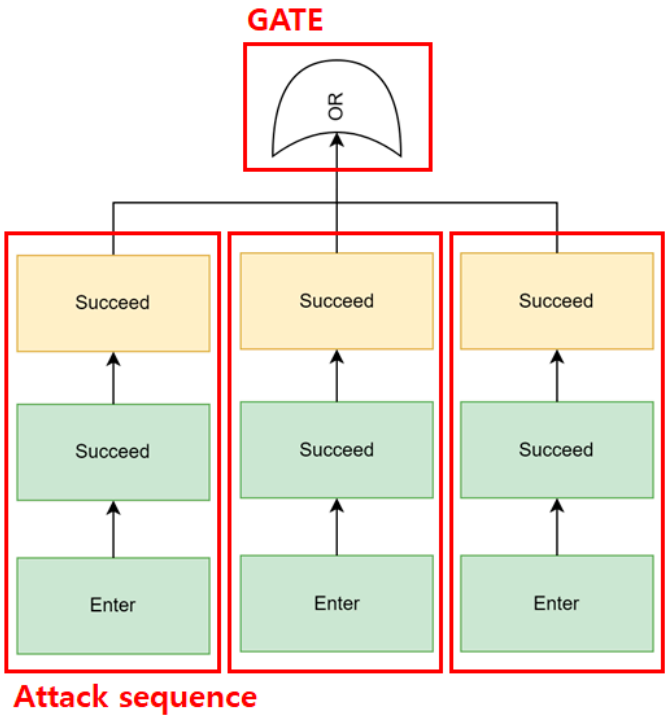


Figure 4. An example of the partial attack tree.

The SAND Gate model is used to reflect the flow of the scenario. The gate representation used by the Sand Gate model is shown in Table 6. If all lower attack sequences must succeed to achieve the attack goal, connect them with an AND gate. If the attack goal can be achieved by the success of only some of the attack sequences among many attack sequences, connect them with an OR gate. If there is a sequence to the attack sequence that requires all lower attack sequences to succeed, it can be expressed through a SAND gate.

Table 6. Gates Graphic representation.

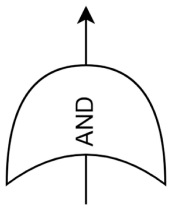
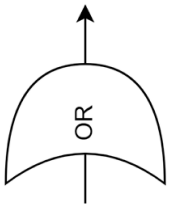
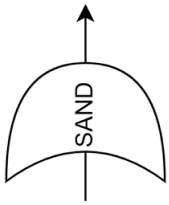
Gate	Description
	Conjunctive gate (AND gate)

Table 6. *Cont.*

Gate	Description
	Disjunctive gate (OR gate)
	Sequentially conjunctive gate (SAND gate)

3.4. Step 4: Quantifying Cyberattack Scenarios

The attack scenario quantification step determines which scenario is more appropriate for exercises with the CVE information, CVSS score, and DREAD. Various attack scenarios can be derived through the attack tree. Since each scenario has different characteristics, a criterion is needed to select the most suitable scenario for exercises. In this step, we evaluate the vulnerabilities assigned to the nodes in the attack tree, quantify each node (attack procedure) with a score, and then derive an attack scenario based on these scores.

To quantify the attack procedure, nodes are evaluated by scoring for categories (damage potential, reproducibility, exploitability, affected users, and discoverability) of DREAD using CVSS, which means the severity of CVE vulnerabilities allocated to nodes. In order to calculate the score through the CVSS and DREAD, we utilize the CVE vulnerability information assigned to each node of the attack tree in the previous step. However, since it is difficult to identify CVE vulnerabilities related to some MITRE ATT&CK's techniques (i.e., native APIs), scenario developers can calculate DREAD scores based on their expertise or may not be included in the score calculation.

Damage potential. DREAD's damage potential represents the expected damage to a system or asset if a particular vulnerability is successfully exploited. We adopt the impact score within the base scores of CVSS to score this category. The impact metric of CVSS represents the impact of a vulnerability on a system and comprises three subcategories: confidentiality, integrity, and availability impact. The CVSS scores for the given vulnerability are found in the NVD database. We can adopt the impact score within the CVSS scores provided by this database. If the NVD database does not provide scores for the vulnerability, the scenario developer can evaluate the impact of vulnerabilities on the confidentiality, integrity, and availability of assets. We record the level of damage potential assessed against the vulnerability in the form shown in Table 7.

Table 7. A scoring format for evaluating the damage potential of the CVE vulnerability.

DREAD	CVSS	Scoring
Damage	potential	Impact metric

Reproducibility. DREAD's reproducibility is evaluated by considerations such as the effort, time, and technical resources required for an attacker to exploit the vulnerability. If a vulnerability can be easily reproduced, it has a high score. Conversely, reproducibility would be low if many resources and expertise were required to exploit a vulnerability. We adopt the attack complexity as the CVSS item corresponding to DREAD's reproducibility. In CVSS, the attack complexity is a binary evaluating metric: low and high. Reproducibility

has an inverse evaluation of attack complexity, as the lower the complexity of an attack, the more difficult it is to reproduce it. If the vulnerability is difficult to reproduce, it is qualitatively rated as 'low' and quantitatively assigned a score of 5. On the other hand, if the vulnerability is easily reproducible, it is evaluated as 'high' and quantitatively given a score of 10. Quantitative scores may vary depending on the scenario developer's view. Scenario developers can utilize Table 8 to evaluate the vulnerabilities' reproducibility.

Table 8. A scoring format for evaluating the reproducibility of the CVE vulnerability.

DREAD	CVSS	Scoring	
Reproducibility	Attack complexity	Low(10)	High(5)

Exploitability. The exploitability evaluates how easily a vulnerability can be exploited. The high exploitable vulnerabilities do not require advanced skills or resources. On the other hand, vulnerabilities requiring conditions such as expertise or tools have low exploitability. We use the CVSS's exploitability metric to evaluate the exploitability of vulnerabilities. CVSS's exploitability is evaluated by integrating the attack vector, attack complexity, privileges required, and user interaction. The attack vector describes the type of access required to carry out an attack, and a high score is given when the attacker belongs to a highly accessible network. Attack complexity refers to the complexity of exploiting a vulnerability. The complexity can be higher if certain conditions are satisfied to conduct an attack, implying that the vulnerability is difficult to exploit. The privileges required indicate the level of privilege required to exploit the vulnerability. The user interaction means whether user interaction is required for the attack to succeed. Table 9 can be used to evaluate these factors. We adopt the average of the scores of all factors as the exploitability score of DREAD.

Table 9. A scoring format for evaluating the exploitability of the CVE vulnerability.

DREAD	CVSS	Scoring			
Exploitability	Attack vector	Network (10)	Adjacent network (6)	Local (4)	Physical (2)
	Attack complexity	Low (10)		High (5)	
	Privileges required	None (10)	Low (6)	High (2)	
	User interaction	None (10)		Required (5)	
Exploitability score	(Attack vector + Attack complexity + Privileges required + User interaction)/4				

Affected users. DREAD's affected users indicate how many users the vulnerability could affect. This category assesses the range or proportion of users that could be affected within a system, network, or application if a vulnerability were exploited. CVSS evaluates how a vulnerability can affect one system to another through a metric called scope. However, instead of using the rates for the scope provided by CVSS, we slightly modified the scoring system to fit the concept of affected users in DREAD. We divided the targets affected by vulnerabilities into internal components and external components. If the vulnerability affects only the internal components of the ICS, a score of 5 is given. Otherwise, if the vulnerability also affects external components along with internal components, a score of 10 is given. The scenario developer can utilize Table 10 to evaluate affected users.

Table 10. A scoring format for the affected user of the CVE vulnerability.

DREAD	CVSS	Scoring	
Affected users	Scope	Internal factor (5)	External factor (10)

Discoverability. DREAD's discoverability measures how easy it is to find vulnerabilities. This category considers the time, effort, and skill required for an attacker to discover and understand a particular vulnerability. In this context, we define the surface through which an asset is exposed as the discoverability of a vulnerability. Accordingly, we evaluate the discoverability of vulnerabilities by utilizing the attack vector of CVSS. If exploitation requires physical access to an asset, discoverability is rated low. On the other hand, if an attacker can access an asset through a public network such as the Internet, the vulnerability is easily discovered. The scenario developer can evaluate the discoverability of vulnerabilities with Table 11.

Table 11. A scoring format for the discoverability of the CVE vulnerability.

DREAD	**CVSS**	**Scoring**			
Discoverability	Attack Vector	Network (10)	Adjacent network (6)	Local (4)	Physical (2)

Once each category of DREAD is scored for the vulnerability assigned to each node of the attack tree, we calculate the average of these scores and assign it to the corresponding node. Table 12 shows an example of score calculation by applying DREAD to attack techniques and vulnerabilities of MITRE ATT&CK that can be assigned to ICS assets.

Table 12. Examples for evaluating the vulnerabilities through DREAD.

Attack Technique and Vulnerability	**D**	**R**	**E**	**A**	**D**	**Tot**	**Avg.**
Supply Chain Attack (T1569) (CVE-2023-23397)	5.9	10	10	5	10	40.9	8.18
Exploitation of Remote Services (T1210) (CVE-2020-1472)	6	10	10	10	10	46	9.2
Exploit Public-Facing Application (T1190) (CVE-2020-0688)	5.9	10	9	5	10	39.9	7.98
Process Injection (T1055) (CVE-2020-7315)	5.9	10	6	5	4	30.9	6.18
Replication Through Removable Media (T1091) (CVE-2010-2568)	10	7	9	10	10	46	9.2
Program Download (T843) (CVE-2015-7937)	10	10	10	10	5	45	9.0
Exploitation for Client Execution (T1203) (CVE-2018-4878)	5.9	10	10	5	10	40.9	8.18
Manipulation of View (T0832) (CVE-2020-0688)	5.9	10	9	5	10	39.9	7.98

Nevertheless, the methodology of evaluating scenarios in such a manner presents a notable issue. Specifically, scenarios with a fewer number of child nodes inherently receive higher scores. Consequently, this system may inadvertently underrepresent the severity of a scenario that, despite embodying a critical attack, comprises a substantial number of child nodes, thus failing to achieve a commensurate score. To ameliorate this discrepancy, it is imperative to conduct a normalization procedure Normalized Scenario Score $= \frac{\text{Scenario Score}}{\text{Number of Scenario Nodes}}$ Scenario Score divided by Number of Scenario Nodes prior to finalizing the scenario's score, which serves to moderate the score disparity between scenarios with divergent numbers of child nodes.

3.5. Step 5: Generating Cyberattack Scenarios

In this step, an attack scenario is generated from the attack tree in the previous step. In this section, we describe the process for the attack scenario generation by taking the EWS-PLC attack as an example.

The genesis of scenario generation commences with the construction of an attack tree targeting the field device. This is exemplified by the orchestration of Malicious Behavior on the PLC, as illustrated in Figure 5. Alternatively, an attack tree may be formulated to depict the modification of the PLC control logic as a consequence of Malicious Behavior facilitated by the execution of malware.

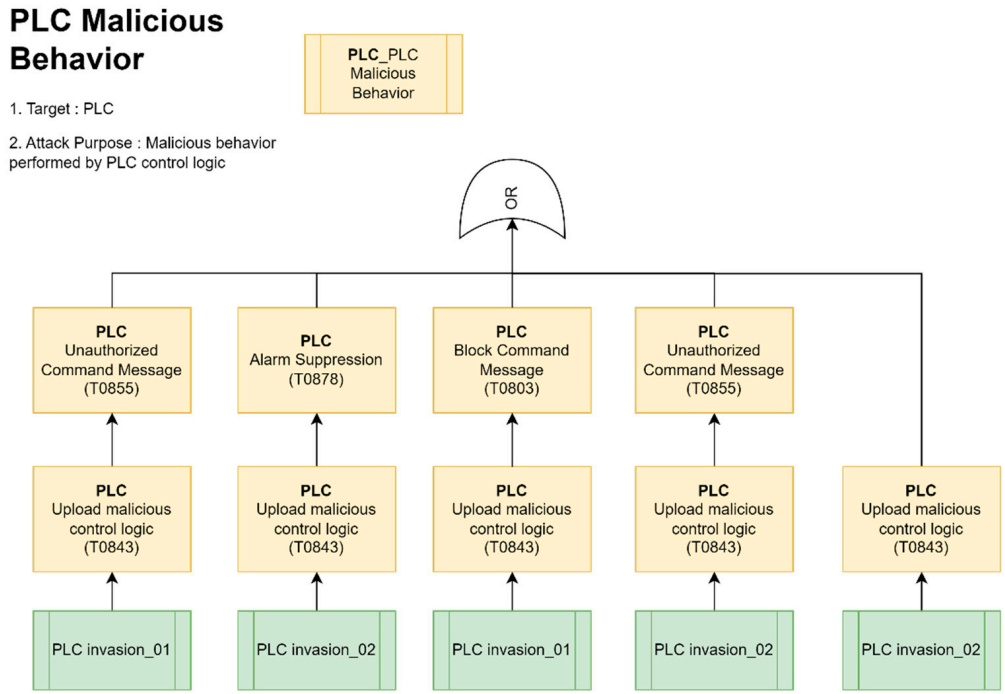


Figure 5. Example attack tree from PLC to field devices.

Figure 6 illustrates the construction of an attack tree, which represents the invasion of a PLC from the EWS. Within the example tree, nodes are designated to signify the execution of Malicious Behavior, encompassing the deployment of nefarious scripts or the manipulation of application vulnerabilities to initiate malware within the EWS. The nodes associated with the EWS are meticulously chosen to compile an attack tree that enables the transmission of malicious control logic to the PLC, thereby posing a potential threat to the integrity of the field device.

Figure 7 presents an example of an attack tree delineating the invasion of the EWS from an internal network PC. Nodes within the tree are selected to represent actions capable of executing malicious code on the EWS and include nodes that facilitate the propagation of attacks from the internal network to the EWS, such as the exploitation of remote services. The nodes situated at the lowest tier of the tree denote the initial commencement of the assault and correspond to Techniques categorized under 'Initial Access' within the MITRE ATT&CK framework.

Utilizing the established attack trees, it is possible to generate a comprehensive attack tree as exemplified in Figure 8. Scores have been assigned exclusively to nodes associated with a corresponding CVE identifier. The DREAD score for each node was allocated by referencing the example provided in Step 4. Subsequent to the summation of the assigned scores, the aggregate is divided by the number of nodes present in the tree to determine the final score for the scenario. In this instance, it is discernible through the scoring that the scenario positioned furthest to the right represents the most favorable outcome.

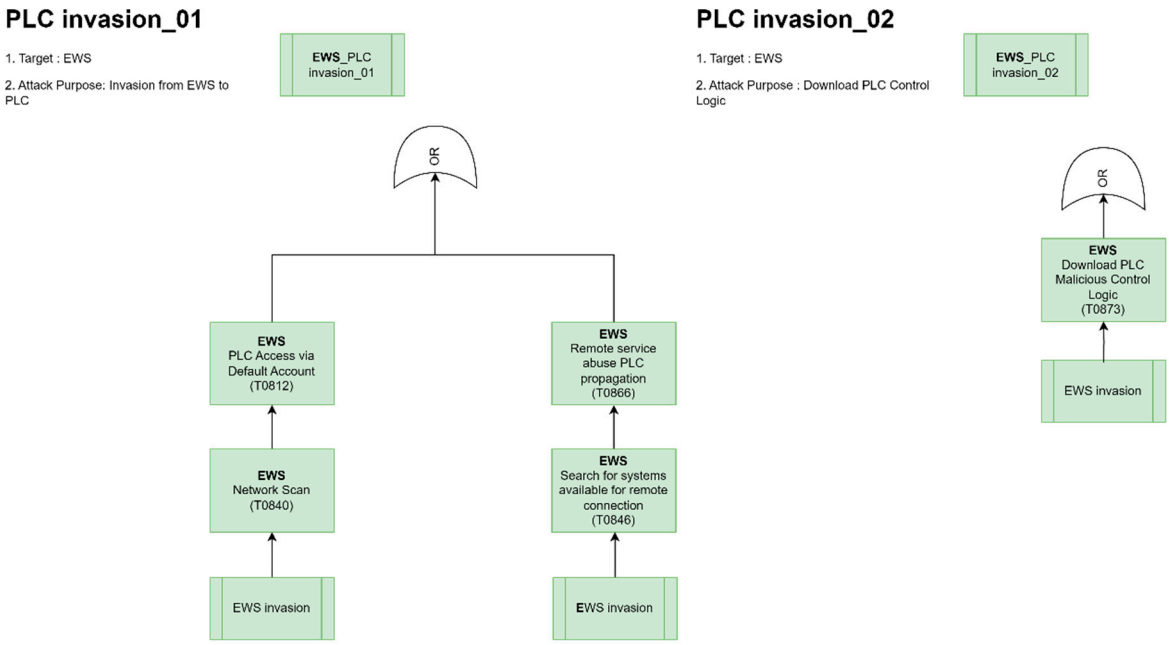


Figure 6. Example PLC invasion attack tree.

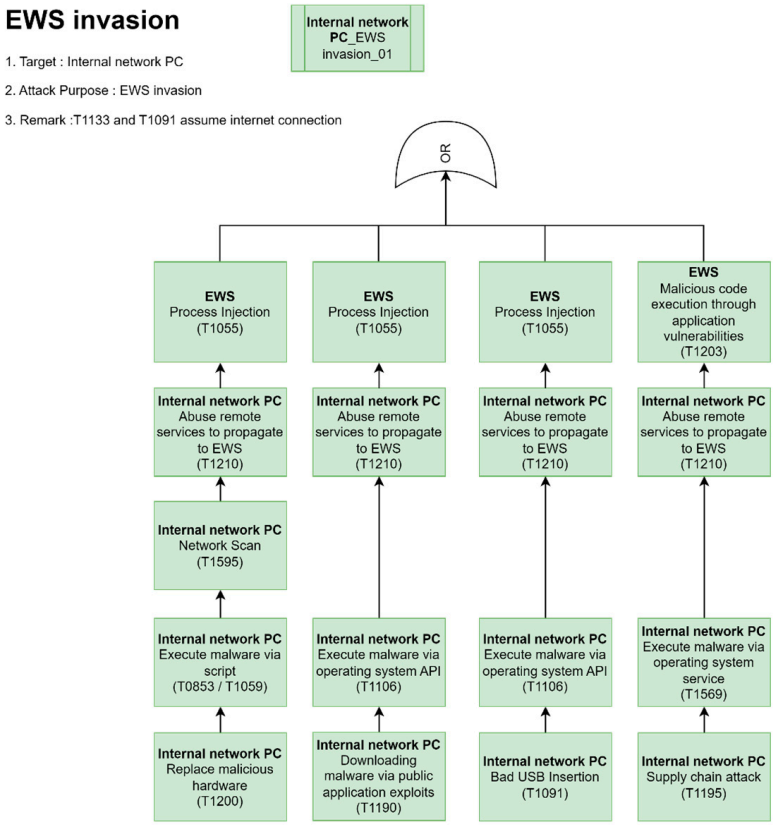


Figure 7. Example EWS invasion attack tree.

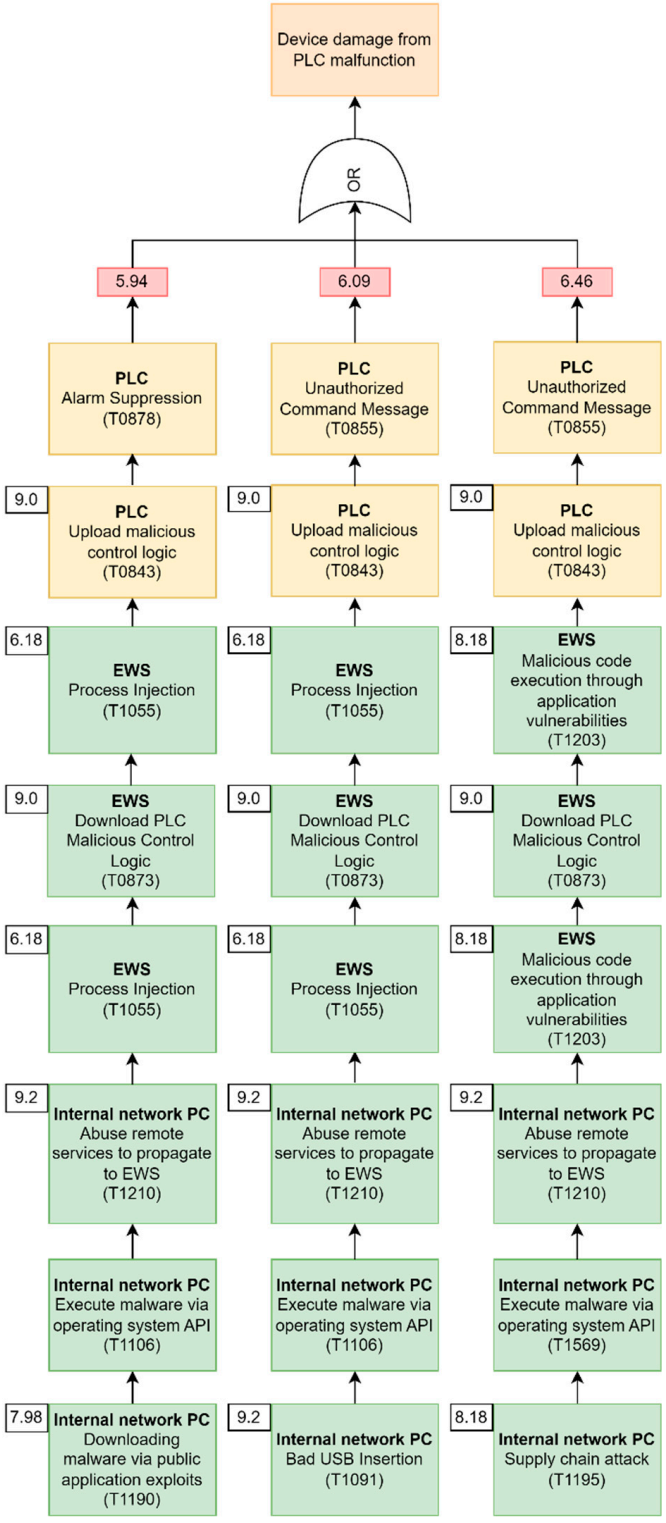


Figure 8. Example of Attack Scenario attack tree.

4. Evaluation

In this section, we evaluate GENICS, an attack scenario generation framework for cybersecurity exercises proposed in this paper. The evaluation is carried out in two directions. A qualitative study compares the process of deriving training scenarios with existing studies. The case study evaluates whether the steps of a cyberattack conducted against an actual ICS facility can be derived as an attack scenario. We employ the Triton attack case conducted against Saudi Aramco as a case study.

4.1. Qualitative Study

The proposed method was compared with the existing scenario generation methods. We established the following three considerations for comparison between scenario generation methodologies. The scenario generation methodologies compared in this section are studies related to the 'generation' of threat models such as Attack Sequence, Graph, Path, and Tree that we analyzed, and we selected studies that considered one or more of the following: realism of the scenario, formalization of the generation, and training effectiveness. First, the realism of the scenario for the exercise must be considered. Exercises are conducted to prepare for possible future cyberattacks in a real-world environment. Therefore, trainees can train more effectively when they train under realistic conditions. In HSEEP for Security Training and Evaluation, modeling and simulation are used for realism to replicate various attacks in a realistic manner. Therefore, the trainees can train more effectively when they train under realistic conditions. Second, the attack scenario generation process should be systematic: the exercise planner should generate attack scenarios systematically without missing steps, and the scenarios should be generated systematically using the same method for a fair evaluation. For example, Masaki Inokuchi et al. [35] present a number of rules to establish a systematic procedure for generating an attack graph. Third, ensure the effectiveness of exercises utilizing attack scenarios. NIST 800-61 [36] states that Phase 1 (Preparation) of the incident response methodology includes training the incident response team. The training should be effective in demonstrating the use of tools and familiarization with the response process for responding to cyber incidents. Evaluate existing research and whether it includes methods for quantitative evaluation to ensure training effectiveness. From the above considerations, we prepared seven evaluation items for the proposal.

- Generation scenarios targeting ICS: Evaluating whether the method can model general scenarios for various ICS facilities.
- Identifying assets: Evaluating whether critical assets of the facility are identified during scenario modeling.
- The basis for attack techniques: Evaluating whether the attack techniques appearing in the scenario were presented using a known framework such as MITRE ATT&CK
- The basis for Vulnerability: Evaluating whether the vulnerability information presented in the scenario was delivered using a knowledge base such as CVE.
- Scenario modeling for each asset: Evaluating whether the attack technique that is performed from the asset where the attack starts to the final asset is presented for each asset.
- Quantification: Evaluating whether one can prioritize the identified scenarios

Table 13 shows the results of comparing the methods for generating cyberattack scenarios. Utilize a straightforward notation system to evaluate the capabilities of the methodologies in relation to the listed criteria. An 'O' mark indicates that the methodology can effectively provide the function associated with the respective criterion, whereas an 'X' denotes that providing the function is challenging or not feasible within the context of that methodology. The seven criteria are a selection of the conditions required for effective scenario building and training in an ICS environment. Serru et al. [13] generated an attack sequence targeting the ship control system. This study identified in-ship assets such as shore control center (SCC) and global positioning system (GPS), and attack techniques constituting attack scenarios are based on MITRE ATT&CK's techniques. The vulnerabilities are not used in the attack scenario, and multiple attacks are modeled for each component and link located in the attack vector. Also, no quantification of scenarios is carried out. Ö. S. Ferda et al. [16] suggested a framework dynamic generating an attack graph, which is not intended for ICS. In this study, assets located in the Internet zone and demilitarized zone were identified. CWE was used for attack techniques, and CVE was used for vulnerabilities; threat modeling and scenario quantification for each asset in the attack path were not conducted. A. Dutta et al. [29] generated an attack sequence targeting the electric power grid. To generate the attack sequence, attack intent, and targets were modeled

through electric sector high-risk failure scenarios (EHFS), and assets included in the power grid were identified. Based on the attack information of the scenario, an attack tree was constructed through CPAEC, and CWE and CVE were mapped to it. This study quantified and prioritized the success rate of CWE. G. Kavallieratos and S. Katsikas [32] generated an attack path targeting a CPS. This study identified CPS assets and generated an attack path for the identified assets. However, attack techniques and vulnerabilities were not provided. The attack path's risk and asset's importance were calculated through DREAD.

Table 13. Comparison between the methods for generating cyberattack scenarios.

	[13]	[16]	[29]	[32]	Proposal Scheme
Generation scenarios targeting ICS	O	X	O	O	O
Identifying assets	O	O	O	O	O
The basis for attack techniques	O	O	O	X	O
The basis for Vulnerability	X	O	O	X	O
Scenario modeling for each asset	O	X	O	X	O
Quantification	X	X	O	O	O

On the other hand, GENICS aims to generate realistic attack scenarios that can occur in an ICS environment. To do this, we systematically collect assets and assign attack information to each asset. This information is used to build the attack tree. We quantify each node configuring the attack tree and derive a scenario based on it.

4.2. Case Study: Triton

In this section, we conduct a case study using Triton [3], a cyberattack conducted against a Saudi Aramco facility. The Triton attack case was selected as a case study due to its pivotal role in exposing significant vulnerabilities and risks within industrial control systems. Notably, the Triton attack employed novel techniques to compromise equipment that was previously considered secure. The inclusion of this case study in our research is deliberate; it serves to demonstrate the efficacy of our proposed methodology in detecting and responding to unexpected attacks. We confirm that Triton's attack scenario is reproduced through GENICS. This, in other words, means that GENICS can generate realistic cyberattack scenarios for a given ICS environment. In addition, by adopting realistic scenarios in cyber incident response exercises, training participants' experiences can be enhanced.

Preparation. In the case of the Triton case, the attack is conducted against the SIS of the Aramco facility. In this case, the main assets targeted for attack are the EWS, HMI, and the Triconex Tricon [37].

Threat analysis. We identify Triton's threats as:

- Goal: Causing economic damage due to facility shutdown.
- Target: Triconex 3008.
- Impact: Facility shutdown, safety control system disablement, SIS shutdown.

Attack information identification. Attack information identification in GENICS is performed through MITRE ATT&CK. Table 14 shows Triton's attack techniques disclosed in MITRE ATT&CK. In MITRE ATT&CK, 18 attack techniques exploited by Triton to shut down the SIS are identified. This includes techniques to modify the control logic and download malicious code to the SIS to perform adversarial actions. In addition, Triton's malware includes a function to identify control devices, propagate itself, change file names, and disable integrity checks to bypass the detection. Building upon the identified Triton attack techniques, the GENICS framework amalgamates these with other tactical methods to delineate a comprehensive set of attack techniques for the development of training scenarios in cybersecurity exercises.

Modeling Cyberattack Scenarios. GENICS models attack scenarios through the attack tree. The attack tree is constructed based on the attack information identified through

MITRE ATT&CK. Figure 9 shows the result of attack scenario modeling based on the attack information. The initial access techniques utilized by Triton are not identifiable in the current version of the MITRE ATT&CK framework. We address this gap by constructing attack trees using alternative initial access techniques found within MITRE ATT&CK. While each attack tree incorporates different techniques, they ultimately converge on the objective of launching an attack against the SIS and inducing a shutdown.

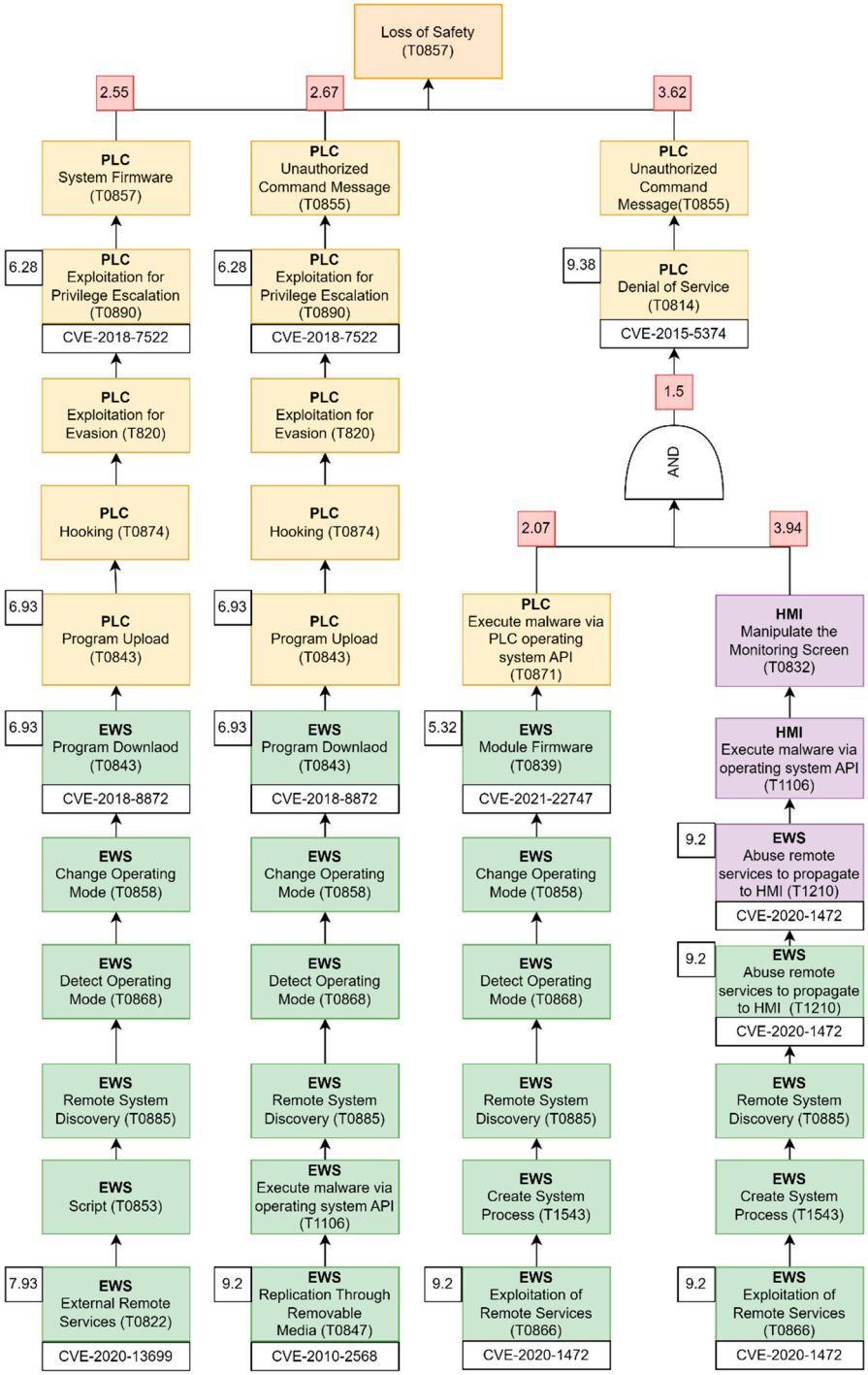


Figure 9. Attack tree for SIS Shutdown Scenario.

Table 14. Quantification of Triton's Attack Techniques as Disclosed in MITRE ATT&CK.

Attack Technique and Vulnerability	D	R	E	A	D	Tot	Avg.
Exploitation for Privilege Escalation (T0890) (CVE-2018-7522)	5.9	10	6.5	5	4	31.4	6.28
Program Download (T0843) (CVE-2018-8872)	5.9	5	8.75	5	10	34.65	6.93
Module Firmware (T0839) (CVE-2021-22747)	3.6	10	6	5	2	26.6	5.32
External Remote Services (T0822) (CVE-2020-13699)	5.9	10	8.75	5	10	39.65	7.93
Replication Through Removable Media (T0847) (CVE-2010-2568)	10	7	9	10	10	46	9.2
Exploitation of Remote Services (T0866) (CVE-2020-1472)	6	10	10	10	10	46	9.2
Denial of Service (T0814) (CVE-2015-5374)	6.9	10	10	10	10	46.9	9.38

5. Limitation

The efficiency of the proposed framework for generating attack scenarios is fundamentally contingent upon the comprehensive identification of vulnerabilities within the attack scenario modeling of each asset. This identification process is crucial, as it forms the bedrock for evaluating the threat landscape and establishing appropriate defense strategies. However, challenges can arise due to the complexity of assets, which may hinder the consistent application of vulnerability assessments across various systems.

In the domain of ICS, the use of specialized system components is prevalent. The characteristic utilization of proprietary equipment within ICS can complicate the vulnerability identification necessary for effective threat modeling.

The construction of attack scenarios depends on the expertise and discretion of the developers. While this dependence enables the customization of scenarios to fit a range of operational needs, it introduces a subjective element into the scenario development process. This subjectivity can influence the effectiveness and consistency of cybersecurity training exercises derived from these scenarios. The reliance of the framework on the developers' expertise highlights the importance of in-depth field knowledge, which can impact the framework's application across various scenario constructions and training implementations.

6. Conclusions

Appropriate security controls should be established to respond to cyberattacks in the ICS environment. Due to the nature of the ICS environment, it is challenging to check security controls, so the current security level of the facility should be confirmed through cybersecurity exercises. Realistic attack scenarios against the ICS environment should be assumed for these cybersecurity exercises to be conducted properly. However, currently, there is a lack of research to generate highly reliable attack scenarios. In this paper, we propose GENICS, a framework generating attack scenarios for practical cybersecurity exercises in an ICS environment. GENICS leverages well-known knowledge bases such as MITRE ATT&CK and CVE to derive attack scenarios against ICS facilities. In addition, GENICS systematically expresses the attack procedures for ICS facilities through an attack tree and quantifies each procedure. This supports the reliable attack scenario for each asset on the path from the entry point to the attack target in the ICS environment. This can ensure training effectiveness and contribute to an improved participant experience. However, GENICS is limited in that vulnerabilities should be used to quantify derived scenarios.

In future work, we intend to overcome this limitation and research the automation of generating exercise attack scenarios.

Author Contributions: Conceptualization, I.S. and S.J.; methodology, I.S., S.J. and D.K.; validation, D.K. and M.G.L.; writing—original draft preparation, I.S., S.J., D.K. and M.G.L.; writing—review and editing, S.J.; visualization, I.S. and D.K.; supervision, J.T.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Nuclear Safety Research Program through the Korea Foundation of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea (No. 2106058, 40%). This work was partly supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government (MOTIE) (20224B10100140, 40%) and this work was supported by the Gachon University research fund of 2023 (GCU-202308140001, 20%).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. *ISO/IEC 27001:2022*; Information Security, Cybersecurity and Privacy Protection-Information Security Management System-Requirements. International Electrotechnical Commission (IEC): Geneva, Switzerland, 2022.
2. Farwell, J.P.; Rohozinski, R. Stuxnet and the future of cyber war. *Survival* **2011**, *53*, 23–40. [CrossRef]
3. Di Pinto, A.; Dragoni, Y.; Carcano, A. TRITON: The first ICS cyber attack on safety instrument systems. *Proc. Black Hat USA* **2018**, *2018*, 1–26.
4. Kim, J.; Kim, K.; Jang, M. Cyber-physical battlefield platform for large-scale cybersecurity exercises. In Proceedings of the 2019 11th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 28 May–31 May 2019; pp. 1–19.
5. Ota, Y.; Aoyama, T.; Nyambayar, D.; Koshijima, I. Cyber incident exercise for safety protection in critical infrastructure. *Int. J. Saf. Secur. Eng.* **2018**, *8*, 246–257. [CrossRef]
6. Philpott, D.R.; Gantz, S.D. *FISMA and the Risk Management Framework: The New Practice of Federal Cyber Security*; Newnes: Oxford, UK, 2012.
7. Federal Government Cybersecurity Incident and Vulnerability Response Playbooks. Available online: www.cisa.gov/sites/default/files/publications/Federal_Government_Cybersecurity_Incident_and_Vulnerability_Response_Playbooks_508C.pdf (accessed on 12 May 2023).
8. MITRE. MITRE ATT&CK®. 2021. Available online: https://attack.mitre.org/ (accessed on 12 May 2021).
9. Schneier, B. Attack trees. *Dr. Dobb's J.* **1999**, *24*, 21–29.
10. Nagaraju, V.; Fiondella, L.; Wandji, T. A survey of fault and attack tree modeling and analysis for cyber risk management. In Proceedings of the 2017 IEEE International Symposium on Technologies for Homeland Security (HST), Waltham, MA, USA, 25–26 April 2017; pp. 1–6.
11. Common Vulnerability Scoring System Version 3.1: Specification Document. 2019. Available online: https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf (accessed on 12 May 2021).
12. Meier, J. *Improving Web Application Security: Threats and Countermeasures*; Microsoft Press: Redmond, WA, USA, 2003.
13. Serru, T.; Nguyen, N.; Batteux, M.; Rauzy, A. Modeling Cyberattack Propagation and Impacts on Cyber-Physical System Safety: An Experiment. *Electronics* **2022**, *12*, 77. [CrossRef]
14. Choi, S.; Yun, J.-H.; Min, B.-G. Probabilistic attack sequence generation and execution based on mitre att&ck for ics datasets. In Proceedings of the Cyber Security Experimentation and Test Workshop, Virtual, 9 August 2021; pp. 41–48.
15. Nguyen, S. Automated Attack Tree Generation and Evaluation: Systemization of Knowledge. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2020.
16. Sönmez, F.Ö.; Hankin, C.; Malacaria, P. Attack dynamics: An automatic attack graph generation framework based on system topology, CAPEC, CWE, and CVE databases. *Comput. Secur.* **2022**, *123*, 102938.
17. Takahashi, Y.; Shima, S.; Tanabe, R.; Yoshioka, K. {APTGen}: An Approach towards Generating Practical Dataset Labelled with Targeted Attack Sequences. In Proceedings of the 13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20), Virtual, 7–11 August 2020.
18. Alatwi, H.A.; Morisset, C. Threat Modeling for Machine Learning-Based Network Intrusion Detection Systems. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; pp. 4226–4235.

19. AL-Dahasi, A.E.M.; Saqib, B.N.A. Attack tree model for potential attacks against the scada system. In Proceedings of the 2019 27th Telecommunications Forum (TELFOR), Belgrade, Serbia, 26–27 November 2019; pp. 1–4.
20. Wang, J.; Phan, R.C.-W.; Whitley, J.N.; Parish, D.J. Augmented attack tree modeling of distributed denial of services and tree based attack detection method. In Proceedings of the 2010 10th IEEE International Conference on Computer and Information Technology, Bradford, UK, 29 June–1 July 2010; pp. 1009–1014.
21. Poolsapassit, N.; Ray, I. Investigating computer attacks using attack trees. In Proceedings of the Advances in Digital Forensics III: IFIP International Conference on Digital Forensics, National Centre for Forensic Science, Orlando, FL, USA, 28–31 January 2007; pp. 331–343.
22. Ray, I.; Poolsapassit, N. Using attack trees to identify malicious attacks from authorized insiders. In Proceedings of the Computer Security–ESORICS 2005: 10th European Symposium on Research in Computer Security, Milan, Italy, 12–14 September 2005; pp. 231–246.
23. Hui, W.; Shufen, L.; Xinjia, Z. An improved model of attack probability prediction system. *Wuhan Univ. J. Nat. Sci.* **2006**, *11*, 1498–1502. [CrossRef]
24. Kim, K.H.; Kim, K.; Kim, H.K. STRIDE-based threat modeling and DREAD evaluation for the distributed control system in the oil refinery. *ETRI J.* **2022**, *44*, 991–1003. [CrossRef]
25. Zhang, L.; Taal, A.; Cushing, R.; de Laat, C.; Grosso, P. A risk-level assessment system based on the STRIDE/DREAD model for digital data marketplaces. *Int. J. Inf. Secur.* **2021**, *21*, 509–525. [CrossRef]
26. Hemberg, E.; Kelly, J.; Shlapentokh-Rothman, M.; Reinstadler, B.; Xu, K.; Rutar, N.; O'Reilly, U.-M. Linking threat tactics, techniques, and patterns with defensive weaknesses, vulnerabilities and affected platform configurations for cyber hunting. *arXiv* **2020**, arXiv:2010.00533.
27. Kern, M.; Liu, B.; Betancourt, V.P.; Becker, J. Model-based Attack Tree Generation for Cybersecurity Risk-Assessments in Automotive. In Proceedings of the 2021 IEEE International Symposium on Systems Engineering (ISSE), Vienna, Austria, 13 September–13 October 2021; pp. 1–7.
28. Ibrahim, M.; Al-Hindawi, Q.; Elhafiz, R.; Alsheikh, A.; Alquq, O. Attack graph implementation and visualization for cyber physical systems. *Processes* **2019**, *8*, 12. [CrossRef]
29. Dutta, A.; Purohit, S.; Bhattacharya, A.; Bel, O. Cyber attack sequences generation for electric power grid. In Proceedings of the 2022 10th Workshop on Modelling and Simulation of Cyber-Physical Energy Systems (MSCPES), Milan, Italy, 3 May 2022; pp. 1–6.
30. Polatidis, N.; Pavlidis, M.; Mouratidis, H. Cyber-attack path discovery in a dynamic supply chain maritime risk management system. *Comput. Stand. Interfaces* **2018**, *56*, 74–82. [CrossRef]
31. Islam, S.; Papastergiou, S.; Kalogeraki, E.-M.; Kioskli, K. Cyberattack path generation and prioritisation for securing healthcare systems. *Appl. Sci.* **2022**, *12*, 4443. [CrossRef]
32. Kavallieratos, G.; Katsikas, S. Attack path analysis for cyber physical systems. In Proceedings of the Computer Security: ESORICS 2020 International Workshops, CyberICPS, SECPRE, and ADIoT, Guildford, UK, 14–18 September 2020; pp. 19–33.
33. Polatidis, N.; Pimenidis, E.; Pavlidis, M.; Papastergiou, S.; Mouratidis, H. From product recommendation to cyber-attack prediction: Generating attack graphs and predicting future attacks. *Evol. Syst.* **2020**, *11*, 479–490. [CrossRef]
34. Microsoft Security Bulletin MS10-073–Important. Available online: https://learn.microsoft.com/en-us/security-updates/securitybulletins/2010/ms10-073 (accessed on 5 November 2023).
35. Inokuchi, M.; Ohta, Y.; Kinoshita, S.; Yagyu, T.; Stan, O.; Bitton, R.; Elovici, Y.; Shabtai, A. Design procedure of knowledge base for practical attack graph generation. In Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, Auckland, New Zeland, 7–12 July 2019; pp. 594–601.
36. Cichonski, P.; Millar, T.; Grance, T.; Scarfone, K. *NIST Special Publication 800-61 Rev 2: Computer Security Incident Handling Guide*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2012.
37. Mekdad, Y.; Bernieri, G.; Conti, M.; Fergougui, A.E. A threat model method for ICS malware: The TRISIS case. In Proceedings of the 18th ACM International Conference on Computing Frontiers, Virtual, 11–13 May 2021; pp. 221–228.