Hindawi
Security and Communication Networks
Volume 2022, Article ID 3455647, 19 pages
https://doi.org/10.1155/2022/3455647

Research Article

Consider the Consequences: A Risk Assessment Approach for Industrial Control Systems

Aram Kim,[1] **Junhyoung Oh**,[2] **Kookheui Kwon**,[1] **and Kyungho Lee**[2]

[1]*Division of Cybersecurity, Korea Institute of Nuclear Non-proliferation and Control, Daejeon, Republic of Korea*
[2]*School of Cybersecurity, Korea University, Seoul, Republic of Korea*

Correspondence should be addressed to Kyungho Lee; kevinlee@korea.ac.kr

Received 19 November 2021; Revised 6 May 2022; Accepted 17 May 2022; Published 22 June 2022

Academic Editor: Ilsun You

The development of information and communication technologies extended the application of digitalized industrial control systems (ICSs) to critical infrastructure. With this circumstance, emerging sophisticated cyberattacks by adversaries, including nation-backed terrorists, target ICSs due to their strategic value that critical infrastructure can cause severe consequences to equipment, people, and the environment due to the cyberattacks on ICSs. Therefore, critical infrastructure owners should provide high assurance to those involved, such as neighboring residents and governments, that the facility is adequately protected against cyberattacks. The risk assessment that identifies, estimates, and prioritizes risks is vital to provide high assurance. This study proposes a framework for evaluating risks by quantifying the likelihood of cyber exploitation and the consequences of cyberattacks. The quantification of the likelihood of cyber exploitation is inspired by research on Bayesian attack graphs (BAGs), allowing probability evaluation that considers the causal relationship between ICSs and multistage attacks. For the cyberattack consequences quantification, we propose a methodology to evaluate how far an impact will spread and thus how many functions will be influenced when an ICS is exploited. The methodology is conducted by ICS experts identifying and listing functional dependencies and essential function goals among ICSs that they are already familiar with and do not require in-depth cybersecurity knowledge. Through experiments, we demonstrated how to apply our framework to assess the risks of the plant protection system, which is a safety-grade digital system used in nuclear power plants. The result shows that risk can be multidimensionally assessed than previous literature, such as discovering that components that were not considered important have high risk due to their functional connectivity.

1. Introduction

Over the last decade in critical infrastructure sectors, digital systems have been replacing traditional analog systems because of the former's numerous benefits, such as high-performance capability, reliability, convenient maintainability, and simple procedures when it comes to conducting changes [1]. Industrial control systems (ICSs) are often found in critical infrastructures such as nuclear power plants, chemical plants, electrical grids, transportation systems, or water distribution systems. Critical infrastructure means systems and assets vital to a nation, where the incapacity or destruction of such systems and assets will have a debilitating impact on general security, national economic security, national public health or safety, or any combination of the things [2]. The strategic value of critical infrastructure makes it the potential target of adversaries, including nation-backed terrorists. Therefore, the United States has, through a presidential directive, categorized critical infrastructures into energy, public health, water treatment systems, nuclear power facilities, finance, etc., and required the protection of each sector by the designated sector-specific agency [3]. ICSs generally encompass several types of digital systems, including supervisory control and data acquisition (SCADA), distributed control systems (DCS), or programmable logic controllers (PLC); they perform monitoring, processing, and control functions in critical infrastructure [4].

The migration of ICSs toward digital technologies has facilitated the development and deployment of complex and highly interconnected systems and has enabled remote

control and supervision of critical infrastructure. However, the security of digital systems remains a significant challenge [5, 6]. ICSs, inasmuch as they are using air-gapped or proprietary systems, have long been considered immune to the cyberattacks that have significantly impacted corporate information systems. Nevertheless, several studies have found that this confidence is to be misplaced. Like any other information system, ICSs face security threats [7]. Open-standard off-the-shelf systems, such as Ethernet or web technologies, allow hackers to take advantage of the control industry's ignorance [8]. The application of diverse device technologies has led to vulnerabilities to various malicious intrusions [9] and attacks that can overcome air gapping, such as was seen with Stuxnet [10] or the attack on SolarWinds' supply chain [11].

Indeed, several cyberattacks against ICSs have been reported. The so-called Slammer worm attacked a vulnerability of the buffer overflow of the Microsoft SQL Server 2000. It caused data overloads in the site network, slowing computer communications at the Davis-Besse NPP in 2003. As a result, the safety parameter display system became unavailable and remained useless for four and a half hours [12]. Stuxnet [10], which targeted Iranian nuclear centrifuges by exploiting vulnerabilities in Microsoft Windows, showed how cyberattacks could cause physical damage, with a nation-backed advanced persistent threat attack even possible. Recently, the Colonial Pipeline, which supplies fuel for roughly 45% of the east coast of the United States, suffered a cyberattack [13]. The ransomware outbreak in that facility system drove up gasoline prices and sparked shortages at filling stations after the company shut down the approximately 5,500-mile-long pipeline [14].

These cyberattacks against ICSs demonstrate that the incapacity or destruction of critical infrastructure can have debilitating impacts on our physical or economic security, public health, and safety [15]. In particular, cyberattacks on safety-related infrastructure can lead to death, injury, occupational illness, damage to or loss of equipment or property, and damage to the environment [16]. To prevent such a debilitating impact and maintain a secure state against cyberattacks on ICSs, risk management must be used to identify and mitigate risks to an acceptable level. In this context, risk assessment is critical to identify, estimate, and prioritize risks [17].

Among the risk assessment approaches reported in the literature, quantitative assessment has the advantage of providing rigorous, repeatable, and reproducible assessment results while reducing cost, expert time, and effort [17]. Moreover, quantitative assessment most effectively supports cost-benefit analysis of alternative risk responses or courses of action [17]. In particular, in a network environment in which multiple systems are interconnected, graphical security models such as Bayesian attack graphs (BAGs) [18–24] are suitable for risk assessment. BAGs provide powerful tools that represent information about causal relationships among vulnerabilities while compensating for the drawback of the attack graph (AG), which cannot provide information on vulnerability exploiting probabilities, which is an essential factor for risk assessment [21]. BAGs perform static and dynamic analyses to calculate the probabilities that an attacker can reach each state in the graph [25]. Moreover, a BAG can represent a multistage attack that exploits a sequence of vulnerabilities, providing a causal relationship between random variables and a formalism for reasoning about partial beliefs under uncertain conditions [22].

Previous BAG-related studies proposed automated methods for quantitative risk assessment. These studies reported on the automatic generation of BAGs from host vulnerabilities obtained through vulnerability scanners such as Nessus [26], Open VAS [27], and the National Vulnerability Database (NVD) [28], and from network topologies obtained through network discovery tools such as NMAP [29]. Concerning quantitative risk assessment, methods such as the common vulnerability scoring system (CVSS) [30] were proposed to quantify each type of vulnerability in a standardized way. Most quantitative BAG-based methods use the exploitability metrics of the CVSS to indicate the probability of successful exploitation of vulnerabilities [21]. For this reason, we also propose automated quantitative risk assessment using the exploitable metric of a CVSS based on BAGs.

When assessing the potential impact of ICS incidents on critical infrastructure, it is important to incorporate effects on physical processes/systems and impacts on dependent systems/processes, among other possibilities [4]. However, BAGs usually focus on the possible attack propagation path or its probability. Let us consider BAGs for two networks with the same configuration. The outputs will be the same (i.e., possible attack path or probability). However, if one of the networks performs safety-related functions for a nuclear power plant in unsafe conditions, whereas the other performs an auxiliary diagnostic function, the outputs should differ.

In this study, we propose a framework that evaluates risks by quantifying the likelihood of cyber exploitation and the consequences of cyberattacks. The proposed framework applies BAGs to quantify the likelihoods of vulnerability exploitation while considering causal relationships among systems. In view of BAGs' limitations in measuring consequences, our framework also adds a consequence layer that quantifies impacts based on functional dependencies to measure how far the impact propagates when a system is exploited, and the attacker controls functions. The reason to measure consequences considering the impact propagation is that if one device is exploited by a cyberattack, its failure can spread to other devices because, in ICSs, devices are related to each other when it comes to executing a process. Additionally, we applied the framework to assess the cybersecurity risk of a simplified safety-grade digital system used in nuclear power plants and demonstrate its efficacy.

The proposed risk assessment framework contributes to the state of the art as follows:

(1) It provides a functional dependency modeling approach in an environment in which devices are functionally dependent on each other and analyzes how the impact propagates when a specific device is

exploited. This is conducted by listing the functional dependencies in an ICS, its essential function goals, and its failure modes; all these aspects are well known by ICS experts.

(2) The framework minimizes the need for ICS experts and security experts to understand each other; ICS experts do not have to address security issues. ICS experts only need to identify and list functional dependencies. On the other hand, security experts only need to comprehend the network architecture and vulnerabilities of systems. Ensuring that personnel who understand both areas do not need to participate can reduce the costs and time required to perform risk assessments.

(3) It provides a sophisticated risk assessment framework that evaluates risks reflecting ICS characteristics that perform one process through the combination of multiple devices. The risk is calculated by quantifying the likelihood of exploitation and its consequences, considering the causal relationship between several devices.

The remainder of this study is organized as follows. The following section presents an overview of ICSs. Next, we review related studies. An overview of the proposed framework is then described, and the likelihood and consequence assessment approaches, which are components of the framework, are presented. In the experimental section, we apply the proposed framework as a test network for a plant protection system used in nuclear power plants. Finally, conclusions and future research plans are represented.

2. Overview of Industrial Control Systems

In this section, we review the overall structure of ICSs and their security issues. The structure of an ICS consists of a physical layer and a cyber layer. The physical layer comprises sensors that measure temperature, levels, or pressures from the physical environment and transmit the measured variables to the cyber-layer and actuators. The cyber-layer logic result is then applied to the physical environment using devices such as pumps, motors, and valves. The cyber-layer consists of hardware and software to monitor the signal input from the physical layer, process received data, and execute necessary controls through sequences of logic. The hardware of the cyber-layer encompasses controllers such as PLC, DCS, and SCADA, and traditional information technology (IT) devices such as computers and servers. A wide range of digital transmitters, indicators, protective relays, or recorders are furthermore included in the ICS [31].

Figure 1 depicts the overall architecture of an ICS compromising a cyber-layer and a physical layer. Controllers in the cyber-layer receive signals from sensors and transmit control signals to actuators using a series of complex equations and logic. With one controller, it is possible to process the input signals and control actuators; however, when the ICS needs to handle a complex process, several controllers can work together, and one controller can control actuators using the results of other controllers. The

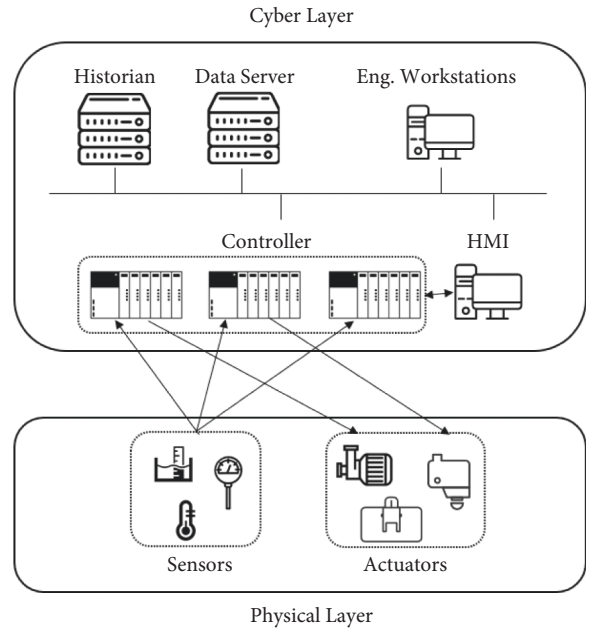

Figure 1: Example of ICS architecture.

controllers executing such complex processes do not control the actuator but only send their calculated data to the other controllers or servers. A human-machine interface (HMI) is mainly used by the ICS operator to monitor the facility status and to handle the logic by changing, if necessary, the process parameters, referred to as set points. The data server allows the data received from the controller to be shared with other controllers or with the HMI, and a historian is used to record process variables. Engineering workstations allow staff to perform necessary maintenance tasks for systems in the cyber-layer. In some cases, this layer may be connected to a corporate network or the internet.

From a security point of view, an ICS has unique features. In critical infrastructure, physical access to ICSs is generally strictly managed. Logical access to an ICS from outside is difficult because ICSs are not connected to the internet, and even purchasing an ICS is difficult when a facility uses proprietary systems. In particular, critical infrastructures that apply safety standards such as International Electrotechnical Commission (IEC) 61508 [32] can be considered to have a security advantage because they meet the safety requirements (e.g., single-failure criterion, completion of protective action, system integrity, and quality assurance independence) and are deemed safe after several safety analyses. However, ICSs also present several drawbacks in terms of security. Unlike IT environments, which have a lifetime on the order of three to five years, ICSs have a lifetime ranging from 15 to 20 years or longer [4], meaning that there is a possibility of vulnerabilities and difficulties acquiring when occurring security patches that are intended to mitigate vulnerabilities. ICSs also have difficulties applying host-based detection systems, widely used in IT, due to low performance or operational restrictions. In addition, it is not easy to determine the application of security patches for reasons such as continuity of facility operation or

functional safety. Furthermore, because critical infrastructure such as nuclear power plants has a strategic value for adversaries, they become the target of well-resourced (e.g., sufficient money, skills, and time) and experienced malicious cyber actors who can leverage various skills to undermine the trust model at every level [33]. In a world of increasing connectivity and cyber-threat innovation, we must assume that ICSs in critical infrastructure can be compromised, and no system can be certified as fully secure [33].

In IT systems, vulnerabilities and consequences are tightly coupled; therefore, an exploited vulnerability that impacts the so-called CIA triad (i.e., confidentiality, integrity, and availability) directly leads to consequences such as information disclosure, system disruption, and/or data alteration [34, 35]. However, concerning ICSs, penetrating the cyber-layer is not the same as failing the system [36]. The impact on the physical layer determines the consequence of a cyberattack on an ICS. Moreover, cyberattacks on ICSs have different consequences depending on which physical layer is affected by the attack. The heating, ventilation, and air conditioning (HVAC) system used in a nuclear power plant is taken as an example. Suppose that the attacker exploits vulnerabilities and turns off the HVAC system that provides cooling to the equipment room where instrumentation and control (I&C) systems are located. This can adversely affect the I&C systems in the equipment owing to the high temperature of the equipment room, which can, in turn, have consequences on ICS operability. However, it takes some time for the temperature to go up, and operators can detect it by monitoring the room temperature and restoring it through incident response procedures before the overall process becomes adversely affected. As another example, cases in which the main control room (MCR) HVAC system is exploited are considered. The MCR is a place where operators monitor and control power plants, and one of the functions of the MCR HVAC system is to maintain positive pressure so that radioactive materials do not enter the MCR when a radioactive accident occurs. Therefore, attacks on the MCR HVAC system seriously impact safety by preventing operators from ensuring their activities for safety responses in the MCR. Thus, even if identical systems are exploited, the consequences vary depending on the system function and related physical layer. To conduct a successful attack against an ICS, attackers need to acquire sufficient knowledge about the process behavior and ICS structure [37]. This equally applies to security managers. Thus, to effectively evaluate the risk, it is necessary to identify the consequences of attacks on all of the different systems of an ICS.

To accurately evaluate the consequences of cyberattacks on ICSs, it is necessary to consider how impacts cascade when a system is exploited. Impact cascading includes a system in the cyber-layer that affects the physical layer and a system influencing other systems in the cyber-layer. Wyman proposed an ICS impact model in which the exploitation of the CIA triad in a cyber-system triggers impacts cascading to other layers (e.g., physical, safety, business, and community). This model is depicted in Figure 2 [35]. Impact cascading due to system failure in the cyber-layer was analyzed through methodologies such as failure mode and effect analysis [38], fault tree analysis [39], and system-theoretic process analysis [40]. These previous approaches assumed that failure or a human error related to equipment constitutes the starting event of a hazard. However, cyberattacks can use the functionality of the trusted system to bypass safety barriers [33]. Therefore, it is necessary to consider impact propagation caused by the cyberattack from the cyber-layer to the physical layer. The framework we propose allows an expression of impact propagation from the cyber-layer to the physical layer by identifying functional goals based on CIA triads and involving ICS experts to evaluate weights.

The other factor in achieving accurate consequence assessment is evaluating relationships between systems in the cyber-layer. Systems in the cyber-layer of ICSs often rely on other systems to perform their logic. Controllers such as PLCs receive data from sensors and execute their logic, while a DCS or SCADA also receives information from PLCs scattered in various places and runs programs. In this association structure, if the integrity of a system is broken and corrupted values are transmitted, or the availability is exploited and does not transmit data to other systems, the dependent systems in the chain will also be affected. The example depicted in Figure 3 shows that a PLC that receives a temperature value from a temperature sensor and transmits it to another PLC is exploited. In this particular scenario, the attacker exploits the PLC's integrity, manipulating the memory of the PLC's input card and modifying the actual value from 350 ° C to 300 ° C. Then, the exploited PLC reads the false temperature from the input card memory and transmits it to the other PLC. In this case, because the logic of the PLC transmitting the false data is not changed, the receiving PLC is not straightforward to realize that there has been a modification of values, which operates the pump at 900 rpm instead of 1,000 rpm. In addition, the PLC transmits the corrupted temperature value to other monitoring systems so that operators will see the tempered value. At this time, the operator may also suffer adverse effects such as skipping a procedure to be executed or performing unnecessary control resulting from a wrong judgment by allowing the operator to observe false data. In the aforementioned example, there is only one exploited PLC, but the impact propagates to other PLCs and monitoring systems that are not exploited. If the risk assessment does not consider the function-failure impact propagation, devices other than the attacked PLC would be considered to normally operate in risk analysis. The framework we propose can model impact propagation from the cyber-layer to the physical layer by using ICS experts to enumerate functional dependencies. The identified information is then used to measure how the impact propagates when a system is exploited and how many essential functions are affected.

3. Related Work

In this section, we review previous studies on a security assessment. Several researchers aimed to evaluate and quantify the impacts of cyberattacks on the physical processes of critical infrastructure for risk assessment in ICSs [41]. By investigating changes in atomic behavior patterns of

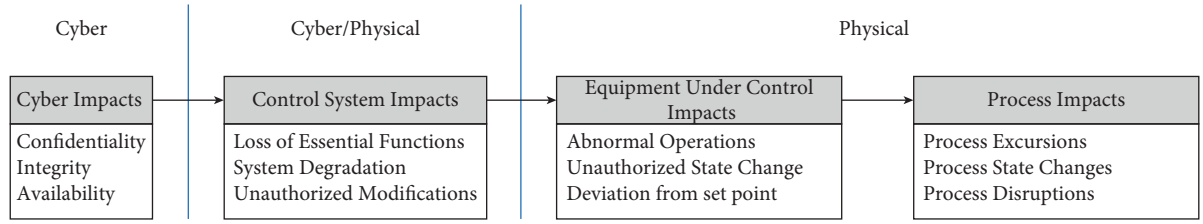

Figure 2: Impact cascade model [35].

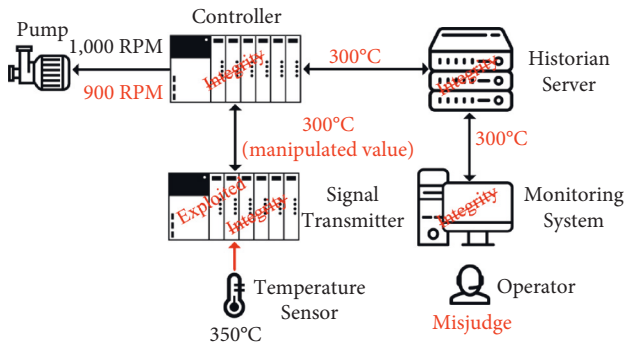

Figure 3: Example of cyber-event impact propagation.

a dynamic system, Ford proposed a formal analysis method based on system dynamics to identify loop dominance during a given time interval [42]. Loop dominance allows the feedback structure to identify the loop with dominant behavior according to changes in control variables. Thus, the identification of loop dominance was used in different studies for sensitivity analysis to quantify the impact of variable changes on security [37, 41, 43–45].

Extending Ford's model, Huang et al. presented an integrated sensitivity analysis method based on improving the understanding of dominant loops in the model [44]. Sensitivity analysis was used to quantitatively measure how sensitive a model is to change in its control variable. While Ford's model provides binary answers (yes or no) about loop dominance, Huang's sensitivity model computes variances between the reference model and the deactivated feedback loop to quantitatively measure the sensitivity.

Genge et al. proposed the cyberattack impact assessment (CAIA) methodology, which further refines Ford's and Huang's research and is adapted to address the complexity and ubiquitous nature of cyberattacks on critical infrastructure [41]. The CAIA compares the behavior of system processes in the presence and the absence of accidental or deliberate interventions to evaluate the significance of cyber-assets, thus making it possible to identify and rank assets in complex, large-scale, and heterogeneous critical infrastructure. This methodology is based on the calculation of cross covariances between observed variables before and after executing a particular intervention involving control variables, and is used to quantify the significance of control variables and measure the impact propagation of cyberattacks.

Orojloo and Azgomi proposed an approach to estimate impacts on ICSs and ranked the critical assets of the system in terms of their sensitivity to individual or combinations of attacks [37]. Haller and Genge presented a systematic methodology for designing intrusion detection systems by combining sensitivity analysis and cross-association assessment [43]. Variable cross-association makes it possible to identify groups of interventions that have the same impact on the group of observed variables, thereby reducing the number of variables to be monitored by determining a variable group that shows a sensitivity similar to that of the attack group.

Zhou et al. proposed a dynamic impact assessment approach that predicts trends in cybersecurity impact and considers the impact propagation analysis and quantification [45]. The presented approach abstracts assets with properties of construction, function, performance, location, and business, and models individual components (e.g., sensor, actuator, and controller) of ICSs utilizing Petri nets [46], a graphical and mathematical modeling tool, to quantify the impact of cyberattacks. After integrating each component model while considering interactions of assets, a system-level asset model is constructed, and common cyberattack models are also integrated into this model to derive the impact propagation of a cyberattack.

The above-discussed approaches based on system dynamics assist in analyzing and quantifying impact propagation and ranking critical assets in ICSs in cases of adversarial attacks. However, given that these studies evaluate consequences based on control or measurement variables, there is a limitation in terms of impact analysis of cyberattacks occurring in systems or components. It is evident that when utilizing the method suggested by Zhou et al. [45], consequence analysis for cyberattacks on a component or system is possible. However, modeling the internal behavior of each component requires detailed information about the device and logic. Thus, there is a limitation in modeling components in environments in which off-the-shelf equipment or commercial software is used. In contrast, the methodology proposed in this study can evaluate and quantify the impact of a cyberattack on a specific component by simply identifying the components necessary for its operation.

Another research area in security assessment is the so-called AG. An AG is capable of modeling attacks using data such as connectivity schemes, running services, or vulnerabilities of hosts and can show how an attacker can compromise a network. AGs have been used in several risk assessment studies [47–53]. Based on model checking techniques, Sheyner et al. [52] introduced a tool to automatically generate and analyze AGs. In this case, the AG

comprised nodes and edges reported in the literature; the nodes represented the host's states, whereas the edges denoted atomic attacks. The host's states were modeled as a set of Boolean variables, representing configurations and privileges, whereas the edges represented atomic attacks that transited the host's states. The security property was written in computation tree logic, and if the conditions of these properties were not met, the NuSMV model checker [54] generated all possible counterexamples.

To deal with the scalability problem, Ammann et al. have proposed a more compact AG model than that of Sheyner et al. [47]. Their model addresses the scalability problem [47] that occurs because the AG node represents the overall network state after each atomic attack, which causes the number of nodes to exponentially increase as the number of hosts and attacks increases. This type of scalability problem limits the applicability of AGs to small networks. A logical AG, which directly illustrates logical dependencies among attack goals and configuration information, was introduced by Ou et al. [51] to address the scalability problem. These authors provided a network security analyzer that utilizes logical programming based on the MulVAL tool [55]. The authors assumed that, compared to the exploitation dependency graph proposed in earlier studies, an AG using predicates and logical connectives to represent security correlations in a network is hardened and better suited for rigorous security analysis [47, 56].

Ingols et al. proposed the multiple prerequisite graph, which nearly linearly scales as the size of the network increases; they also proposed the network security planning architecture (NetSPA) tool [48]. The AGs in the literature are automatically generated by importing data, including the network topology, vulnerability information, and credentials. The authors suggested the reachability group concept, through which redundancies are collapsed using binary decision diagrams [57] to reduce complex, time-consuming tasks. Quantification of the likelihood of potential multistep attacks is also an objective pursued in the AG field.

Several other metrics have been proposed in the AG field, such as the percentage of compromised network assets [58], the minimal set of initial conditions that the weakest adversary should meet [59], the number of attack paths (shortest, average, and total) [60], and a suite that takes into account CVSS and topological aspects (connectivity, depth, and a number of cycles) [61]. However, AGs have drawbacks regarding how they reflect multistep attacks when the probability of exploiting a vulnerability is applied, especially in a dynamic analysis in which evidence of ongoing attacks is detected. To overcome these drawbacks, the Bayesian inference may be useful, as this method offers a convenient method for probabilistic analysis while simultaneously providing a representation of attack paths more compact than those of the conventional AG methods.

A Bayesian network (BN) is a directed acyclic graph (DAG) in which the nodes represent variables, the edges denote the existence of direct causal dependencies among the linked variables, and the strengths of these dependencies are quantified by conditional probabilities [62]. The Bayesian inference enables an estimation of the risk of compromises in the system given the vulnerabilities and interconnections; it also accounts for the spreading of a multistep attack while considering evidence of exploitation [63]. The application of a BN for network security assessment was initially proposed by Liu and Man [22], with the resulting model referred to as a BAG. The authors modeled potential attack paths as more compact and scalable than the authors used the BAG to model potential attack paths as more compact and scalable than those in conventional AGs; they provided methods to obtain quantitative values through calculations of the unconditional probability or posterior probability. Frigault et al. [19] assigned a vulnerability exploitation probability to a BAG using standard techniques such as CVSS [64]; they also allocated a disjunctive or conjunctive identifier to a node with more than two parents. Using a genetic algorithm, Poolsappasit et al. [23] revisited the BAG concept to present a BN framework that can perform a risk assessment and quantify the expected benefits of an investment based on a user-specified cost model.

Muñoz-González et al. [25] introduced efficient algorithms to make exact inferences on the BAG model via practical static and dynamic analyses. The application of the variable elimination (VE) algorithm [22] is limited to small networks owing to its computational complexity. However, the authors showed through an extensive experimental evaluation that, in terms of time and memory resources required by the VE algorithm, their junction tree algorithm is capable of improving the performance even in a network consisting of hundreds to thousands of nodes. Muñoz-González et al. [63] introduced an approximate inference technique based on the loopy belief propagation algorithm, which linearly scales in terms of the number of nodes for both static and dynamic analyses. An approximate calculation method was applied by focusing on the fact that absolute accuracy is less critical because the CVSS values used in the BAG are typically rough estimates. The quantitative results of the risk assessment are mainly used to determine the priority rather than to obtain accurate values. The authors demonstrated through several experiments that the approximate method has linear scalability, and the accuracy is sufficient to allow system administrators to decide necessary actions. The security risk is a function of the likelihood of the occurrence of a threat event and its potential adverse impact. However, previous studies did not include this type of impact in their frameworks [19, 22, 23]; they considered only the impacts of single hosts [21] and did not consider relationships among impacts of vulnerability exploitations. This is different from the case of BAGs, which reflect causal relationships among vulnerabilities.

Some BAG-related studies have tried to handle impacts or propagations of impacts. Khosravi-Famad and Ghaemi-Bafghi [21] introduced a risk assessment framework that includes a calculation of vulnerability exploitation impact. However, their study considered only the effects of single hosts and did not address impact propagation. Feng et al. [18] proposed a model that performs vulnerability propagation analysis when the probability of a BN node exceeds a certain threshold. However, the propagation in this study is focused on finding the most vulnerable path based on

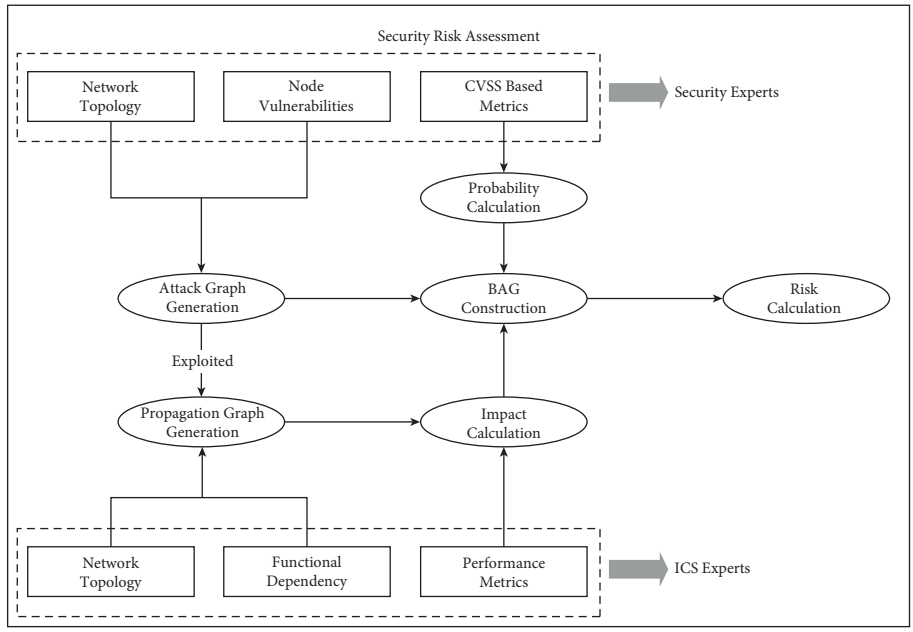

Figure 4: Proposed security risk assessment framework.

vulnerability information. Mahmoud et al. [65] set values ranging from 0.0 to 1.0 for the function and class values according to the node's function (function value) and the traffic generated from the node (class value) for each node to consider the propagation of adverse impacts. However, in our study, the impact is calculated by considering only the function importance and network traffic. Thus, this study presents an approach for assessing the risk using the likelihood obtained through a BAG and the consequences resulting from the proposed framework.

4. Proposed Framework

The architecture of the proposed security risk assessment framework is depicted in Figure 4. The framework is composed of two principal parts: consequence and likelihood. In the likelihood part, the probability of a component being attacked and exploited is calculated. This part first collects information on network topology, security vulnerabilities, and exploitability probabilities. This information can be automatically obtained by tools such as vulnerability scanners, network scanners, and vulnerability databases or manually by security experts. The obtained information is used as input to a BAG enabling static and dynamic probability analyses that can consider a multistage attack that exploits a sequence of vulnerabilities.

The consequence part quantifies the impact of the process when cyberattacks compromise one or several digital devices. In this part, identification of the network topology, functional dependencies among components, and essential functions for the operation of the process is conducted by ICS experts. The identified information is fed into the proposed functional dependency model so that the consequences of a cyberattack on a component can be quantified as numeric values. While taking into account the causal relationships among components in ICSs, the security analyst can assess the cascading impact of a cyberattack on one component that spreads to other systems and processes.

After conducting the consequences and likelihood assessments, the overall risk can be obtained by multiplying the results. Security analysts can perform various assessments using the obtained risk; they can rank the risk for the identification of the critical components that need more attention or can mitigate the risk below a certain threshold by applying security controls to components at high risk. Calculated risks can also be used for what-if assessments, which apply assumptions about operational security vulnerabilities, such as the use of vulnerable maintenance equipment for multiple components or the sabotage of an ICS by an insider who has authority for multicomponents. Details of the consequences and likelihood assessments are described in subsequent sections.

4.1. Likelihood Assessment. As a critical factor for calculating security risk, likelihood means the probability that an attacker can successfully exploit the vulnerabilities of a particular component. The CVSS is an open framework for communicating the characteristics and severity of software vulnerabilities [30] and is frequently used as an indicator of likelihood because the exploitability metrics of the CVSS denote the probability of successfully exploiting a vulnerability based on a standardized framework, through which the ease of exploitation can be quantified. However, there is a limitation. Exploitability metrics can be used only to show the likelihood of successfully exploiting a vulnerability in an ICS. This is because an ICS component is not a stand-alone system but is a part of a network, and an attacker does not execute a cyberattack only on the target component. Indeed, an attacker can attack right in front of the target component and perform the attack through other components

connected to the target component. Therefore, when evaluating the likelihood, it is necessary to consider not only an attack directly applied to the target component but also a multistage attack in which an attacker starts from an access terminal and reaches the target by sequentially exploiting accessible components.

While considering these constraints, we use BAGs as a method to calculate the likelihood. The BAGs have been applied in industrial fields for decision-making, dealing with uncertainty, and risk assessment for the past two decades [66]. This is because BAGs are a powerful tool for static and dynamic risk assessment of networks because it considers the interdependencies between vulnerabilities that indicate how attackers can compromise various network resources by modeling multistage attacks that exploit sequences of vulnerabilities [22, 67]. The BAGs offer a compact means to encode the entire range of conditional relationships, suitable for representing security metrics based on attack graphs [19]. The ability to evaluate the likelihood of attack goals and predict potential upcoming attacks using probabilistic inference techniques is the primary motivation we choose BAGs for the likelihood assessment. Furthermore, BAGs also have the advantage of updating the model by calculating the posterior probabilities when new evidence is available.

A BN is a probabilistic graphical model in which nodes symbolize random variables and edges indicate conditional dependencies between pairs of nodes using a DAG and a set of conditional probability distributions [22]. Let us assume a set of node states in a BAG, denoted by $S = \{s_1, s_2, \ldots s_n\}$; the directed Markov property gives the joint probability distribution of S as follows:

$$\Pr(S) = \Pr(s_1, s_2, \ldots, s_n) = \prod_{n_{i=1}} \Pr(s_i|\mathrm{Pa}(s_i)), \quad (1)$$

where $\mathrm{Pa}(s_i)$ indicates the set of parent nodes of s_i. Each parent node has a direct edge pointing to s_i, and the existence of an edge represents causal probabilistic dependence between the two nodes.

Formally, a BAG is defined as a tuple BAG = (S, τ, ϵ, P) [21, 23] based on a BN, where

(1) S denotes the set of node states

(2) τ denotes the set of ordered pairs; each pair represents an edge connecting the nodes in the graph

(3) ϵ denotes a set of conjunctive or disjunctive relationships among multiple edges pointing to a node with a possible value of {AND, OR} and

(4) P represents the set of conditional probability tables (CPTs) associated with the BAG nodes.

The nodes in a BAG are Bernoulli random variables that represent the different security states that an attacker can reach. If an attacker achieves his goal (compromising a node $s_i \in S$), the state is represented as *true* ($s_i = 1$). The probability that a node s_i is compromised will then be expressed as $Pr(s_i = 1) = p$. Consequently, the probability that a node is not compromised (or the node state is *false*) is $\Pr(s_i = 0) = 1 - p$, with $p \in [0, 1]$.

In a BAG, an edge represents a possible vulnerability from one $Pa(s_i)$ to s_i. The probability that an attacker will successfully exploit a node s_i from its parent node $Pa(s_i)$ using vulnerability e_i is represented as $\Pr(e_i)$. A common approach to obtain $\Pr(e_i)$ is to utilize a CVSS. In previous studies [19, 21, 23], the exploitability metrics of a CVSS are preferred because this strategy provides a standardized framework to quantify the ease of exploitation. A method for assigning a certain probability to zero-day vulnerabilities was proposed in [63]. The exploitability metrics were calculated by (2) in CVSS version 2, and equation in CVSS version 3, with $\Pr(e_i) \in [0, 1]$.

$$\Pr(e_i) = 2 \times \mathrm{AV} \times \mathrm{AC} \times \mathrm{AU}. \quad (2)$$

$$\Pr(e_i) = 0.822 \times \mathrm{AV} \times \mathrm{AC} \times \mathrm{PR} \times \mathrm{UI}. \quad (3)$$

The exploitability metrics of CVSS version 2 comprise the access vector (AV), access complexity (AC), and authentication (AU). The exploitability metrics of CVSS version 3 comprise the attack vector (AV), attack complexity (AC), privileged required (PR), and user interaction (UI).

The characteristics of vulnerabilities can change over time. If the exploitation technique is not yet mature or is unproven, such that exploitation is possible, the exploitability metrics may be smaller than the original metrics. To take into account, the properties of vulnerabilities that change over time, the application of CVSS temporal metrics was proposed in [21]. These metrics adjust the value of the exploitability (equations (2) or (3)) relative to the time of the assessment, as shown in the following:

$$\mathrm{TP}(e_i) = \Pr(e_i) \times \mathrm{ECM} \times \mathrm{RL} \times \mathrm{RC}. \quad (4)$$

Here, TP denotes the temporal probability and adjusts the exploitability metrics at the time of risk assessment. ECM refers to the exploitation code maturity status, RL defines the remediation level of the vulnerability, and RC measures how confident the vulnerability report is [64].

Under BN representation, each node s_i has a CPT that is a tabular form of the conditional probability distribution of s_i, comprised by its parent nodes, $\Pr(s_i|Pa(s_i))$. If a node s_i has no parent, the CPT is a marginal probability distribution of s_i, denoted as $\Pr(s_i)$. If s_i has more than one parent, the conditional probability distribution is calculated as a function of ϵ. If attackers need to compromise all parent nodes to compromise node s_i, then ϵ = AND. In this case, the local conditional probability distribution is defined by equation (5). When the relationship between incoming edges to node s_i is *OR* (ϵ = OR), the local conditional probability distribution is defined by equation (6).

Because devices that are widely used in ICS environments, such as PLCs and DCSs, are not exposed to IT environments, there are only limited known vulnerabilities. This characteristic of ICSs implies that there may be more zero-day vulnerabilities than in an IT environment. Therefore, we apply the leak factor $\Pr(z_i)$, presented in [63], in equations (5) and (6) The leak factor models whether there is a certain probability by which s_i becomes *true* even when all parent nodes are in *false* state. The aforementioned study [63] suggested

that the leak factor can be estimated using a common weakness scoring system [68], but a precise approach was not provided. Therefore, we assume in this study that expert judgment is used to determine the leak factor.

$$\Pr(s_i|Pa(s_i)) = \begin{cases} \Pr(z_i), & \exists s_j \in Pa(s_i)|s_j = 0, \\ 1-(1-\Pr(z_i))\left(1-\Pr\left(\bigcap_{s_j=1} e_j\right)\right), & \text{otherwise.} \end{cases} \tag{5}$$

$$\Pr(s_i|Pa(s_i)) = \begin{cases} \Pr(z_i), & \forall s_j \in Pa(s_i)|s_j = 0, \\ 1-(1-\Pr(z_i))\left(1-\Pr\left(\bigcup_{s_j=1} e_j\right)\right), & \text{otherwise.} \end{cases} \tag{6}$$

In equations (5) and (6), $\Pr(e_j)$ denotes the probability of exploitation, indicating how efficiently an attacker exploits node s_j. In this study, we use the adjusted metrics in equation (4) to assign the probability of $\Pr(e_j)$. The product rules of the probability to derive $\Pr(\bigcap_{s_j=1} e_j)$ are as follows:

$$\Pr\left(\bigcap_{s_j=1} e_j\right) = \prod_{s_j=1} \Pr(e_j). \tag{7}$$

To derive $Pr(\bigcup_{s_j=1} e_j)$, Liu and Man [22] presented a method based on a noisy-OR [69] operator. The product rules of the probability to derive $Pr(\bigcup_{s_j=1} e_j)$ are as follows:

$$\Pr\left(\bigcup_{s_j=1} e_j\right) = 1 - \prod_{s_j=1} \left[1-\Pr(e_j)\right]. \tag{8}$$

Once the CPTs have been assigned to all attributes in the BAG, we can conduct static analysis by obtaining the unconditional probability of each node, which is calculated by merging the marginal cases at the node. This indicates the likelihood that an attacker successfully reaches the target security state of the node. Using the Bayes rule, it is possible to calculate the unconditional probability distribution $\Pr(s_i)$ from the CTPs as follows:

$$\Pr(s_i) = \Pr(s_1, s_2, \ldots, s_i) = \prod_{j=1}^{i} \Pr(s_j|Pa(s_j)). \tag{9}$$

Over time, a certain probability of an attack occurring during the lifetime of the network exists. If we detect attacks, we can compute the posterior probability by updating the values relevant to the observed attacks in the BAG. The BAG's posterior probability calculation property allows dynamic analysis throughout the lifetime of the network. Let $E = \{s_1, s_2, \ldots, s_n\}$ be a set of nodes observed as exploited (i.e., $s_j = 1, \quad \forall s_j \in E$). Likewise, let $S' = S - E$ be a set of nodes whose posterior probability must be determined. The posterior probability distribution for dynamic analysis is $\Pr(S'|E)$, which can be computed using the Bayes theorem as follows:

$$\Pr(S'|E) = \frac{\Pr(E|S') \times \Pr(S')}{\Pr(E)}. \tag{10}$$

4.2. Consequence Assessment. Among areas potentially affected by an attack, system functions are critical for risk assessment in ICSs. Even if two systems present the same configuration (e.g., hardware, software, and vulnerability), the results of cyberattacks will be different if one system performs critical functions (e.g., tripping a nuclear reactor in abnormal situations) and the other performs auxiliary functions. It is crucial in risk assessment to evaluate consequences when cyberattacks compromise the functions of a system. As discussed earlier, given that a system's functionality is usually dependent on other systems in the ICS, to evaluate the consequences of a compromised function, the analyst needs to simultaneously assess the consequences on the attacked system and the effects on other systems that rely on the attacked system.

An exploited vulnerability that impacts the CIA triad directly leads to consequences that may include information disclosure, system disruption, and/or data alteration [35]. In ICSs, adversaries tend to attack availability and/or integrity rather than confidentiality [70]; thus, system disruption and data alteration constitute the primary consideration. System disruption and/or data alteration result in loss of essential functions, system degradation, and/or unauthorized changes (algorithm, set point, and process data) [35]. These results may affect the physical layer, for instance, by causing failure of control valves or pumps, by causing abnormal control as a result of altered information or algorithms, by preventing operators from monitoring the condition of the facility, or by providing false information to operators leading to faulty control. The effects of system disruption and/or data alteration can propagate to other systems. The disruption of a system will affect the function of other systems serviced by the disrupted system. Likewise, data alteration will adversely affect other systems by delivering results from altered information or algorithms. Affected systems can again affect the physical layer, and these influences repeatedly propagate to other connected systems. Therefore, when assessing the potential impact on critical infrastructure from a possible ICS incident, it is important to incorporate effects on the physical process/system and on dependent systems/processes, among other possibilities [4].

In the consequence part, we propose a functional dependency model (FDM) that allows ICS experts to quantify the consequences by describing what other components or a

certain component properly depends on to function. The FDM enables a representation of the relationship among components constituting an ICS such that ICS experts (e.g., operator, system engineer, and safety engineer), who understand the systems and processes, can provide a list of functional dependencies among components. Once the FDM is complete, another task is to list the functional goals of the ICS. Functional goals refer to essential functions for a system or process to properly operate. For example, for monitoring or control systems, availability and integrity will be essential functions, whereas for safety-grade protection systems with fail-safe, applied integrity will be a functional goal. Functional goals may have different levels of importance in the overall process; these levels can be expressed through numerical weights. With the FDM and functional goals, security analysts can identify after a cyberattack how much impact a specific component has on the essential function of the systems and processes and can represent the consequences as quantified numeric values.

We begin with the definition of the terms component, system, and network used in the FDM. A component C_i is a basic unit constituting a system. Each component executes logic by itself, and components can transfer and receive signals through the network or through hardwired interfaces. In ICSs, an individual device such as a PLC or a personal computer (PC) becomes a component. A system S_i is a unit in which components are grouped to perform one or more functions. A system is defined as a set of components, $S_i = \{C_1, C_2, \ldots, C_n\}$. Finally, a network N_i denotes a facility composed of systems to conduct business and N_i is a set of systems, $N_i = \{S_1, S_2, \ldots, S_n\}$. The relationship between component, system, and network is shown in Figure 5.

Definition 1 (failure mode). The exploitation of vulnerabilities can impact a component, and such an impact can lead to multiple failures. In this study, we represent the potential failures of a component due to the exploitation of vulnerabilities as a failure mode. The analysis team can establish the failure mode according to the characteristics of a component. A failure mode for a given component C_i is defined as a set consisting of a CIA triad, $\mathrm{FM}_i = \{C_i^C, C_i^I, C_i^A\}$, which represents the consequences of the security event. Each element of FM_i is a Boolean variable; if the value is set to true, it means that the related CIA element is affected by the exploitation. To determine the failure mode, we utilize the impact metrics of the CVSS, which measure the impact on the confidentiality, integrity, and availability with three scales (v2: none, partial, and complete; v3: none, low, and high). For strictness sake, if the impact exceeds "none," the related CIA triad is considered to be compromised.

Definition 2 (propagation launching). Propagation begins when vulnerability V_j of component C_i is exploited. We differently apply propagation launching depending on which vulnerability is being exploited. If the vulnerability exploitation result is the acquisition of root privileges, we assume that all elements of the failure mode are affected because an attacker has full authority over the system.

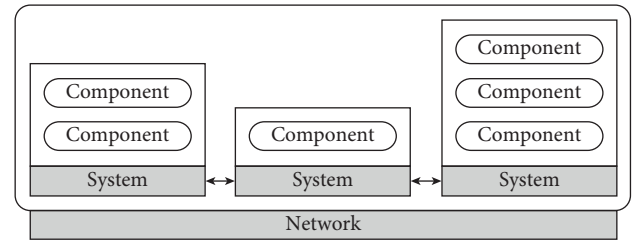

Figure 5: Relationship between component, system, and network.

Otherwise, the failure mode depends on the impact of the vulnerability when compromised:

$$\mathrm{FM}_i = \begin{cases} \left\{C_i^C = T, C_i^I = T, C_i^A = T\right\}, & acq.priv = \text{root}, \\ \bigcup_n^{C,I,A} \{C_i^n = T \text{ if } \mathrm{Impact}_n > \text{none}\}, & \text{otherwise}. \end{cases} \tag{11}$$

Definition 3 (function failure based on impact propagation). To analyze how a functional impact propagates when one or more components are compromised, we define a functional dependency for a component C_i as follows: $D = \mathrm{Sup} \longrightarrow \mathrm{Dep}$, if the state of Dep (dependent) depends on the state Sup (supplier) to achieve the state Dep. Sup is a logical operation consisting of the failure-mode elements of components on which C_i depends for functionality. A logical operation can have logical operators (i.e., AND and OR) and logical functions depending on the system's characteristics. Dep refers to a function that is affected when Sup is compromised. A component C_i may have multiple functional dependencies.

We define a set of functional dependencies as $\mathrm{FD}_i = \{D_1, D_2, \ldots, D_n\}$ when a component C_i has multiple functional dependencies.

Definition 4 (essential functional goals). A component C_i has its essential functional goals, $\mathrm{FG}_i \subseteq \{C_i^C = T, C_i^I = T, C_i^A = T\}$. Essential functional goals are functions that C_i must satisfy to operate as intended. A component does not always have to satisfy all CIA triads. Because goals differ according to a component's characteristics, they can be identified by an expert on the corresponding component (e.g., design and operation).

To analyze how a functional impact propagates when a component is compromised, we need only FD_i and FG_i for each component C_i. This information is used to define an information template. ICS experts need to only identify those components that their components depend on; they do not have to understand the entire system's configuration to identify functional dependencies. Identification of functional goals can also be accomplished by simply understanding their own components' main functions and failure modes. We expect that FDM characteristics will reduce the cost required for both the communication between ICS and security experts and human errors due to miscommunication.

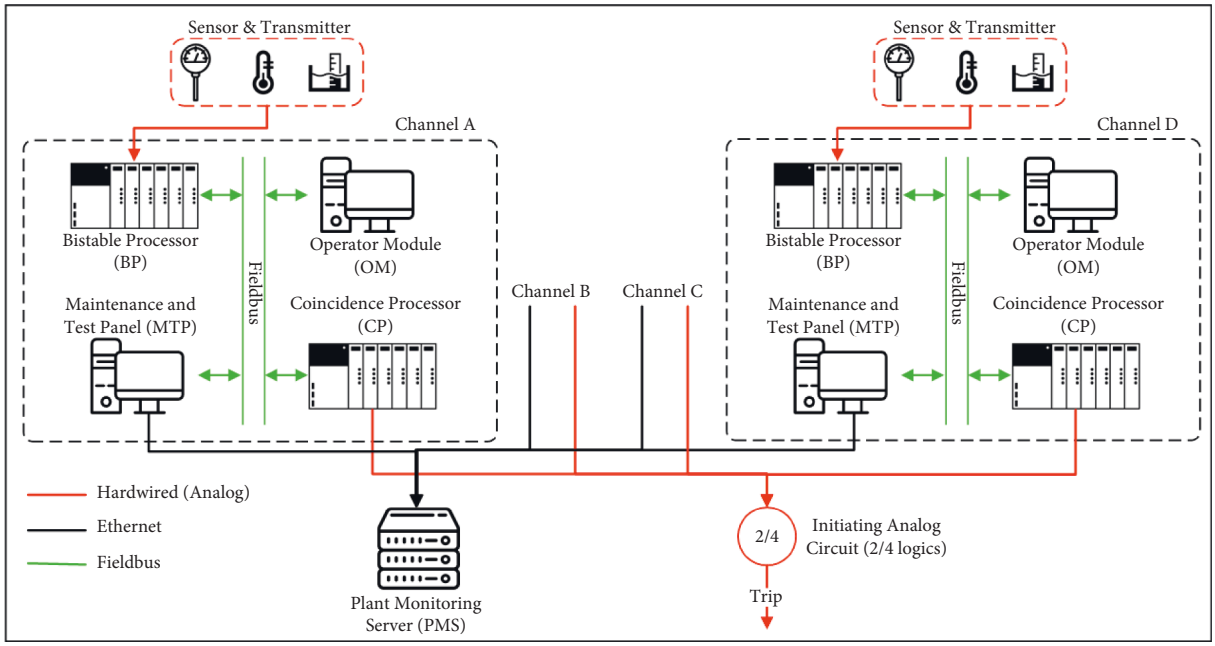

Figure 6: Test network: plant protection system.

In the proposed model, when a component is compromised, propagation is launched and can identify which failure-mode elements of that component are impacted using FM_i. Then, it is possible to identify which components are affected by searching for components for which Sup of FD_n becomes true due to FM_i. At this time, whether the identified infringed component's function is essential can be checked through a comparison with FG_n of the corresponding component C_n. If the check shows that essential functional goals are compromised, it can be concluded that the component is not behaving as intended. By repeating these processes, it is possible to analyze how far the impact will spread when one component is compromised.

The consequences of a cyberattack on a component are expressed as the weighted mean of the failed FG s to the total number of FG s, with weights representing the importance of each functional goal. Because the consequences are given by the ratio of failed functions among all FGs, we call this a failure metric. This failure metric is evaluated on a single system or on a global network, depending on the purpose of the analysis. Let $FG = \{x_1, x_2, \ldots, x_n\}$ be a set of essential functional goals in the system or global network; FG has a Boolean value, and $W = \{w_1, w_2, \ldots, w_n\}$ denotes the weight set for each FG, where n is the number of components. The weight is an integer value between 0 and 10 determined by facility experts depending on its significance; the failure metric is obtained as follows:

$$M = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}. \tag{12}$$

4.3. Risk Assessment. The security risk is a function of the likelihood of a threat event's occurrence and the potential adverse impact should the event occur [17, 71]. Thus, the risk is calculated as Risk = Likelihood × Consequence. We can obtain the risk by considering the causal relationship between the likelihood and the consequences using the unconditional probability of the BAG for the likelihood and failure metrics for the consequences. The risk can also be obtained from the dynamic analysis, and the results differ from static analysis. Evidence is used to adjust the probabilities obtained from the BAG and to accordingly change the risk. The failure metric is also changed according to the evidence. However, if there is evidence that a particular component has been exploited, it can be concluded that components that depend on that component are also malfunctioning. Therefore, when evidence exists, determining the function-failure components is as critical as updating the risk value.

5. Experiments and Results

This section applies the proposed risk assessment framework for a plant protection system of the nuclear power plant depicted in Figure 6. We first introduce the test network, which is a plant protection system (PPS) in a nuclear power plant. Then, we compute the consequences and likelihood of the test network by feeding information such as network topology, vulnerabilities, and functional dependencies to the framework. As a result, we can obtain risks for the components, system, and entire network; through these results, we show that exploitation of one device alone can have impacts on the entire system or network because of the high functional relationship among components.

5.1. Test Network and Experimental Configurations. Figure 6 depicts the test network used in this study. The test network is a simplified PPS associated with an existing

Table 1: List of vulnerabilities in the test network.

Host	Vulnerability	Score	Type
BP	Remote buffer overflow	0.16	r2r
CP	Remote buffer overflow	0.16	r2r
OM	Acquire privileges by providing an arbitrary program name as a command-line argument	0.39	l2r
MTP	Acquire privileges by providing an arbitrary program name as a command-line argument	0.39	l2r
PMS	Remote buffer overflow	0.16	r2r

nuclear power plant. The PPS trips the reactor when the observed variables deviate from specified safety set points to preserve the reactor coolant pressure boundary (RCPB) and assist in mitigating the consequences of postulated accidents [72]. The PPS receives signals from field sensors (e.g., temperature, pressure, and level sensors), executes logic that generates a reactor trip signal when the received values are out of the predefined ranges, and transmits the received information to other systems, e.g., the plant monitoring system (PMS). The absence of a trip signal from the PPS during an emergency will prevent the insertion of the control element assemblies to shut down the nuclear chain reaction and reduce the heat generation rate [73]; this can cause excessive core temperatures above fuel design limits. Exceeding such limits can lead to a collapse of the RCPB, which can cause the release of radioactive material into the environment and threaten the health and safety of the general public [73].

The PPS comprises quadruple channels with identical configurations from A to D. Each channel is in turn composed of a bistable processor (BP), a coincidence processor (CP), an operator module (OM), and a maintenance and test processor (MTP). The BP and CP execute their logic on the PLC, and the MTP and OM run on an industrial personal computer. We assume that for PLCs, POSAFE-Q [74] is used, and the QNX operating system is employed for the MTP and OM. A fieldbus network is used for communication inside the PPS, and the MTP transmits process values through TCP/IP to the plant monitoring server of the PMS, which is a nonsafety system.

The BP receives process values from field sensors via hardwired connections using analog signals (e.g., 4–20 mA and 0–10 VDC) and transmits the processed values to the fieldbus. The CP receives trip signals from four channels of BPs through hardwired connections (omitted in the figure for simplicity) and generates trip signals through the 2/4 voting logic. The operator uses the OM for manipulation, such as to change the set points of the PPS, and the MTP is employed to maintain and test the PPS. Another MTP role is to deliver PPS information to the PMS. The PMS receives data from four redundant MTPs and performs its role if more than two instances of MTP data are available and consistent. Operators use the PMS to monitor the information needed for daily operation. If the PMS is not available, such a condition interferes with the daily operations required to generate electricity. Thus, nonperformance of PMS (nonsafety system) is associated with the stable supply of electricity and financial losses, rather than releasing radiological materials into the public environment. We assume that the PMS uses Linux (Red Hat 8.3) as the operating system.

Given that finding public vulnerabilities in the systems of the test network is difficult, we used hypothetical vulnerabilities for our experiments, as shown in Table 1. One hypothetical vulnerability for BP and CP, which operate on the PLC, is a "remote buffer overflow," which allows a change of the logic (i.e., application in the PLC) through the acquisition of privileges of the PLC over the network. Because an attacker can acquire root privileges through the network, we categorize this vulnerability as remote-to-root (r2r) type. In addition, we assume that the PLC does not have a user interface but can connect to a support equipment (e.g., laptop) for maintenance. The hypothetical vulnerability for OM and MTP is "acquire privileges by providing an arbitrary program name as a command-line argument." The type of these vulnerabilities is local-to-root (l2r). The score (exploitability metric) is calculated through CVSS V3 metrics. The other assumption for the test is that the metrics take a long time to change. Therefore, we did not consider temporal metrics in our experiments.

We present the functional dependencies (FD) and essential functional goals (FG) of the PSS we assumed for the experiment in Table 2. The left-most column provides the corresponding system name; each component has an information template consisting of FD and FG. For example, the first row means that the FG of the BP is integrity, and the integrity of the BP depends on the integrity of the MTP or OM. The 2/4 denotes the two-out-of-four logic, which returns *true* when more than two inputs are *true*. Since FD*s* and FG*s* are related to the system's functionality, they are usually identified by ICS experts rather than security experts. ICS experts can identify FG*s* and FD*s* without deep security knowledge by comprehending the meaning of the CIA triad and simple logical operations.

The security administrator can set weights for each functional goal according to its importance and analysis purpose. Table 3 represents two possible weights differently set according to the analysis purpose. The first example (Weight_1) is weights set from the perspective of power plant operation. In this case, the BP and CP weights are set to 10, with the highest level denoting those components that directly perform safety-related functions. The BP generates a trip signal when the monitored conditions approach specified safety settings; BP then transmits a trip signal to CPs. The CP, using a two-out-of-four voting logic, delivers a trip command to the related hardware to shut down the plant and preserve the RCPB. Given that OM and MTP do not perform a direct safety-related function but play roles in supporting BP and CP, we set their weights to two, i.e., the lower value. The weight of PMS is set as one because the system performs a nonsafety monitoring function, but it is

Table 2: Information template examples.

Component	Element	Description
BP	FD_{BP} FG_{BP}	$[MTP^I \text{ or } OM^I] \longrightarrow BP^I$ BP^I
CP	FD_{CP} FG_{CP}	$[2/4\,(BP^I) \text{ or } MTP^I \text{ or } OM^I] \longrightarrow CP^I$ CP^I
MTP	FD_{MTP} FG_{MTP}	$[BP^I \text{ or } OM^I] \longrightarrow MTP^I$ MTP^I
OM	FD_{OM} FG_{OM}	$[BP^I \text{ or } MTP^I] \longrightarrow OM^I$ OM^I
PMS	FD_{PMS} FG_{PMS}	$2/4\,(MTP^I) \longrightarrow PMS^I$ $2/4\,(MTP^A) \longrightarrow PMS^A$ PMS^I and PMS^A

Table 3: Example of functional goals' weights.

FG	BP^I	CP^I	MTP^I	OM^I	PMS^I	PMS^A
$Weight_1$	10	10	2	2	1	1
$Weight_2$	10	10	0	0	0	0

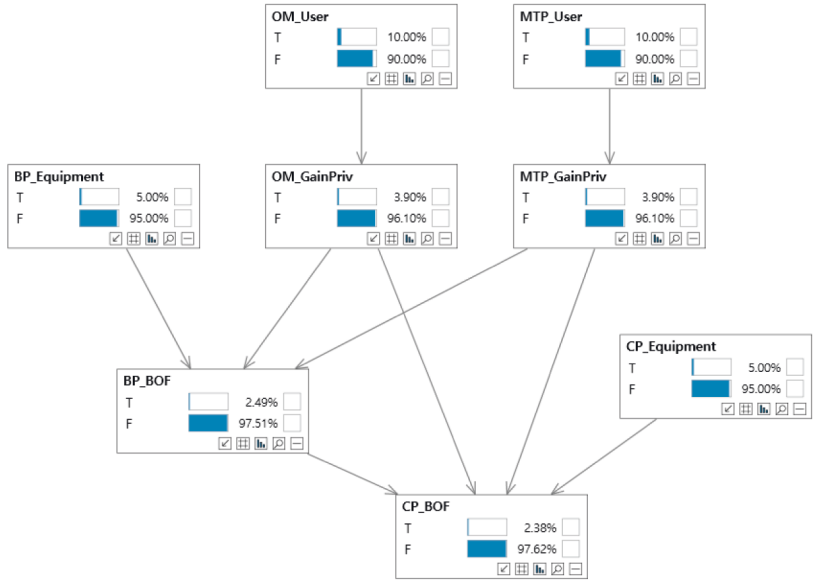

Figure 7: BAG of test network with unconditional probabilities.

related to the daily operation. If there is an adverse event on PMS, operators will stop the plant, and the plant cannot generate electricity. However, when performing the analysis from a safety perspective, the values should be set as shown in $Weight_2$ because safety-related functions are essential. In this case, only the BP and CP, which directly perform the safety-related function, set the weight to ten, and set the remaining weights to zero.

5.2. Experimental Results. Our framework is mainly composed of likelihood and consequence parts. In this section, the experimental results are described in order. When the security administrator performs a risk assessment based on the criticality, a BP or CP is highly likely to be selected as an attacker's target because it has a high weight, as shown in Table 3. As a consequence, if a CP is selected as a target, the likelihood part of our framework generates a BAG and calculates unconditional probabilities, as shown in Figure 7, by utilizing the vulnerabilities listed in Table 1. Each node in Table 1 is a Bernoulli random variable representing the state of vulnerability exploited. The entry points of the graph are attacks from users of each monitoring component (e.g., MTP and OM) or of maintenance equipment for PLCs (e.g., BP and CP). Since the test network has no network connection from the outside, we assume insiders as the only sources of attacks. The probabilities of nodes reflect subjective prior knowledge of insider capabilities, and we set the value of Pr(C) to 10% for monitoring components and 5% for PLCs. The unconditional probabilities calculated from BAG are displayed for each node with a T label. We implemented the BAG model using Bayes Server 9.5 [75].

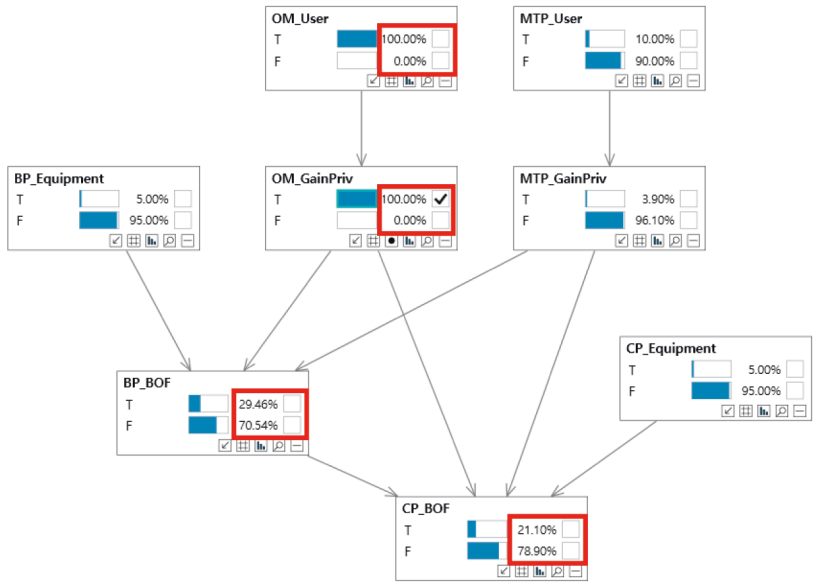

Figure 8: BAG of test network with one detected attack.

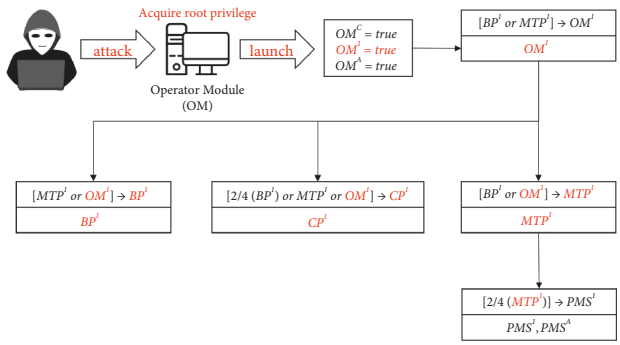

Figure 9: Example of impact propagation.

If there is evidence of an attack on a specific component, a BAG can update the posterior probability distribution based on the Bayesian theorem. For example, when a security administrator detects an attack on an OM with a BAG, we can obtain the posterior probability distribution as shown in Figure 8. When there is evidence of an attack on OM, we can recognize that the probability of an attacker gaining access privilege of BP and CP increases from 2.49% to 29.46% and from 2.38% to 21.10%, respectively. Accordingly, the security administrator will be probably more interested in the BP and CP and proceed with their investigation rather than the MTP.

The consequence assessment based on Table 2 is conducted as follows. The impact is propagated when an attacker attacks the OM and acquires root privileges, as depicted in Figure 9. In the beginning, all elements of the OM's failure mode are affected after the attacker gains root privileges of OM, according to Definition 2. Thus, $\mathrm{FM}_{\mathrm{OM}} = \{\mathrm{OM}^C = T, \mathrm{OM}^I = T, \mathrm{OM}^A = T\}$ and the OM do not behave as intended because their essential functional goal OM^I is compromised. The compromise of OM^I satisfies the Sup of other components (BP, CP, and MTP); consequently, related Dep functions are affected. Because the functions of BP, CP, and MTP affected by FD coincide with the essential functional goal, the three components are also identified as not working as intended. The compromise of BP^I affects the FD of MTP and OM, but because they have already been identified as compromised, there is no need to further investigate the FD. CP^I does not affect other components. Among the uncompromised components, MTP^I is related to the FD of the PMS, but because a compromise of one channel alone does not satisfy a two-out-of-four logic (2/4 logic), the PMS is not affected by the MTP. In the figure, the upper rectangle represents FD, the lower rectangle represents FG, and the text shown in red indicates the compromised functions.

Table 4 shows the static risk obtained by combining the consequence assessment results considering impact propagation and the likelihood results obtained through BAG. The likelihood in the table represents the exploitable probability obtained through BAG; consequence means the ratio of functional goals in single-channel (Ch. A), four-channel (System), and global networks affected by each component exploitation. Risk is a value obtained through the product of likelihood and consequence (Risk = Likelihood × Consequence) because the security risk is a function of the probability of a threat event's occurrence and the potential adverse impact should the event occur [17, 71]. We assume that all attacks are performed by an insider who already has user privileges. Therefore, Table 4 enumerates only the root privilege-related exploitations. The failure metric corresponding to a single system is omitted because the PMS is a nonredundant system.

If there is evidence that a component is already exploited, knowledge about attack incidents is used to update the probabilities using the Bayesian inference techniques of forward and backward propagation [23]. Table 5 shows the dynamic risk when probabilities are updated by

Table 4: Static risk for the test network.

Event		Likelihood (%)	Consequence			Risk		
			Ch. A	System	Global	Ch. A	System	Global
W_1	BP_r	2.49	100.00%	25.00%	24.49%	2.49	0.62	0.61
	CP_r	2.38	41.67%	10.67%	10.20%	0.99	0.25	0.24
	MTP_r	3.90	100.00%	25.00%	24.49%	3.90	0.98	0.96
	OM_r	3.90	100.00%	25.00%	24.49%	3.90	0.98	0.96
W_2	BP_r	2.49	100.00%	25.00%	25.00%	2.49	0.62	0.62
	CP_r	2.38	50.00%	12.50%	12.50%	1.19	0.59	0.59
	MTP_r	3.90	100.00%	25.00%	25.00%	3.90	0.98	0.98
	OM_r	3.90	100.00%	25.00%	25.00%	3.90	0.98	0.98

Table 5: Dynamic risk for test network (evidence: OM_r).

Event	Likelihood (%)	FG	Consequence			Risk		
			Ch. A	System	Global	Ch. A	System	Global
$BP_r \mid OM_r$	29.46	✓	100.00%	25.00%	24.49%	29.46	7.37	7.21
$CP_r \mid OM_r$	21.10	✓	100.00%	25.00%	24.49%	21.10	5.28	5.17
$MTP_r \mid OM_r$	3.90	✓	100.00%	25.00%	24.49%	3.90	0.98	0.96
OM_r	100.00	✓	100.00%	25.00%	24.49%	100.00	25.00	24.49

Bayesian inference techniques, reflecting the likelihoods of other potential outcomes in light of a detected attack incident in OM. Since there is no significant difference in result values due to the differences in weight, the table shows only the application of $Weight_1$. The FG column of the table indicates whether it has already been adversely affected by attack incidents. For example, if FG of $BP_r \mid OM_r$ is marked, it means that BP has already lost the integrity of its functional goal due to an attacker's acquisition of the root privilege of OM. If there is no dependency between the functional goals of OM and BP, the FG value of $BP_r \mid OM_r$ will remain unchecked. Note that the risk columns in Table 4 and 5 are multiplied by 10,000 to prevent the numbers from becoming too small and, therefore, difficult to interpret. Security administrators can scale the risk values as needed. Furthermore, probabilities are used to prioritize risk; thus, their absolute or accurate values are less critical, and the inputs of a risk assessment (e.g., CVSS) are already a rough estimate [63].

5.3. Evaluation and Comparison of Results. Our experimental results show that our proposed framework produces likelihoods, consequences, and combined risks of exploitation. In this section, we evaluate the efficacy and benefits of our framework. For this, we assess the experimental results and compare them with those of other studies. First, as we use BAGs to calculate the likelihood, the security administrator can perform static analysis to calculate the probability that an attacker can reach each state in the graph and engage in a multistage attack that exploits a sequence of vulnerabilities [22, 25]. Under the assumption of insider attack probabilities of PC and PLC types, BAGs calculate the likelihood of vulnerability exploitation for each component. As Liu et al. stated, BAGs provide a formalism for reasoning about partial beliefs under uncertain conditions [22]. For the reason that BAGs provide likelihoods considering the multistage attack, the security administrator can make decisions such as assigning security resources based on the probabilities. BAGs update the probabilities forward and backward when there is evidence of a cyberattack on a specific component, as depicted in Figure 8.

The risk assessment using only the likelihood does not reflect the importance of the components; consequently, our framework evaluates both the likelihood and the consequence. Our experimental results in Figure 7 show that the vulnerability to acquire privileges by providing an arbitrary program name to the command-line argument (for OM and MTP) has a slightly higher probability than the remote buffer overflow vulnerability (for BP and CP). Therefore, for the mitigation plan, the security administrator may prioritize fixing or applying security controls for the first vulnerability rather than the last one. However, mitigation plans based only on likelihood have their drawbacks. Suppose we assess the facility's risk from the point of view of safety, as shown by the $Weight_2$ shown in Table 3. In that case, MTP or OM does not affect safety; therefore, prioritizing the first vulnerability is not practical in terms of safety, so we can notice that the mitigation plan is inaccurate. Despite their low exploitation probability, security administrators should reconsider prioritizing safety-related components (e.g., CP or BP) for their mitigation plan. The reconsideration is also applied to the assessment based on the operation view, such as in $Weight_1$ in Table 3.

Owing to the consequences of cyberattacks being diverse and affecting both tangible and intangible assets, only a few scholars have studied quantitative methods that consider the consequences [76]. In the study of Khosravi-Famad and Ghaemi-Bafghi [21], which dealt with quantified consequences as part of the risk assessment, the impact on the organization's assets caused by exploiting vulnerabilities was computed considering the appropriate environmental

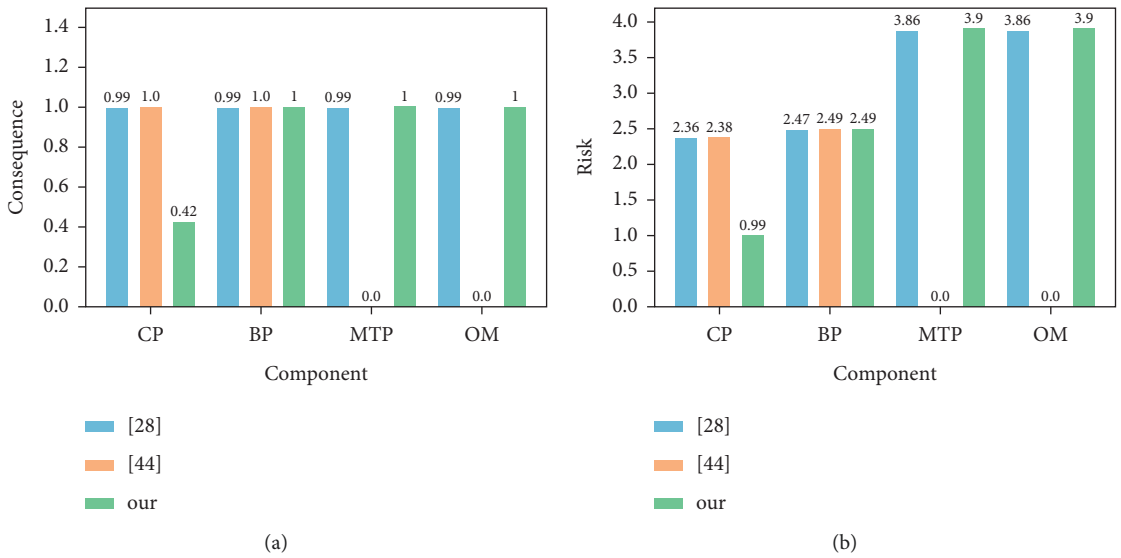

Figure 10: Comparison of risk assessment results.

metrics of CVSS. When the author's model was applied to the test network since all applied vulnerabilities resulted in the acquisition of root privilege, all impact metrics were set as complete (0.66). In addition, we assumed that integrity, availability, and confidentiality requirements were set as high (1.51), medium (1), and low (0.5) according to their levels of importance in the system. At this time, the adjusted impacts of all components had identical values of 0.99. Mahmoud et al. [65] used the degree of importance obtained by multiplying the class value by the function value. Since the function value uses the functionality of the node, we used FG, used for FDM, as the function value (scale: 0.0 $\sim$ 1.0), and the class value used the traffic generated of the node, and so the same value (1.0) was assigned to all components of the PPS that used the same network. As a result, CP and BP, which have higher impacts, have higher consequences than those of MTP and OM.

On the other hand, our framework shows different results because it assesses the consequences by considering the functional relationship among components. Our results show that all critical functions of a single channel are adversely simply affected by an attacker gaining root privileges of OM or MTP, which are considered less functionally important. This means that even if an attack succeeds in only one of the components with the most insignificant consequences, all functions of the system can malfunction. Undoubtedly, attacks on BP, identified as an important function, also affect the single channel. However, it can be seen that an attack on CP, which is also identified as an important function, affects only 50% of the critical function of a single channel. Moreover, despite the failure of a single channel, it can be confirmed that there is no adverse effect on the entire system thanks to the application of the redundancy design, consisting of four channels. Based on these results, the security administrator can recognize that BP, OM, and MTP more significantly affect system soundness through FDM; this is different from the opinions of ICS experts who initially thought of CP and BP as the most critical. Figure 10 provides a ranking of the components in order of impact, with values obtained by applying each methodology to compare our method with those of other studies considering the consequence.

As reported in the assessment results, considering both the likelihood and consequences, the security administrator can confirm that MTP and OM are the riskiest components due to higher exploitation probability and significant consequences. If the analysis is based on the BAG results, the analyst concludes that all components have similar likelihoods. However, using the results of risk assessment and considering the consequences based on functional dependencies, BP, MTP, and OM are riskier than CP, given the more significant operational consequences (obtained by applying Weight_1). Moreover, all components have similar risk levels in terms of safety (applying Weight_2). This means that while CP has a trip-related function, the four components have similar levels of risk owing to their functional linkage. If MTP or OM has a higher likelihood level, those components may have a more negligible effect on the probability of CP, owing to the attack path distance when using only the BAG. However, considering the consequences of using the failure metric under the same circumstances, an increased risk level will be found because both MTP and OM have high consequences, and this will attract the security analyst's attention. If the risk exceeds the threshold, the security administrator will attempt to reduce the risk. In particular, the security administrator can apply security controls for components with high likelihood or consequence levels to reduce the risk. Alternatively, the administrator may change the topology of the system or network to reduce the consequences and hence reduce the risk. The procedure of applying cost-optimal security control is described in detail in Poolsappasit et al. [23] and Khosravi-Farmad and Ghaemi-Bafghi [21].

6. Conclusions

In this study, a risk assessment framework for ICSs is proposed. In this framework, a BAG is constructed using a network topology generated using network scanners and host vulnerabilities, which can be obtained through vulnerability scanners. The BAG is used to identify possible attack propagation paths and their probabilities and provides static and dynamic analysis capability. However, if only a BAG is used, impact propagation on ICSs that directly or indirectly depend on the exploited ICS is not considered.

Thus, we propose an FDM to assess how an impact spreads through functional dependency relationships when one node is exploited. This FDM is constructed by ICS experts, who list functional dependencies of ICSs, essential function goals, and failure modes that they already know, and assign weights for each essential function goal. Suppose ICSs are safety-related systems that meet strict requirements, such as International Electrotechnical Commission 61508. In that case, FDM can be more efficiently constructed based on solid analysis results such as probabilistic safety assessment (PSA), failure mode and effect analysis (FMEA), hazard and operability study (HAZOP), and system-theoretic process analysis (STPA). This is cost- and time-efficient because ICS experts do not need to have a deep understanding of security. We also show that the framework makes it possible to assess how many functions, per system and per network, are not properly operating when a particular ICS component is exploited through a quantified value reflecting the importance of each function. Through impact propagation assessment on the application of the test network, we show that an attack on a component that seems insignificant can impact an essential function without any direct attack on the component performing that essential function when such component depends on the insignificant one. The proposed method enables accurate risk assessment by identifying risks with high consequences that cannot be found through the BAG analysis.

We also provide formal definitions for propagation launching under the FDM. This connects the BAG and the FDM to simultaneously assess the probability and consequence of security incidents. We also show that the risk can be calculated by assigning probability through the BAG and consequence through the FDM. Thus, an ICS with a high functional impact can have a higher risk compared to an ICS with the same likelihood and lower impact. In addition, we show that if there exists evidence that an ICS has been exploited, the probability can be updated through the BAG, and the risk can also be updated using the accumulated failure metric. However, given that the risk at this time is a risk that reflects the probability of and the consequence being exploitable, information on which ICS functions are malfunctioning is not provided. Thus, we show that it is possible to identify which functional goals have malfunctioned. When a cyber incident occurs, this enables a response based on the function of the ICS, thereby mitigating the risk to an acceptable level for critical infrastructure and maintaining safe conditions.

We plan to bridge the gap between the existing ICS hazard analysis scheme and our framework in our future work. Hazard analysis schemes such as PSA, FMEA, HAZOP, and STPA are the widely used traditional methods to identify potential system risks. Our framework assumes that ICS experts manually list functional dependencies based on their knowledge. Indubitably, experts can perform generating FDM based on the already analyzed hazard analysis results. However, manual operation is tedious, error prone, and requires a lot of resources such as time and human resources. Therefore, we expect that integrating our framework with the existing hazard analysis schemes in an automated way can reduce the resources required for analysis and increase the reliability of the analysis.

Data Availability

The data used to support the findings of the study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors also would like to thank Dr. Luis Muñoz-González for providing the MATLAB scripts of the attack graphs that he used for his experiments in the paper. This work was supported by the Nuclear Safety Research Program through the Korea Foundation of Nuclear Safety (KoFONS), granted financial resources from the Nuclear Safety and Security Commission (NSSC), Republic of Korea (2106012). This work was supported by a Korea University Grant.

References

[1] J. Park, Y. Suh, and C. Park, "Implementation of cyber security for safety systems of nuclear facilities," *Progress in Nuclear Energy*, vol. 88, pp. 88–94, 2016.

[2] T. U. Code, *5195c—critical Infrastructures protection*, Legal Information Institute. Cornell Law School, Ithaca, 2001.

[3] of Homeland Security, D. D, *Homeland Security Presidential Directive 7: Critical Infrastructure Identification, Prioritization, and protection*, CISA, Rosslyn, 2003.

[4] K. Stouffer, J. Falco, and K. Scarfone, *Guide to industrial control systems (ics) security*, NIST special publication, vol. 800-82, p. 16, Gaithersburg, 2011.

[5] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand, "A survey of approaches combining safety and security for industrial control systems," *Reliability Engineering & System Safety*, vol. 139, pp. 156–178, 2015.

[6] M. Leitner, M. Frank, G. Langner et al., "Enabling exercises, education and research with a comprehensive cyber range," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 12, pp. 37–61, December 2021.

[7] L. König, S. Unger, P. Kieseberg, S. Tjoa, and J. R. C. Blockchains, "The risks of the blockchain a review on current vulnerabilities and attacks," *J. Internet Serv. Inf. Secur*, vol. 10, pp. 110–127, 2020.

[8] R. J. Turk, "Cyber incidents involving control systems," Tech. Rep, Idaho National Laboratory (INL), Falls, Idaho, 2005.

[9] M. Elshrkawey, M. Alalfi, and H. Al-Mahdi, "An enhanced intrusion detection system based on multi-layer feature reduction for probe and dos attacks," *Journal of Internet Services and Information Security (JISIS)*, vol. 11, pp. 40–57, 2021.
[10] R. S. Langner, "Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.
[11] S. I. Eyadema, *Outsource supply chain challenges and risk mitigation*, PhD thesis, Utica College, Utica, 2021.
[12] N. R. Commission, "Potential vulnerability of plant computer network to worm infection," *Nuclear Regulatory Commission Information Notice U.S. Nuclear Regulatory Commission, Office of Nuclear Reactor Regulation*, vol. 14, Washington, 2003.
[13] O. Analytica, *Us Pipeline Hack Signals Critical Infrastructure Risks*, Emerald Expert Briefings, Bingley, 2021.
[14] Bloomberg, "Colonial pipeline sued for gas crisis from ransomware attack," 2021, https://www.bloomberg.com/news/articles/2021-06-22/colonial-pipeline-is-sued-in-proposed-gas-station-class-action.
[15] Cybersecurity and Infrastructure Security Agency, "Infrastructure Security," 2021, https://www.cisa.gov/infrastructure-security.
[16] U. DoD, *Mil-std-882e, department of defense standard practice system safety*, US Department of Defense, Virginia, 2012.
[17] R. Ross, "Nist sp 800-30 rev. 1: guide for conducting risk assessments," 2012, http://www.nist.gov/customcf/get_pdf.cfm.
[18] N. Feng, H. J. Wang, and M. Li, "A security risk analysis model for information systems: causal relationships of risk factors and vulnerability propagation analysis," *Information Sciences*, vol. 256, pp. 57–73, 2014.
[19] M. Frigault and L. Wang, "Measuring network security using bayesian network-based attack graphs," in *Proceedings of the 2008 32nd Annual IEEE International Computer Software and Applications Conference*, pp. 698–703, IEEE, Turku, Finland, August 2008.
[20] M. Frigault, L. Wang, A. Singhal, and S. Jajodia, "Measuring network security using dynamic bayesian network," in *Proceedings of the 4th ACM Workshop on Quality of protection*, pp. 23–30, Alexandria, VA, USA, August 2008.
[21] M. Khosravi-Farmad and A. Ghaemi-Bafghi, "Bayesian decision network-based security risk management framework," *Journal of Network and Systems Management*, vol. 28, pp. 1–26, 2020.
[22] Y. Liu and H. Man, "Network vulnerability assessment using bayesian networks," in *Proceedings of the Data mining, intrusion detection, information assurance, and data networks security 2005*, vol. 5812, pp. 61–71, International Society for Optics and Photonics, Orlando, Florida, United Statesvol, March 2005.
[23] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 1, pp. 61–74, 2011.
[24] J. Wang, M. Neil, and N. Fenton, "A bayesian network approach for cybersecurity risk assessment implementing and extending the fair model," *Computers & Security*, vol. 89, Article ID 101659, 2020.
[25] L. Muñoz-González, D. Sgandurra, M. Barrère, and E. C. Lupu, "Exact inference techniques for the analysis of bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 2, pp. 231–244, 2017.
[26] H. Anderson, *Introduction to Nessus*, Retrieved from Symantec, Tempe, 2003.
[27] Greenbone Openvas, "Open vulnerability assessment scanner," 2021, https://www.openvas.org/.
[28] Nist, "National vulnerability database," 2021, https://nvd.nist.gov/.
[29] N map, "Nmap: the network mapper," 2021, https://nmap.org/.
[30] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system," *IEEE Security & Privacy*, vol. 4, no. 6, pp. 85–89, 2006.
[31] N. E. Institute, *Cyber Security Control Assessments, NEI 13-10 (rev.6)*, Nuclear Energy Institute, Washington, 2010.
[32] E. Iec, *61508 Series,"functional Safety of Electrical/electronic/programmable Electronic Safety-Related Systems"*, International Electro technical Commission, Geneva, 2010.
[33] J. D. Price and R. S. Anderson, "Cyber-informed engineering: the need for a new risk informed and design methodology," Idaho National Lab.(INL), Idaho Falls, ID (United States), Tech. rep, 2015.
[34] G. Lacava, A. Marotta, F. Martinelli et al., "Cybsersecurity issues in robotics," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 12, pp. 1–28, 2021.
[35] R. Wyman, "Consider the consequences: a powerful approach for reducing ics cyber risk," *Cyber Security: A Peer-Reviewed Journal*, vol. 1, no. 1, pp. 28–43, 2017.
[36] M. Krotofil, A. Cardenas, J. Larsen, and D. Gollmann, "Vulnerabilities of cyber-physical systems to stale data—determining the optimal time to launch attacks," *International journal of critical infrastructure protection*, vol. 7, no. 4, pp. 213–232, 2014.
[37] H. Orojloo and M. A. Azgomi, "A method for evaluating the consequence propagation of security attacks in cyber-physical systems," *Future Generation Computer Systems*, vol. 67, pp. 57–71, 2017.
[38] D. H. Stamatis, *Failure Mode and Effect Analysis: FMEA from Theory to Execution*, Quality Press, Plankinton, 2003.
[39] L. Xing and S. V. Amari, "Fault tree analysis," in *Handbook of Performability Engineering*, pp. 595–620, Springer, London, 2008.
[40] N. Leveson and E. SaferWorld, *Systems Thinking Applied to Safety*, MIT press, Boston, MA, USA, 2011.
[41] B. Genge, I. Kiss, and P. Haller, "A system dynamics approach for assessing the impact of cyber attacks on critical infrastructures," *International Journal of Critical Infrastructure Protection*, vol. 10, pp. 3–17, 2015.
[42] D. N. Ford, "A behavioral approach to feedback loop dominance analysis," *System Dynamics Review: The Journal of the System Dynamics Society*, vol. 15, no. 1, pp. 3–36, 1999.
[43] P. Haller and B. Genge, "Using sensitivity analysis and cross-association for the design of intrusion detection systems in industrial cyber-physical systems," *IEEE Access*, vol. 5, pp. 9336–9347, 2017.
[44] J. Huang, E. Howley, and J. Duggan, "The ford method: a sensitivity analysis approach," in *Proceedings of the 27th International Conference of the System Dynamics Society*, Albuquerque, USA, July 2009.
[45] X. Li, C. Zhou, Y.-C. Tian, N. Xiong, and Y. Qin, "Asset-based dynamic impact assessment of cyberattacks for risk analysis in industrial control systems," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 608–618, 2017.
[46] T. Murata, "Petri nets: properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
[47] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph-based network vulnerability analysis," in *Proceedings of the 9th*

ACM Conference on Computer and Communications Security, pp. 217–224, Washington, DC USA, November 2002.

[48] K. Ingols, R. Lippmann, and K. Piwowarski, "Practical attack graph generation for network defense," in *Proceedings of the 2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*, pp. 121–130, IEEE, Miami Beach, FL, USA, December 2006.

[49] R. P. Lippmann, K. W. Ingols, C. Scott et al., "Evaluating and Strengthening enterprise Network Security Using Attack Graphs," Project Report IA-2, MIT Lincoln Laboratory, Lexington, 2005.

[50] S. Noel, M. Elder, S. Jajodia, P. Kalapa, S. O'Hare, and K. Prole, "Advances in topological vulnerability analysis," in *Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security*, pp. 124–129, IEEE, Washington, DC, USA, March 2009.

[51] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 336–345, Alexandria VA USA, November 2006.

[52] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pp. 273–284, IEEE, Berkeley, CA, USA, May 2002.

[53] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia, "An attack graph-based probabilistic security metric," in *IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 283–296, Springer, Berlin, 2008, Lecture Notes in Computer Science.

[54] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri, "NuSMV: a new symbolic model verifier," in *International Conference on Computer Aided Verification*, pp. 495–499, Springer, Berlin, 1999.

[55] X. Ou, S. Govindavajhala, and A. W Appel, "Mulval: a logic-based network security analyzer," in *Proceedings of the USENIX security symposium*, vol. 8, pp. 113–128, Baltimore, MD, August 2005.

[56] S. Noel and S. Jajodia, "Managing attack graph complexity through visual hierarchical aggregation," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pp. 109–118, Washington DC USA, October 2004.

[57] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *Computers, IEEE Transactions on*, vol. 100, no. 8, pp. 677–691, 1986.

[58] R. Lippmann, K. Ingols, C. Scott et al., "Validating and restoring defense in depth using attack graphs," in *Proceedings of the MILCOM 2006-2006 IEEE Military Communications Conference (2006)*, pp. 1–10, IEEE, Washington, DC, USA, October 2006.

[59] J. Pamula, S. Jajodia, P. Ammann, and V. Swarup, "A weakest-adversary security metric for network configuration security analysis," in *Proceedings of the 2nd ACM workshop on Quality of protection*, pp. 31–38, Alexandria , VA, USA, October 2006.

[60] W. Li and R. B. Vaughn, "Cluster security research involving the modeling of network exploitations using exploitation graphs," in *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)*, vol. 2, p. 26, May 2006.

[61] S. Noel, "Metrics suite for network attack graph analytics," in *Proceedings of the 9th Annual Cyber and Information Security Research Conference*, pp. 5–8, Oak Ridge TN USA, April 2014.

[62] J. Pearl, "Bayesian networks: a model of self-activated memory for evidential reasoning," in *Proceedings of the 7th Conference of the Cognitive Science Society*, pp. 15–17, University of California, Irvine, CA, USA, 1985.

[63] L. Muñoz-González, D. Sgandurra, A. Paudice, and E. C. Lupu, "Efficient attack graph analysis through approximate inference," *ACM Transactions on Privacy and Security (TOPS)*, vol. 20, no. 3, pp. 1–30, 2017.

[64] P. Mell, K. Scarfone, and S. Romanosky, "A complete guide to the common vulnerability scoring system version 2.0," *Published by FIRST-forum of incident response and security teams*, vol. 1, p. 23, 2007.

[65] M. S. B. Mahmoud, N. Larrieu, and A. Pirovano, "A risk propagation based quantitative assessment methodology for network security-aeronautical network case study," in *Proceedings of the 2011 Conference on Network and Information Systems Security*, pp. 1–9, IEEE, La Rochelle, France, May 2011.

[66] X. Lyu, Y. Ding, and S.-H. Yang, "Safety and security risk assessment in cyber-physical systems," *IET Cyber-Physical Systems: Theory & Applications*, vol. 4, no. 3, pp. 221–232, 2019.

[67] L. Munoz Gonzalez and E. Lupu, "Bayesian attack graphs for security risk assessment," in *Proceedings of the IST-153 Workshop on Cyber Resilience*, pp. 64–77, Munich, Germany, October 2017.

[68] B. Martin and S. Christey, "Common weakness scoring system (cwss). Internet]," 2014, http://cwe.mitre.org/cwss.

[69] Pearl and Judea, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Elsevier, Amsterdam, 2014.

[70] N. A. Bonandir, N. Jamil, M. N. A. Nawawi et al., "A review of cyber security assessment (CSA) for industrial control systems (ICS) and their impact on the availability of the ICS operation," *Journal of Physics: Conference Series*, vol. 1860, no. 1, Article ID 012015, March 2021.

[71] I. O. for Standardization, *ISO/IEC 27005:2018: Information Technology–Security Techniques–Information Security Risk Management*, International Organization for Standardization, Geneva, 2018.

[72] I. S. Hwang, Y. G. Kim, W. S. Choi, and S. D. Sohn, "System and software design for the plant protection system for shin-hanul nuclear power plant units 1 and 2," in *Proceedings of the Transactions of the Korean Nuclear Society Autumn Meeting*, Gyeongju, Korea, October 2015.

[73] U. S. N. R. Commission, *Usnrc Hrtd [rev 10/08]: 0520–R325c–Ce Technology Cross Training R325c*, US NRC, Washington DC, USA, 2011, https://www.nrc.gov/docs/ML1125/ML11251A143.pdf.

[74] M. Lee, S. Song, and D. Yun, "Development and application of posafe-q plc platform," *Tech. rep.*International Atomic Energy Agency (IAEA), Vienna, 2012.

[75] B. Server, *Bayes Server-Bayeshttps://www.bayesserver.comian Network Software*, 2015, https://www.bayesserver.com.

[76] M. Eckhart, B. Brenner, A. Ekelhart, and E. Weippl, "Quantitative security risk assessment for industrial control systems: research opportunities and challenges," *Journal of Internet Services and Information Security (JISIS)*, vol. 9, no. 3, pp. 52–73, 2019.