Received: 2 June 2021 | Revised: 24 May 2022 | Accepted: 23 June 2022

DOI: 10.4218/etrij.2021-0181

ORIGINAL ARTICLE

ETRI Journal WILEY

STRIDE-based threat modeling and DREAD evaluation for the distributed control system in the oil refinery

Kyoung Ho Kim[1] | **Kyounggon Kim**[2] | **Huy Kang Kim**[3]

[1]CISO Organization, S-OIL Corporation, Seoul, Republic of Korea

[2]Center of Excellence in Cybercrime and Digital Forensics, Naif Arab University for Security Sciences, Riyadh, Kingdom of Saudi Arabia

[3]School of Cybersecurity, Korea University, Seoul, Republic of Korea

Correspondence
Huy Kang Kim, School of Cybersecurity, Korea University, Anam-ro 145, 02841, Seongbuk-gu, Seoul, Republic of Korea.
Email: cenda@korea.ac.kr

Funding information
This research was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00624, Development of Intelligence Cyber Attack and Defense Analysis Framework for Increasing Security Level of C-ITS) and Security Research Center of Naif Arab University for Security Sciences, under grant agreement No. SRC-PR2-05.

Abstract

Industrial control systems (ICSs) used to be operated in closed networks, that is, separated physically from the Internet and corporate networks, and independent protocols were used for each manufacturer. Thus, their operation was relatively safe from cyberattacks. However, with advances in recent technologies, such as big data and internet of things, companies have been trying to use data generated from the ICS environment to improve production yield and minimize process downtime. Thus, ICSs are being connected to the internet or corporate networks. These changes have increased the frequency of attacks on ICSs. Despite this increased cybersecurity risk, research on ICS security remains insufficient. In this paper, we analyze threats in detail using STRIDE threat analysis modeling and DREAD evaluation for distributed control systems, a type of ICSs, based on our work experience as cybersecurity specialists at a refinery. Furthermore, we verify the validity of threats identified using STRIDE through case studies of major ICS cybersecurity incidents: Stuxnet, BlackEnergy 3, and Triton. Finally, we present countermeasures and strategies to improve risk assessment of identified threats.

KEYWORDS

countermeasures, distributed control system (DCS), DREAD, industrial control system (ICS), network, operation technology (OT), STRIDE, threat modeling

1 | INTRODUCTION

Initially, each process control system was operated as a direct digital control system, in which a single computer handled the process data of several hundreds of loops or more. However, with an increase in the capacity of a single piece of hardware (such as computer memory), system stability has decreased due to the ripple effect in the event of system failure. Distributed control systems (DCSs), which distribute process control functions and failure risks by centralizing process monitoring and driving operation functions, have been developed to address the abovementioned problems.

A DCS is a totally integrated system that communicates between various facility elements, including hardware, and software developed by DCS manufacturers using proprietary network protocols in an isolated network, that is, one that is not connected to the internet or a corporate network.

For this reason, DCS manufacturers considered DCSs safe from cyberattacks and developed them with a focus on reliability and real-time I/O without considering

countermeasures against such attacks. However, numerous sensors and industrial control system (ICS) equipment have been supplying information to IT systems located in relatively insecure corporate networks and the internet to increase productivity and reduce costs; these actions create attack surfaces and thus expose such systems to security risks. Moreover, manufacturers have been migrating vendor-specific operation systems (OSs) and protocols to general-purpose OSs and Ethernet transmission control protocol/internet protocol (TCP/IP) networks to decrease costs. In addition, operators may delay security patches and antivirus updates in production systems due to concerns about system failures caused by such updates. Thus, systems are prone to attacks that exploit outdated malware and vulnerabilities as well as inherent vulnerabilities in the IT environment. Therefore, cybersecurity attacks on ICSs continue to increase. However, operators managing ICSs falsely believe that cybersecurity considerations are unnecessary because ICSs are still operating in closed networks [1].

Cybersecurity threats in ICS environments are increasing, such as the DarkSide ransomware attack on Colonial Pipeline as well as the Dragonfly attackers targeting energy companies with trojan. Recently, safety-focused risk management methodologies, such as HAZOP, OCTAVE, and PASTA, had been mainly implemented in ICSs. However, the focus of threat modeling research on cybersecurity remains insufficient; particularly, threat analysis studies on DCSs are limited.

In this study, we perform STRIDE-based threat modeling, a systematic threat modeling methodology, to identify threats at the key component levels of a DCS operating in a real-world oil refinery. We also validate the threats extracted from the STRIDE methodology through cybersecurity incident cases targeting ICSs, namely, the Stuxnet worm in Iran's nuclear facilities, BlackEnergy 3 in Ukraine's power centers, and the Triton attack on a chemical plant in Saudi Arabia. In addition, we evaluate the risk of the identified threats using the DREAD method and present effective risk mitigation measures based on experience in a real production environment.

The rest of this paper is organized as follows. Section 2 describes major threat modeling methodologies and cybersecurity research for ICSs. Section 3 describes the STRIDE-based threat modeling methodology and STRIDE approach. Section 4 identifies threats against a DCS using STRIDE and evaluates risks of identified threats using DREAD. Section 5 validates the threats identified using STRIDE through a comparison with threats used in actual cybersecurity cases. The final section summarizes the conclusions of this study and highlights areas of future research to further develop countermeasures and methodologies against the identified threats.

2 | BACKGROUNDS AND RELATED WORK

In Section 2.1, we define and characterize threat modeling in detail. In Section 2.2, we explain the major threat modeling methodologies for and cybersecurity research on ICSs.

2.1 | Background

Threat modeling, the first step in risk assessment, is a structured method for identifying and classifying potential threats to target systems and services. It is used to build a secure system by considering security aspects from the initial stage, that is, system development and construction. Threat modeling also identifies security requirements in the analysis phase, not the test phase, and eliminates security vulnerabilities from the design stage in advance to enhance software security while reducing development costs. A threat model contains a complete process, including identifying threats based on security objectives and an understanding of the system, assessing risks and risk priorities by considering the likelihood and impact of threats, and establishing countermeasures based on the risk assessment results.

Studies on ICS security are limited. Nonetheless, research in this field has been increasing since the emergence of the malware called Stuxnet, which manipulated centrifuges in Iran's nuclear facilities in 2010 [2]. Attacks on ICSs are not limited to monetary damage caused by system interruption, such as attacks on existing IT systems. The ripple effect and scope of accidents and human damage can expand, and the importance of research on the subject will increase accordingly. For example, a cyberattack on an oil refinery can cause a cascade of explosions throughout the facility, leading to environmental pollution and numerous casualties. In addition, the ensuing gasoline and diesel shortages can lead to a rise in consumer prices, which will affect society as a whole.

2.2 | Related work

2.2.1 | History of threat modeling

Threat modeling has been studied since 1990 along with advances in software development life cycle (SDLC) and threat modeling methodologies suitable for different environments. In general, software undergoes frequent requirement changes throughout the SDLC. Performing threat modeling during such changes consumes

additional time and incurs cost, thus putting a strain on organizations.

Sindre and Opdahl of Norway used the unified modeling language (UML), a visualization tool for system design, to employ the use case and express the abnormal behavior of systems using the opposite concept of the misuse case [3]. Amoroso introduced the concept of a threat tree, which is a transformation of the fault tree used in system safety engineering [4]. In 1998, Schneider proposed the use of attack trees to graphically model threats in "Toward a Secure System Engineering" and "Attack Trees" [5]. An attack tree visualizes, structures, and expresses an attack with the AND/OR symbol by setting the root node, which is the final target of the attacker, and the way to achieve the target, which is set as the leaf node.

In 1999, Microsoft's Loren Kohnfelder and Praerit Garg introduced the STRIDE methodology through the article "The Threats to Our Products," which includes the systematic management of various threats from the design stage of all Microsoft products [6]. Thereafter, Microsoft founder Bill Gates introduced a process of building secure applications, which eventually became the origin of Microsoft's STRIDE and Threat Analysis and Modeling (TAM) (2002) [7]. Swidersky and Snyder introduced TAM, an approach based on data flow diagrams (DFDs), to the existing STRIDE (2004) [8]. Subsequently, TAM was adopted to replace Microsoft's security development life cycle (SDL) (2011).

Recently, Microsoft launched the Threat Modeling Tool, and other organizations have developed various other threat models. OCTAVE, by Carnegie Mellon University's Software Engineering Institute, aims to manage risks for organizational information protection [9]. The US Department of Homeland Security's common vulnerability scoring system (CVSS) provides a good understanding of software vulnerabilities and assesses the resultant threats [10]. The hybrid threat modeling method (hTMM) was developed by the SEI in 2018. It combines the security quality requirements engineering method (SQUARE), security cards, and persona non grata (PNG) activities [11]. The quantitative threat modeling method (quantitative TMM) is a hybrid approach composed of attack trees, STRIDE, and CVSS applied in synergy [12]. Octotrike's open-source threat modeling methodology and tool, Trike, is a security audit framework that uses threat modeling from a risk management and defensive perspective [13]. The visual, agile, and simple threat (VAST) modeling approach is based on ThreatModeler, an automated threat modeling platform [14]. Threat models for specific targets include PASTA [15], which analyzes threats to business logic; Klocwork's threat model, which includes techniques for secure embedded software development [16]; HAZOP, which analyzes hazard and system operability [17], "Attack Trees" [5], which are diagrams that depict attacks on systems in tree form; PNG [11], which focuses on the motivation and skill of human attackers; and "Security Cards" [18], which identify unusual and complex attacks. The recently launched LINDDUN is a privacy-specific threat modeling technique for applications in social media network environments [19]. Table 1 lists the advantages and disadvantages of the 12 main threat modeling methods for identifying an appropriate approach for DCSs.

TABLE 1 Advantages and disadvantages of major threat modeling methodologies

Threat modeling methodology [20]	Maturity	Focus/ perspective	DFD-based	Mitigation	Automation	Consistent results
STRIDE [8, 21]	High	Defender	O	O	O	X
PASTA [15]	High	Risk	O	O	X	X
LINDDUN [19]	High	Privacy concerns	O	O	X	X
CVSS [10]	High	Scoring	X	X	O	O
Attack Trees [5]	High	Attacker	X	X	X	O
PnG [11]	Medium	Attacker	X	X	X	O
Security Cards [18]	Medium	Unusual attacks	X	X	X	X
hTMM [11]	Low	Attacker/defender	O	X	X	O
Quantitative TMM [12]	Low	Attacker/defender	O	X	X	O
Trike [13]	Low	Risk	O	O	X	X
VAST [14]	High	Attacker	O	O	O	O
OCTAVE [9]	Medium	Operational risks	X	O	X	O

Abbreviations: CVSS, common vulnerability scoring system; DFD, data flow diagram; hTMM, hybrid threat modeling method; VAST, visual, agile, and simple threat.

2.2.2 | Cybersecurity research for ICSs

In response to the rise in threats to and incidents in ICSs, interest in ICS security is accordingly increasing in national agencies and related research. However, cybersecurity researchers have had limited opportunities to experience ICSs directly or indirectly, so studies on ICS security remain lacking.

The ICS-CERT in the United States aims to strengthen frequently used vulnerabilities, security trends, and research for critical infrastructure protection [22]. The National Institute of Standards and Technology (NIST; US) develops and provides a cybersecurity framework for enhancing infrastructure cybersecurity. It provides guidelines for strengthening security, which are evaluated and used for various industries operating ICSs [23, 24]. The Australian Cyber Security Center publishes a report on threats to and incidents in ICSs [25].

In accordance with the ICT Protection Act, Korea designates ICT infrastructure and enforces vulnerability inspection and risk assessment activities for each critical infrastructure. Khan et al. proposed a threat modeling method for cyber-physical systems (CPSs) using the STRIDE model [26]. Kim et al. proposed STRIDE-based threat modeling for the assessment of smart home systems [27, 28]. Yampolskiy et al. evaluated the feasibility of conducting a systematic analysis of cyberattacks on CPSs as a DFD-based approach [29]. Ralstona et al. presented a cybersecurity risk assessment method for supervisory control and data acquisition (SCADA) and DCS networks [30]. Cherdantseva et al. proposed a cybersecurity risk assessment technique for SCADA systems [31].

3 | STRIDE METHODOLOGY FOR DCSS

Threat modeling is a security analysis approach that identifies and classifies potential threats to analytical targets to determine critical security risks. The goal of threat modeling is to mitigate risk to an acceptable level. Safety is an important factor in oil refineries and petrochemical factories, and risk assessment focused on safety has been conducted using various methodologies, such as HAZOP and OCTAVE.

In this paper, we focus on identifying security threats against DCSs, which are the core of oil refinery operation, and propose measures to mitigate the identified risks with limited resources. A DCS is generally isolated from IT networks and interacts with limited components within one or an operation process associated with the process. Considering these DCS characteristics, we use the STRIDE methodology, which is a mature and optimal approach, to classify trust boundaries and identify cyber threats to each system component and their interaction from the defender's point of view. The DREAD methodology is then adopted to prioritize the identified risks and derive a risk-based remediation plan.

In the 1980s, the Johnson Space Center (US) coined the term CIA triad, which stands for confidentiality, integrity, and availability [32]. STRIDE categorizes threats corresponding to cybersecurity goals by adding three elements to the CIA triad: authentication, nonrepudiation, and authorization. STRIDE is named after these six threats and can help identify applications' vulnerabilities and potential attacks. Table 2 shows the six threats in STRIDE and the security attributes associated with each.

3.1 | STRIDE-based threat modeling methodology

Due to the lack of a standard methodology, we propose seven high-level steps (Figure 1) for applying STRIDE threat modeling to a DCS. The first step is to identify assets and security objectives. The second step is to create an architecture overview. The created architecture helps to understand the purpose of the target system, the

TABLE 2 Correlation between six threats in STRIDE and security properties

Threat	Security property	Threat definition
Spoofing	Authentication	Impersonate something or someone else
Tampering	Integrity	Modify data or code
Repudiation	Nonrepudiation	Claim to have not performed an action
Information disclosure	Confidentiality	Expose information to someone not authorized see it
Denial of service	Availability	Deny or degrade service to users
Elevation of privilege	Authorization	Gain capabilities without proper authorization

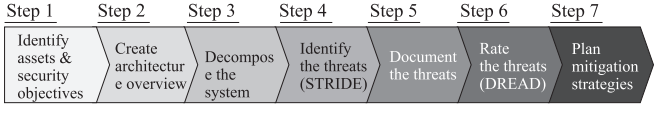

FIGURE 1 STRIDE-based threat modeling methodology

system user, the data contained in the system, and the relationships between each component. The third step is to decompose the target system into its logical components using a DFD, which helps to visualize the functionalities and communication between components within or external to the DCS. A DFD uses four standard symbols: (i) external entity (EE) generates data inputs and use outputs. (ii) Data store (DS) stores data temporarily or permanently. (iii) Process (P) obtains data inputs and generates outputs. (iv) Data flow (DF) indicates data movement between EE, DS, and P. (v) Trust boundary (TS) indicates changes in privilege levels and separates trustworthy and untrustworthy elements. Each DFD element type is susceptible to only a few or all STRIDE threats, as shown in Table 3. The fourth step is to identify threats based on each system component and functionality using the STRIDE methodology. The fifth step is to categorize and write these threats. The sixth step is to rate these threats using a risk assessment model, such as DREAD. This rate helps prioritize the risk mitigation actions in the next step, which is to plan mitigation strategies.

3.2 | STRIDE approach

STRIDE uses a DFD for effective modeling. Microsoft has proposed the STRIDE-per-element and STRIDE-per-interaction methods [33]. The former is a complex method of analyzing STRIDE for each DFD component; it utilizes the security properties associated with a specific threat, as shown in Table 3. However, this method does not identify threats that are difficult to find in a DFD, and these threats emerge through the interactions between system components. The STRIDE-per-interaction method enumerates threats against system interactions by considering tuples (origin, destination, and interaction). The STRIDE-per-interaction method is relatively easy to perform than the STRIDE-per-element approach and can sufficiently protect a system at a general level, given that cyberattacks normally involve malicious interactions between system components.

TABLE 3 DFD elements to STRIDE threats

DFD elements	S	T	R	I	D	E
External entity (EE)	X		X			
Process (P)	X	X	X	X	X	X
Data flow (DF)		X		X	X	
Data store (DS)		X		X	X	

Abbreviation: DFD, data flow diagram.

4 | SYSTEM DESCRIPTION: DCS OPERATION

A DCS centrally collects information from multiple sensors, analyzes the collected data, and sends necessary commands to actuators to adjust the values. This system aims to maintain an optimal operating environment. We explore possible threats by limiting the scope of DCS operation into one production process.

Threat modeling for DCSs consists of seven major steps. In this study, the scope of the analysis target is clearly defined, function and data flow of the analysis target are determined, threats are identified through STRIDE threat modeling, and risk is calculated using the DREAD method. Finally, we propose possible countermeasures that can be applied to DCSs based on the calculated risk criticality.

4.1 | Identify assets and security objectives

Step 1 is to identify the assets and security objectives of the DCS, as shown in Figure 1. A refinery consists of continuous processes, and a failure in one process can affect the next process and consequently the entire production process. Its DCS is essential for its continuous operation. The basic components of DCSs in refineries are similar, regardless of the production process. Therefore, we limit our analysis to a DCS in one process (Figure 2).

The security objective of a DCS is to operate safely without an unplanned shutdown, which is achieved by guaranteeing availability to transfer commands or responses within a set time period. Integrity must be ensured so that accurate values can be transmitted to field devices without tampering. Finally, confidentiality must be guaranteed so that the process operation and product recipe know-how are not accessed by unauthorized persons, as shown in Figure 3. The DCS is important for the principles of the CIA triad in the following order: availability, integrity, and confidentiality.

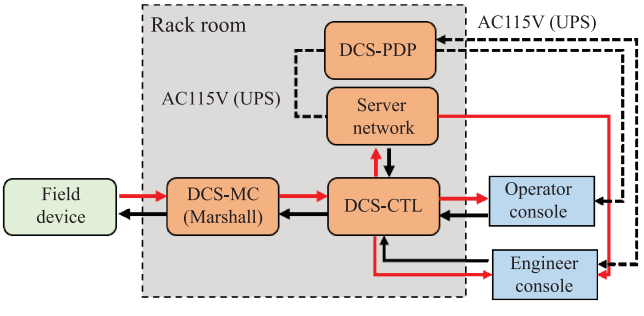

FIGURE 2 Distributed control system (DCS) operation

4.2 | Create architecture overview

Step 2 is to create an architecture overview, as shown in Figure 1. According to Figure 4, the DCS in one production process operates with the DCS controller, DCS servers, engineering workstation (EWS), and operator workstation (OWS), which are essential components. It also works with an active directory (AD) for integrated authentication and account management, GPS server for time synchronization within a closed network, OLE for process control servers for linking information with other processes or products from other manufacturers, and historian for trend analysis of process/operation information. Certain sensors and actuators generate analog and digital signals through serial communication, such as I/O modules that centralize information through RS-232, RS-422, and RS-485 from sensors. In some cases, it directly receives from a process logic controller (PLC). In general, refineries are engineered to configure safety instrument systems (SISs) with DCSs to perform specific control functions to fail-safe operation against unacceptable or dangerous conditions. However, SIS components and interactions, such as the SIS controller, EWS for SIS, and DCS-ICS communication, are not included in the scope of the threat analysis in this study. Itemizing the important characteristics and components of the DCS is helpful in identifying threats using STRIDE (step 4), as shown in Figure 1.

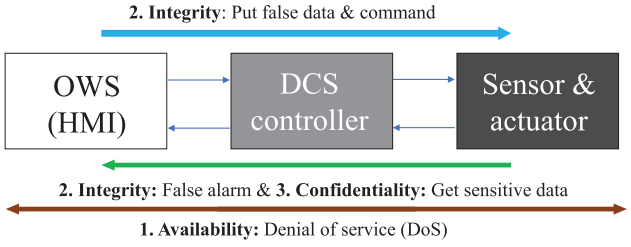

FIGURE 3 Major security concerns of distributed control systems (DCSs)

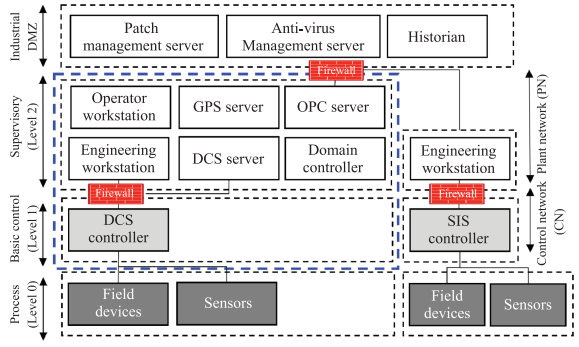

FIGURE 4 Proposed distributed control system (DCS) and safety instrument system (SIS) architectures

S-OIL: Company Use Only

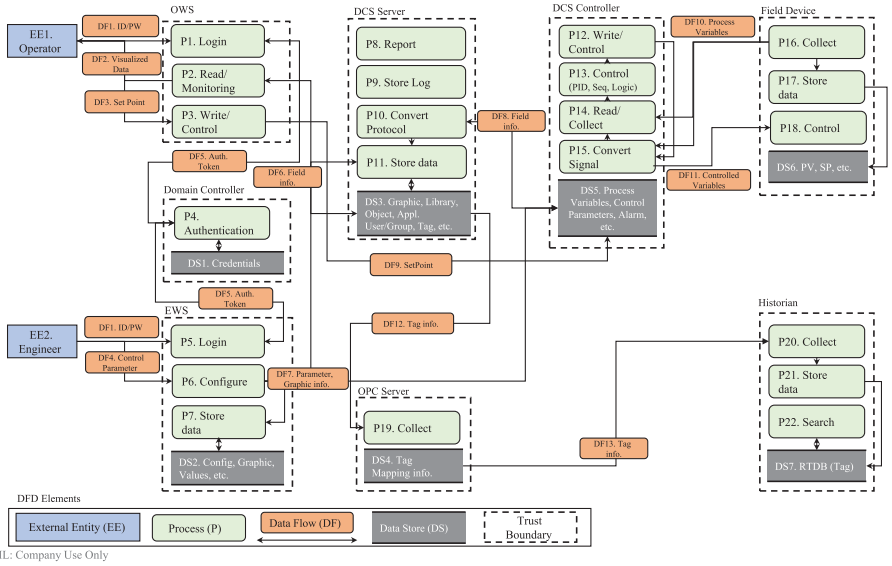

FIGURE 5 Data flow diagrams (DFDs) of distributed control system (DCS) in oil refinery

The roles and significant functions of the essential components of DCSs are as follows: (i) OWS is used by an operator to monitor and adjust set-point values. (ii) EWS is used by an engineer to manage controller setting information. (iii) DCS server is used to provide screen values and user profile information to the OWS. (iv) AD plays a role in centrally managing user accounts, authentication, and group policies. According to the Purdue Enterprise Reference Architecture, the de facto standard of ICS architectures, a DCS is in levels 1 and 2. It communicates between them using different protocols via the DCS server, which has communication gateway functions. The OWS sends set points to and requests data directly from the DCS controller or indirectly through the DCS server. The DCS controller controls actuators based on control commands and feedback from sensors.

TABLE 4 Threat modeling using STRIDE-per-element methodology

STRIDE	DFD elements
Spoofing	EE-1, EE-2, P-1, P-2, P-3, P-4, P-5, P-6, P-7, P-12, P-14, P-18, P-19, P-20, P-21, P-22
Tampering	P-2, P-3, P-6, P-7, P-8, P-9, P-10, P-11, P-12, P-13, P-14, P-15, P-16, P-17, P-18, P-19, P-20, P-21, DF-1, DF-2, DF-3, DF-4, DF-5, DF-6, DF-7, DF-8, DF-9, DF-10, DF-11, DF-12, DF-13, DS-1, DS-2, DS-3, DS-4, DS-5, DS-6, DS-7
Repudiation	EE-1, EE-2, P-1, P-2, P-3, P-4, P-5, P-6, P-7, P-11, P-14, P-15, P-16, P-18, DS-2, DS-6
Information disclosure	P-1, P-2, P-4, P-5, P-8, P-14, P-16, P-18, P-19, P-21, P-22, DF-1, DF-2, DF-3, DF-4, DF-5, DF-6, DF-7, DF-8, DF-9, DF-10, DF-11, DF-12, DF-13, DS-1, DS-2, DS-3, DS-4, DS-5, DS-6, DS-7
Denial of service	P-1, P-2, P-3, P-4, P-5, P-6, P-7, P-10, P-14, P-16, P-17, P-18, P-19, P-20, P-21, P-22, DF-1, DF-5, DF-9, DF-10, DF-12, DF-13
Elevation of privilege	P-1, P-2, P-3, P-4, P-5, P-6, P-7, P-8, P-14, P-16

4.3 | Decompose system

Step 3 is to decompose the system into its components, as shown in Figure 1. A detailed understanding of the DCS mechanism makes it easier for users to uncover more relevant, detailed threats. First, we identify the DCS components, which potential attackers might be interested in. Second, we draw a DFD for each system component, as shown in Figure 5, and map how components communicate to visualize how data flows through the system. Finally, we identify the DCS trust boundaries to focus the analysis on the areas of concern.

4.4 | Identify threats using STRIDE

Step 4 is threat analysis. We use the STRIDE threat modeling methodology to identify the weaknesses associated with threats by focusing on areas where mistakes are most often made. We use details from steps 2 (creation of architecture overview) and 3 (system decomposition) to identify threats relevant to the DCS scenario and context. An attack library is needed to perform complete threat analysis using DFDs. Information that can be utilized as an attack library includes research published in papers and conferences, MITRE's common attack pattern enumeration and classification, common vulnerabilities and exposures (CVE), ICS-CERT reports, and NIST SP 800-30. In this paper, we select three widely documented cybersecurity incidents, namely, Stuxnet, BlackEnergy 3, and Triton, to identify harmful threats used in real-world accidents. The results are summarized in Table 4 and briefly explained in the following.

4.4.1 | Spoofing

DCS is utilized by operators, who operate production processes using the OWS, and by engineers, who work on engineering processes through the EWS (EE-1 and EE-2 in Figure 5).

An attacker poses as an authenticated operator (P1) by manipulating the set-point value (P3) through the controller (P12) to raise the boiler temperature (connected to the actuator) to the highest value (P15). The resulting abnormal increase in temperature will affect the output quality. In addition, it can lead to an explosion if the high temperature persists. Slight differences depend on the DCS manufacturer. However, the controller can operate the set points directly or indirectly via the OWS. Thus, the attacker tries to gain control of the OWS.

In general, the engineer sets the upper and lower limits of an instrument range to prevent any erroneous input of abnormal values by an operator. An attacker posing as an authenticated engineer (P5) can modify the upper or lower limit of the instrument range and graphic information, which is viewed by operators (P6). This attack can provide misinformation to the operator, and any resulting error of the operator can lead to major accidents, such as shutdown, charge down, and explosion.

Nonetheless, in an actual refinery, accidents caused by DCS malfunction and operator mistakes are prevented through a valve controlled by the SIS at points that may cause explosions or danger. The malware Triton, identified in 2017, directly targeted an SIS, taking into account the engineering characteristics and facilities of the petrochemical plant.

4.4.2 | Tampering

Tampering is a highly dangerous attack, but it is difficult to detect as it can easily manipulate a controller to perform unsafe actions. So far, the DCS protocol has not been disclosed to the public. This difficulty of understanding the protocol makes it difficult to detect manipulated values.

First, we examine tampering attacks on the write action. Similar to DF-3 and DF-9, certain attack methods modify and send set points from the OWS. Another way is to tamper with the final set-point value that the controller sends to the actuator (DF-11). From the attacker's point of view, access to the OWS is considerably better than accessing the controller. However, the probability of detection by security solutions is relatively high when the set-point value is tampered from the OWS. Next, we analyze tampering attacks on the read action. If an attacker manipulates visualized graphic and field information (DF-2 and DF-6), an operator can misjudge by monitoring the tampered information (DF-10) and lead to faulty process operation. For example, a normal production process can be recognized as abnormal, and the set point can be adjusted. In contrast, as seen in Stuxnet and BlackEnergy 3, tampered information can be transmitted to the operators to make an abnormal production process appear normal.

4.4.3 | Repudiation

Nonrepudiation can be resolved by storing logs in the OS and applications of a DCS. The security log of a DCS OS is not large, averaging between 30 and 50 MB per day, although it slightly differs between products of DCS makers. Logs stored in a DCS are utilized for detection and response in cybersecurity incidents rather than cyberattack prevention. Generally, a DCS stores process operation logs. However, most refineries are still operating without system security log settings due to concerns about delays in actual operation. Consequently, the causes of cybersecurity incidents are difficult to analyze. Even with stored security logs, most refineries are not configured to analyze real-time logs through links with security information and event management. Therefore, such logs are merely used in incident response.

4.4.4 | Information disclosure

Most refinery technologies are purchased from licensors. Thus, information disclosure by attacks in refinery environments is not risky themselves directly. However, the credential information stored in the AD (DS1), the process operation information transmitted to the historian (DS7), and optimal operation information (DS2, DS3, and DS5) must be protected from leakage. In particular, exposed system configuration (DS5) and credential information (DS1) can be used by attackers in more sophisticated attacks. Processes are especially vulnerable to information attacks (P4, P7, P11, P12, and P19). Nonetheless, data cannot be leaked through sniffing in a DCS network, according to a test in a production environment, because DCSs use vendor-specific protocols.

4.4.5 | Denial of service (DoS)

DoS attacks interrupt or interfere with regular operations by generating excessive traffic in a DCS network or exhausting system resources by calling a specific DCS process. Most of the components that constitute a DCS are vulnerable to DoS attacks. Hence, due to DoS attacks, commands cannot be transmitted to the controller on time, thus delaying actuator operation or causing problems where operation information is not normally displayed on the OWS dashboard.

When we conduct a DoS attack experiment in a factory acceptance test environment, the stage before a DCS is installed in a production environment, ARP-, IP-, and TCP/UDP-based DoS attacks succeed because the DCS components are operated using TCP/IP-based protocols. In addition, security solutions capable of detecting DoS attacks are not installed in DCS networks in most cases. Therefore, in reality, DoS attacks have a significant impact on the normal operation of a DCS, and DoS attack detection and response are challenging.

4.4.6 | Elevation of privilege

Elevation of privilege means a user with normal authority manages to perform an action that requires privilege. Most DCSs are managed based on ADs for integrated management of distributed systems. Because an AD stores the credentials of all components (DS1) constituting the DCS, an attacker aims to obtain the AD administrator's authority. After the attackers steal the administrator's authority, they can create a new account, remotely access other components and manipulate information, or destroy the system.

Table 5 summarizes threat analysis using the STRIDE-per-interaction methodology considering all interactions occurring in one production unit of a refinery. The table shows three main types of interactions: command messages (P3 and P6), data messages (P2 and P10), and authentication messages (P1 and P4). Different

TABLE 5 Threat modeling using STRIDE-per-interaction methodology

Interaction	S	T	R	I	D	E
EE1 to P1: Login info	X	X	X	X	X	X
P2 to EE1: Operation info	X	X		X		
EE1 to P3: SetPoint	X	X	X	X	X	X
EE2 to P6: Parameter	X	X		X	X	
P4 to P1: Auth. Token	X	X	X	X	X	X
DS3 to P2: Operation info	X	X		X	X	
P6 to P7 or P6 to DS3 or P6 to DS5: Parameter, Graphic	X	X		X	X	
DS5 to P10: Operation info.	X	X		X	X	
P3 to DS5: SetPoint	X	X	X	X	X	
P6 to P15: Controlled Value	X	X		X	X	X
P16 to P14: Controlled Value	X	X		X	X	X
D12 to P18: SetPoint	X	X		X	X	X
DS3 to P19: Tag info.	X	X		X	X	
DS4 to P20: Tag info.	X	X		X	X	

interactions are vulnerable to various STRIDE threats. For example, command and authentication messages are vulnerable to all STRIDE threats, whereas data messages are vulnerable only to spoofing, tampering, and DoS.

4.5 | Document threats

Documentation of threats (Step 5 in Figure 1) aims to identify as many potential threats as possible without omitting elements. An attack library is needed to perform a complete analysis of the threats identified using the DFDs in Section 4.4. However, it is difficult to know what and how an attacker is compromising just with the identified threats. Therefore, an attack scenario must be derived to know the attacker's final goal.

To this end, we use an attack tree. The goal of an attack tree is set to be refinery operation shutdown (Figure 6). The threats used in actual ICS incidents (Section 5) are utilized as attack libraries and created based on them.

4.6 | Rate threats

The rating of threats (Step 6 in Figure 1) aims to prioritize threats that require quick action due to high risks. When various threats are identified from an attack tree, it is difficult to remove all of them simultaneously due to the limited resources available for threat management. Risk assessment should be performed to determine the threats to be managed first. A common risk assessment method involves evaluating the likelihood and impact of threats. In this research, we use the DREAD model to evaluate risk [34].

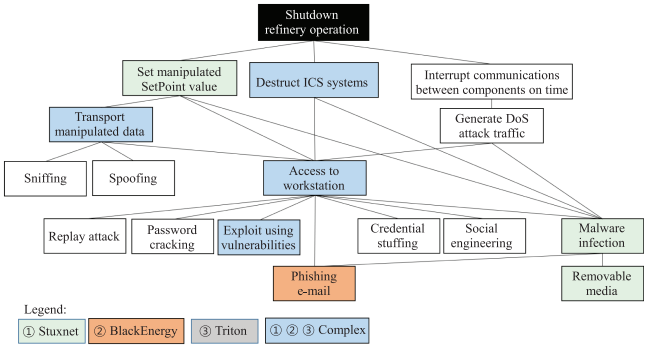

FIGURE 6 Attack Tree for distributed control systems (DCS)

DREAD is a risk rating method, and its name stands for five evaluation categories: damage, reproducibility, exploitability, affected users, and discoverability. Damage is the degree of damage in the dangerous scenario. Reproducibility refers to the reproducibility of the risk scenario. Exploitability indicates the possibility of attacking code attacks in the risk scenario. Affected users denote the extent to which people are affected by the risk scenario. Discoverability represents the degree of discoverability of the risk scenarios. Each category is rated 1 to 3 points (3 for high severity, 2 for medium severity, and 1 for low severity), with 15 points indicating the worst risk (Table 6).

TABLE 6 DREAD threat rating results

Threats	D	R	E	A	D	Sum
Phishing email	3	3	3	3	3	15
Removable media	3	3	3	3	3	15
Malware infection	3	3	3	3	3	15
Exploit using vulnerabilities	3	2	2	3	3	13
Denial of Service	3	3	2	3	1	12
Acquire privilege on the WKS	3	1	1	2	2	9
Transport manipulated data	3	1	1	1	1	7
Destruct DCS controller	3	1	1	1	1	7

Abbreviation: DCS, distributed control system.

4.7 | Plan mitigation strategies

The final step (Step 7 in Figure 1) is to plan mitigation strategies. Risk mitigation involves selecting items that require urgent action (that is, the high-risk threats identified in Step 6) and prioritizing activities considering the cost and difficulty of the measures needed to mitigate the risk and conduct, if any, compliance requirement.

While a general IT system has a life cycle of five to seven years, an ICS usually lasts 15–20 years. In outdated ICSs, security patches are no longer provided after vendors' technical support ends, and security solutions and patches are difficult to install due to insufficient system resources. It is significantly challenging to ensure availability by securing ICSs under these limitations. First, considering the limitations and characteristics of ICSs, standard security management frameworks suitable for refinery DCS environments, such as the NIST cybersecurity framework and the oil and gas cybersecurity capability maturity model (ONG C2M2) [23, 35], are applied to improve security maturity at the refinery level. Second, common cyber threats, such as email phishing and USB-based malicious code infection, can be addressed by applying whitelist-based application control with minimal resource usage instead of updating antivirus engines or installing security patches. Third, we establish a detection-oriented passive security system that increases visibility by identifying assets in the DCS network and detecting abnormal activities, rather than an active security system, which runs the risk of process interruption with blocking due to false positives. Fourth, next-generation firewall is installed for network perimeter security to maintain the advantages of network isolation. In addition, an intrusion detection system (IDS) can be managed and monitored in an integrated manner to understand industrial control protocols and strengthen security for all DCS components. Finally, the DCSs currently in production are operated securely by minimizing changes. In the construction of new DCSs or replacement of obsolete systems, it is necessary to analyze the security requirements from the requirement analysis stage and to design and configure the system safely.

5 | CASE STUDY

Cyberattacks targeting ICSs are becoming increasingly frequent, intelligent, and sophisticated. In this study, we study three significant cybersecurity incidents targeting ICSs and identify the threats utilized in each attack.

5.1 | Stuxnet

Stuxnet is the first case to make people aware about the need for ICS security. Discovered in 2010, this malware significantly damaged the centrifuges used for separating nuclear material in Iranian facilities and was the first to be infected via USB media. Unlike most malware, Stuxnet targeted vulnerable ICSs running WinCC/Step 7 control software; this software is usually adopted to program specific Siemens PLCs, which are used widely in factories, refineries, and power plants. Stuxnet replaces an original .dll file with a malicious .dll file. This malicious file monitors and intercepts all communication between PCs and PLCs. Stuxnet injects its own code into PLCs in a manner undetectable by operators [36].

5.2 | BlackEnergy 3

BlackEnergy 3 is the first known successful cyberattack on a power grid. On December 23, 2015, attackers compromised three energy distribution companies in Ukraine. Conducted within minutes, the cyberattacks targeted 30 substations (seven 110 kv substations and 23 35 kv substations), resulting in power outages affecting approximately 225 000 customers for a few hours.

The attack path and technologies of BlackEnergy 3 were as follows: (i) The attacker accessed the business network through spear phishing. (ii) Through reconnaissance activities in the business network, the attacker obtained credentials and VPNs connected to the ICS network and successfully accessed the ICS network. (iii) The attacker sent commands directly from a remote station similar to an operator HMI using existing remote access tools. (iv) The attacker uploaded malicious firmware to serial-to-Ethernet gateway devices. (v) The attacker executed a modified KillDisk to erase the master boot record and system logs. (vi) To delay recovery, the attacker shut down the uninterrupted power systems and launched

DoS attacks on the call centers to block incoming calls from customers [37, 38].

5.3 | Triton

Triton is the first cyberattack targeting an SIS. In December 2017, a petrochemical plant in Saudi Arabia encountered an SIS shutdown caused by the malware Triton. The SIS is the last line of automated safety defense in an industrial facility, and it is designed to prevent equipment failure and catastrophic incidents, such as explosion or fire. An attack targeting an SIS can be directly linked to human life. Thus, the significance of the Triton attack case differentiates it from existing attack cases.

The attack path and technologies of Triton were as follows: (i) The attack began with a breach of the IT network. (ii) The attacker accessed the OT network using a misconfiguration of a firewall located between the IT and OT networks. (iii) The attacker injected the malware into the EWS of the SIS, which was operated in an isolated network. (iv) The attacker obtained SIS information and reprogrammed the Triconex SIS controller from Schneider Electric. The final stage of the attack failed, and the attack was unsuccessful [39].

6 | CONCLUSION

This study examines threat modeling for ICSs, which are becoming increasingly important as a primary target of cyber warfare. Attackers are particularly targeting DCSs, which are vital for refinery operations. DCSs have been operated safely in isolated networks for a long time. However, the increase in connectivity between DCSs and the internet and corporate networks due to recent environmental changes has worsened cybersecurity threats to DCSs. In addition, successful attacks cause chain effects leading to serious social disruption, becoming the target of more attackers.

To respond to these threats and increasing threat sources effectively, we leverage a STRIDE-and-DREAD-based threat modeling methodology for DCSs in refineries to proactively detect and improve identified threats and respond quickly to incidents. We decompose a DCS into its major components based on our work experience as cybersecurity specialists in a refinery. Then, we derive threats arising from the interactions between components and threats arising from elements. Using case studies of attacks targeting ICSs, we verify the effectiveness of threat identification by the STRIDE methodology. Not all identified threats can be eliminated, as enterprises have limited resources. Therefore, in this study, we use the DREAD methodology to evaluate the risk levels of threats and follow this evaluation in formulating a strategic risk mitigation plan. Risk evaluation allows companies to make cost-effective security investments. This research recommends effective risk mitigation measures based on experience in real production environments.

Enterprises have difficulties performing active defense and response for ICSs due to the priority of ensuring availability. This problem requires a passive method of monitoring-oriented response, but it is also challenging to detect attacks in this scenario due to the proprietary network protocols of vendors. However, the variability in network usage used by an ICS does not change significantly compared with the network usage variability of an IT system. Because ML and AI technologies can distinguish abnormality and normality without understanding protocol specifications, they can detect specific attacks and analyze abnormal behavior. Therefore, an IDS model using ML and AI should be an effective research prospect for immediate application to the production environment.

CONFLICT OF INTEREST

The authors declare that there are no conflicts of interest.

ORCID

Kyoung Ho Kim https://orcid.org/0000-0003-1330-0013
Kyounggon Kim https://orcid.org/0000-0002-5675-4253

REFERENCES

1. Fortinet, *2020 state of operational technology and cybersecurity report*, 2020. Available from: https://www.fortinet.com/content/dam/fortinet/assets/analyst-reports/report-state-of-operational-technology.pdf [last accessed May 2021].
2. C. Stevens, *Assembling cybersecurity: the politics and materiality of technical malware reports and the case of stuxnet*, Contemp. Sec. Policy **41** (2020), no. 1, 129–152.
3. G. Sindre and A. L. Opdahl, *Eliciting security requirements with misuse cases*, Require. Eng. **10** (2005), no. 1, 34–44.
4. E. G. Amoroso, *Fundamentals of computer security technology*, Prentice-Hall, Inc., 1994.
5. B. Schneier, *Attack trees*, Dr. Dobb's J. **24** (1999), no. 12, 21–29.
6. L. Kohnfelder and P. Garg, *The threats to our products*, Microsoft Interf. Microsoft Corp. **33** (1999).
7. B. Gates, *Trustworthy computing*, 2002. Available from: https://www.wired.com/2002/01/bill-gates-trustworthy-computing/ [last accessed May 2021].
8. F. Swiderski and W. Snyder, *Threat modeling*, Microsoft Press, 2004.
9. C. Alberts, A. Dorofee, J. Stevens, and C. Woody, *Introduction to the octave approach*, Tech. report. Carnegie-Mellon Univ. Pittsburgh Software Engineering Inst, 2003.
10. M. Schiffman, A. Wright, D. Ahmad, and G. Eschelbeck, *The common vulnerability scoring system*, National Infrastructure Advisory Council, Vulnerability Disclosure Working Group, Vulnerability Scoring Subgroup, 2004.

11. N. R. Mead, F. Shull, K. Vemuru, and O. Villadsen, *A hybrid threat modeling method*, Technical Report-CMU/SEI-2018-TN-002, Carnegie MellonUniversity-Software Engineering Institute, 2018.
12. B. Potteiger, G. Martins, and X. Koutsoukos, *Software and attack centric integrated threat modeling for quantitative risk assessment*, (Proceedings of the Symposium and Bootcamp on the Science of Security, New York, NY, USA), 2016, pp. 99–108.
13. P. Saitta, B. Larcom, and M. Eddington, *Trike v. 1 methodology document* [draft], 2005. URL: http://dymaxion.org/trike/Trikev1MethodologyDocumentdraftpdf
14. B. Beyst, *Which threat modeling method. threatmodeler*, Apr. 2016. Available from: https://threatmodeler.com/threat-modeling-methodologies-vast/ [last accessed May 2022].
15. T. UcedaVelez and M. M. Morana, *Risk centric threat modeling*, Wiley Online Library, 2015.
16. klockwork, *Threat modeling for secure embedded software*, 2011.
17. T. A. Kletz, *Hazop and hazan: Identifying and assessing process industry hazards*, IChemE, 1999.
18. T. Denning, B. Friedman, and T. Kohno, *Security and privacy threat discovery cards*, 2013. Available from: http://securitycards.cs.washington.edu/assets/security-cards-deck-with-croplines.pdf [last accessed May 2022].
19. K. Wuyts and W. Joosen, *Linddun privacy threat modeling: A tutorial, Technical Report (CW Reports)*, vol. **C685**, (Department of Computer Science, KU Leuven), 2015.
20. N. Shevchenko, B. R. Frye, and C. Woody, *Threat modeling for cyber-physical system-of-systems: Methods evaluation*. Tech. report. Carnegie Mellon University Software Engineering Institute Pittsburgh United, 2018.
21. E. A. AbuEmera, H. A. ElZouka, and A. A. Saad, *Security framework for identifying threats in smart manufacturing systems using stride approach*, (2nd International Conference on Consumer Electronics and Computer Engineering, Guangzhou, China), 2022, pp. 605–612.
22. Cybersecurity & Infrastructure Security Agency (CISA), Ics-cert website. Available from: https://us-cert.cisa.gov/ics [last accessed May 2021].
23. NIST, *Nist cybersecurity framework*, 2017. Available from: https://www.nist.gov/cyberframework [last accessed May 2021].
24. K. Stouffer, J. Falco, and K. Scarfone, *Sp 800-82 rev. 2*, Guide Industr. Contr. Syst. (ICS) Sec. NIST **2** (2015), no. 3, 5.
25. Australian Cyber Security Centre (ACSC), *Cert australia*. Available from: https://www.cyber.gov.au/ [last accessed May 2021].
26. R. Khan, K. McLaughlin, D. Laverty, and S. Sezer, *Stride-based threat modeling for cyber-physical systems*, (IEEE PES Innovative Smart Grid Technologies Conference Europe, Turin, Italy), 2017, pp. 1–6.
27. K. K. Gon and K. S. Hoon, *Using threat modeling for risk analysis of smarthome*, (Proceedings of Symposium of the Korean Institute of Communications and Information Sciences), 2015, pp. 378–379.
28. K. Kim, K. Cho, J. Lim, Y. H. Jung, M. S. Sung, S. B. Kim, and H. K. Kim, *What's your protocol: Vulnerabilities and security threats related to z-wave protocol*, Pervasive Mobile Comput. **66** (2020), 101211.
29. M. Yampolskiy, P. Horvath, X. D. Koutsoukos, Y. Xue, and J. Sztipanovits, *Systematic analysis of cyber-attacks on cps-evaluating applicability of dfd-based approach*, (5th International Symposium on Resilient Control Systems, Salt Lake, UT, USA), 2012, pp. 55–62.
30. PAS Ralston, J. H. Graham, and J. L. Hieb, *Cyber security risk assessment for scada and dcs networks*, ISA Trans. **46** (2007), no. 4, 583–594.
31. Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, and K. Stoddart, *A review of cyber security risk assessment methods for scada systems*, Comput. Secur. **56** (2016), 1–27.
32. Y. Cherdantseva and J. Hilton, *A reference model of information assurance & security*, (International Conference on Availability, Reliability and Security), 2013, pp. 546–555.
33. A. Shostack, *Threat modeling: designing for security*, John Wiley & Sons, 2014.
34. A. Shostack, *Experiences threat modeling at microsoft*, MODSEC@ MoDELS **2008** (2008), 35.
35. P. D. Curtis and N. Mehravari, *Evaluating and improving cybersecurity capabilities of the energy critical infrastructure*, (IEEE International Symposium on Technologies for Homeland Security, Waltham, MA, USA), 2015, pp. 1–6.
36. R. Langner, *Stuxnet: dissecting a cyberwarfare weapon*, IEEE Sec. Privacy **9** (2011), no. 3, 49–51.
37. D. U. Case, *Analysis of the cyber attack on the ukrainian power grid*, Electr. Inform. Shar. Anal. Center (E-ISAC) **388** (2016), 1–29.
38. M. Geiger, J. Bauer, M. Masuch, and J. Franke, *An analysis of black energy 3, crashoverride, and trisis, three malware approaches targeting operational technology systems*, (25th IEEE International Conference on Emerging Technologies and Factory Automation, Vienna, Austria), 2020, pp. 1537–1543.
39. A. Di Pinto, Y. Dragoni, and A. Carcano, *TRITON: the first ICS cyber attack on safety instrument systems*, (Proc. Black Hat USA), vol. **2018**, 2018, pp. 1–26.

AUTHOR BIOGRAPHIES

Kyoung Ho Kim received the BS degree in computer science from Hansung University, Seoul in 2002, the MS degree in information and technology from Sogang University, Seoul, in 2008, and the PhD degree in information security from the School of Cybersecurity, Korea University, Seoul, South Korea, in 2022. He is currently a cybersecurity architect over 15 years at S-OIL corporations and has specialties in IT and OT cybersecurity. Before joining the S-OIL Corp., he was the cybersecurity specialist in KT Hitel and cybersecurity consultant in A3 Security Consulting, which is the first information security consulting company in South Korea. His research interest includes security modeling, CPS and IoT cybersecurity, and intrusion and anomaly detection for Industrial Control System (ICS).

Kyounggon Kim received his BS degree in computer science from Soongsil University in 2008, and MS degree and PhD in information security from Korea University in 2015 and 2020, respectively. He is currently an Assistant Professor at the Department of Forensic Sciences, Naif Arab University for Security and Sciences (NAUSS). He has performed penetration testing for over 130 clients in various industries when he worked for Deloitte, PwC and boutique consulting firms during over 15 years. He was awarded 6th place at DefCon CTF in 2007 and a first prize at the First Hacking Defense Contest hosted by the Korea Information Security Agency. He has authored a book on Internet hacking and security and has translated numerous security books. His research interests include cybercrime and network forensics, vulnerability analysis, smart city security, and CPS and IoT security.

Huy Kang Kim received a BS degree in Industrial Management, MS degree in Industrial Engineering, and PhD degree in Industrial and System Engineering in Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea. He is a serial entrepreneur; he founded A3 Security Consulting in 1999 and AI Spera, the data-driven cyber threat intelligence service company in 2017. Currently, he is a professor in the School of Cybersecurity, Korea University. His recent research is focused on anomaly detection in the intelligent transportation system, online gaming and internet banking by using data analytics, and machine learning techniques.

How to cite this article: K. H. Kim, K. Kim, and H. K. Kim, *STRIDE-based threat modeling and DREAD evaluation for the distributed control system in the oil refinery*, ETRI Journal **44** (2022), 991–1003. https://doi.org/10.4218/etrij.2021-0181