DEVS-based Modeling Methodology for Cybersecurity Simulations from a Security Perspective

Jiyeon Kim[1], Hyung-Jong Kim[2*]
[1] Center for Software Educational Innovation, Seoul Women's University
Harang-ro 621, Seoul - South Korea
[e-mail: jykim07@swu.ac.kr]
[2] Dept. Of Information Security, Seoul Women's Universtiy
Harang-ro 621, Seoul - South Korea
[e-mail:hkim@swu.ac.kr]
*Corresponding author: Hyung-Jong Kim

Received December 3, 2019; revised January 27, 2020; accepted February 6, 2020; published May 31, 2020

Abstract

Security administrators of companies and organizations need to come up with proper countermeasures against cyber-attacks considering infrastructures and security policies in their possession. In order to develop and verify such countermeasures, the administrators should be able to reenact both cyber-attacks and defenses. Simulations can be useful for the reenactment by overcoming its limitations including high risk and cost. If the administrators are able to design various scenarios of cyber-attacks and to develop simulation models from their viewpoints, they can simulate desired situations and observe the results more easily. It is challenging to simulate cyber-security issues, because there is lack of theoretical basis for modeling a wide range of the security field as well as pre-defined basic components used to model cyber-attacks. In this paper, we propose a modeling method for cyber-security simulations by developing a basic component and a composite model, called Abstracted Cyber-Security Unit Model (ACSUM) and Abstracted Cyber-security SIMulation model (ACSIM), respectively. The proposed models are based on DEVS(Discrete Event systems Specification) formalism, a modeling theory for discrete event simulations. We develop attack scenarios by sequencing attack behaviors using ACSUMs and then model ACSIMs by combining and abstracting the ACSUMs from a security perspective. The concepts of ACSUM and ACSIM enable the security administrators to simulate numerous cyber-security issues from their viewpoints. As a case study, we model a worm scenario using ACSUM and simulate three types of simulation models based on ACSIM from a different security perspective.

Keywords: cybersecurity, cyberattacks, DEVS formalism, modeling and simulation

http://doi.org/10.3837/tiis.2020.05.018 **ISSN : 1976-7277**

1. Introduction

As cyber-attacks become increasingly sophisticated, detecting and defending against attacks are challenging. For example, advanced persistent threats (APT) generally consist of seven steps: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives[1]. Attackers try a variety of attacks until they finally achieve the goal of each step. In order to prevent such advanced cyber-attacks, it is necessary to predict various potential attacks and to design and verify countermeasures in advance. However, the following risks and limitations apply to the study of cyber-security in the real world.

- Risk of the destruction of targets - the purpose of most of active cyber-attacks is to cause damage to target systems. If we reenact cyber-attacks in the real world, damages such as the destruction of computer systems, networks, programs, and information, can occur.
- High cost of implementation of experimental environments – deploying a cyber attack infrastructure over the network is costly.

The higher cost of attacks has not only the more spatial restriction to conduct the attacks but also the higher risk by success of the attacks [2]. In other words, larger scale cyber-attacks are associated with higher risks and costs of the development. One alternative to overcome these limitations is a simulation. Dangerous or expensive experiments can be conducted in virtual environments [3,4]. It is also used for experiments that involve unexpected situations or high complex models [5].

Simulations have been used in many fields including manufacturing, network communications, and national defense [6,7]. Numerous studies on domain-specific simulations are in progress. These studies cover topics ranging from modeling methodologies to the development of simulation software that help easy implement, execution, and analysis of simulation models. Arena [8], OPNET (Optimized Network Engineering Tool) [9] and NS (Network Simulator) [10] are representative simulation software actively used in the domains of manufacturing and network communications. These software use pre-defined building blocks to design and implement simulation models for a variety of purposes. This is possible not only because the basic components of simulation are designed, but also because there is methodology to combine them.

Most of cyber-security simulations can hardly use building blocks from other studies, even they deal with same attacks. This is because security issues are diagnosed and resolved from a different perspective depending on the security manager's knowledge and their security policies. Furthermore, lack of theoretical basis for the modeling of security issues makes it difficult to develop and extend simulation models systematically. Applying various security goals to the simulation models is also challenging. A well-defined model for cyber-security simulations is very essential in order to simulate a wide range of security issues.

In this paper, we propose a modeling methodology for the cyber-security simulations. There are two steps in modeling security issues: modeling of cyberattacks and modeling of viewpoints from a security perspective. For the first step, we develop a basic component of cyber-security simulations, called the ACSUM(Abstracted Cyber-Security Unit Model). Each ACSUM is a building block of one scenario of attack. By sequencing and coupling multiple ACSUMs, we are able to model a variety of attack scenarios. For the second step, we develop a composite model, called the ACSIM(Abstracted Cyber-security SIMulation model), which consists of more than one ACSUM. Each ACSIM is modelled to have a single state by abstracting the state of ACSUMs in it.

The abstraction is carried out in consideration of the security issues of interest. In other words, once we model the attack scenario using ACSUMs, we can develop multiple simulation models from various security perspectives. The concept of ACSUM makes it easier for security administrators to model the desired scenario by combining multiple ACSUMs as the building blocks. The concept of ACSIM enables the administrators evaluate their countermeasures against the attacks from a variety of security perspectives.

Our theoretical basis of the modeling methodology is DEVS(Discrete EVent systems Specification) formalism[11], a modeling theory for discrete event systems. DEVS allows us to develop modular models by separating models from input and output interfaces. Even if the scenario changes, all we need to do is change the coupling structures rather than changing the model design. Furthermore, we are able to employ DEVS regardless of the types of system elements(e.g. hosts, networks, applications) or attack mechanisms. DEVS is a general methodology that provides a mathematical frame to discrete event systems. In order to explain our modeling methods, we model numerous cyberattacks from different security perspectives.

The remainder of this paper is organized as follows. Section 2 briefly reviews DEVS formalism and investigates previous studies on modeling and simulation in the cyber-security field. We also describe the concept of abstracting models from a security perspective. In the Section 3, we design ACSUM and ACSIM based on DEVS. In Section 4, we model a worm simulation using ACSUM and ACSIM with three types of scenarios. We show the experimental results in Section 5. Finally, the conclusion, application, and future work are presented in Section 6.

2. Related Work

2.1 DEVS formalism

DEVS provides a modeling method for discrete even systems. There are two kinds of DEVS models: atomic model and coupled model. Hierarchical and modular models can be generated by using these two models. The atomic model *M* is represented by the following formalism:

$$M = < X, S, Y, \delta int\ , \delta ext\ , \lambda, ta > \qquad (1)$$

where *X* is a set of input events, S is a set of states, *Y* is a set of output events, δint is an internal transition function where $S \rightarrow S$, δext is an external transition function where $Q \times X \rightarrow S$, λ is an output function where $S \rightarrow Y$, and *ta* is a time advance function where $Q = \{(s,e) \mid s \in S, 0 \leqq e \leqq ta(s)\}$ where *e* is a time elapsed since last transition. The coupled model specifies connection structures between atomic models. The coupled model *N* is represented by the following formalism:

$$N = (X, Y, D, M_d | d \in D, EIC, EOC, IC, Select) \qquad (2)$$

where X is a set of inputs through interfaces, Y is a set of outputs through interfaces, D is a set of the component names, M_d is DEVS models named one of the elements of D, EIC is connections between external inputs and component inputs, EOC is connections between component outputs and external outputs, IC is connections between component inputs and outputs, and Select is a tie-breaking function that determines priority of execution of components.

2.2 Modeling and simulation studies on cybersecurity

Attack tree[12] is the most well-known modeling method in cybersecurity. Attack tree models a process of achieving the final goal and represents detailed purposes or attacks. Attack tree is not enough for simulation models because there is no state transition function, time advanced function as well as state variables enabling the model to be traceable and executable. Most of previous studies on cyber-security simulations have addressed network attack, such as worm simulations and distributed denial of service (DDoS) simulations using network. Most worm simulation studies focus on network parameters and observe states of hosts on the basis of such parameters, because they model and simulate the simulations using network simulators. Ref. [14] conducts a worm simulation in the packet-level and observes the number of infected hosts according to scan rate, scanning strategy, link delay, network bandwidth, topology, payload size, and so on. Ref. [15] considers time delay, scan rate and a rate of vulnerable hosts in order to analyze a rate of infections of a random scanning worm. Ref. [16] observes the number of infected hosts considering rates of hiding hosts from external networks, online/offline of hosts, and downloading time in order to simulate a passive P2P worm. Refs. [17] configure network parameters including bandwidth, delay, and the types of transmission protocol. They also stochastically configure security properties including vulnerable hosts and system patching. Regarding previous DDoS simulation studies, Ref. [18] configures a network by considering background traffic and subnets. In addition, Ref. [19] also considers malicious traffic, the number of zombie systems, and properties of attack and defense. Although these studies show meaningful insights for worm or DDoS simulations, it is hard to model from a security perspective or to reflect countermeasures unless these features are implemented in the network simulator. In other words, the simulation performances are dependent on network simulators.

2.3 Model Abstraction from Security Viewpoints

Abstraction is a process that finds out the key nature of complex systems. In the security field, abstraction can be defined as a process that classifies the managed objects into specific groups according to security policies. Through the abstraction, security administrators can apply countermeasures to the groups by observing security status of each group, as shown in **Fig. 1**.

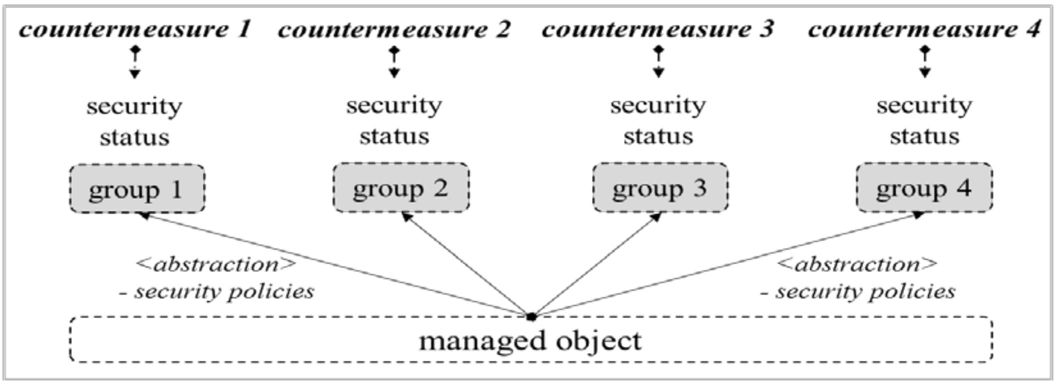


Fig. 1. The concept of abstraction of security issues

Security zone and RBAC (Role-based Access Control) are examples that use abstraction in the security field. In a security zone, a network can be classified into 'trusted zone / untrusted zone' or 'high security area / Internet / Demilitarized zone (DMZ),' according to the rules of the firewall. We call each classified zone a security zone. In addition, we can apply different countermeasures to each security zone according to the security policy, by making different detection rules as shown in **Fig. 2**.

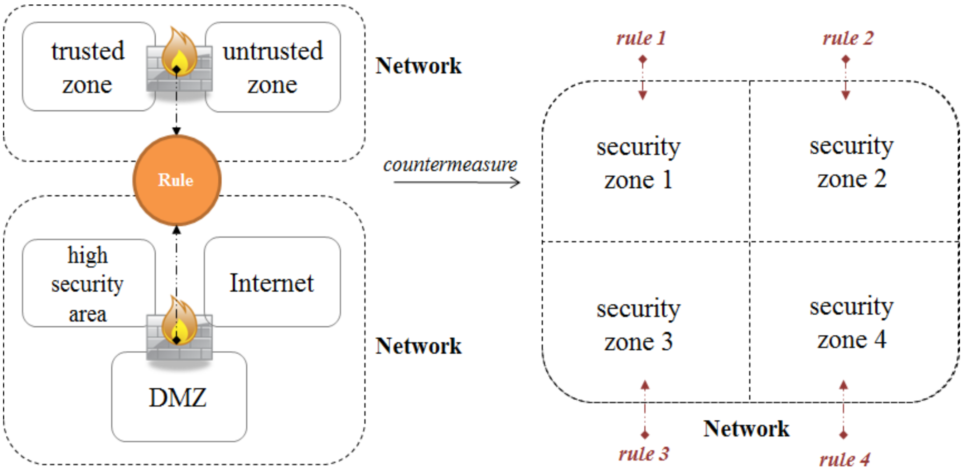


Fig. 2. Security zone – Network abstraction using firewall rules

RBAC is a technique in system security. RBAC restricts the use of resources by abstracting user identities. Users are divided into several groups such as 'student / professor' or 'undergraduate / graduate' according to the roles. In addition, a system can grant different permissions that allow the use of resources based on user group.

Considering the above examples, if we could develop multiple types of simulation models for an attack event, we can easily observe the effectiveness of various security policies and come up with optimal measures.

3. Modeling of ACSUM and ACSIM

3.1 Modeling of ACSUM

In order to model a cyber-security issue, we first need to look at a basic interaction process between a cyber-attack and a defense, because cyber-security is achieved by interactions between attackers and targets. If a target is vulnerable to an attack, the attack will be successful. If a target has countermeasures that can defend against the attack, the attack will fail because of the execution of the defense mechanisms. As shown in **Fig. 3**, by connecting the basic process with the cyber-security factors, attackers and targets can be matched to the elements of cyberspace, and their attack and defense mechanisms can be similarly matched to the attack and defense behaviors.

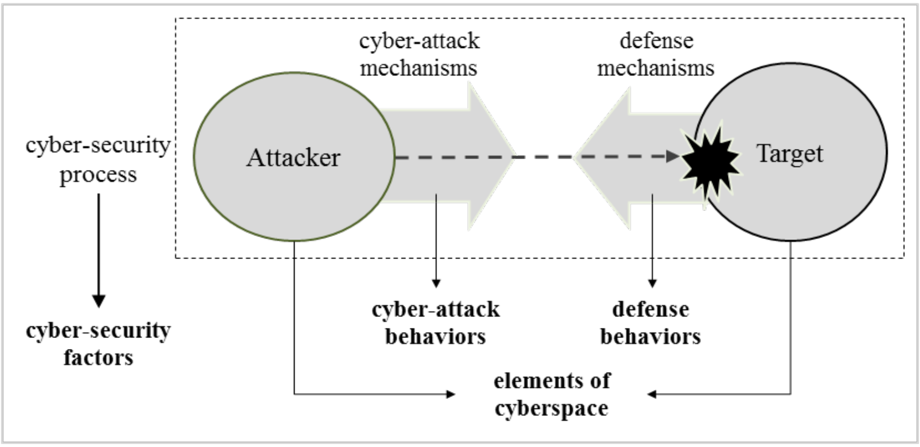


Fig. 3. Defining cyber-security factors from a basic cyber-security process

Accordingly, we need to design the elements of cyberspace and the cyber-attack behaviors for the modeling of the attackers, and to design the elements and the defense behaviors for the modeling of the targets. ACSUM is a unit model that can include a set of information about cyber-security factors. Therefore, we can design a basic scenario of cyber-attacks with a combination of ACSUMs.

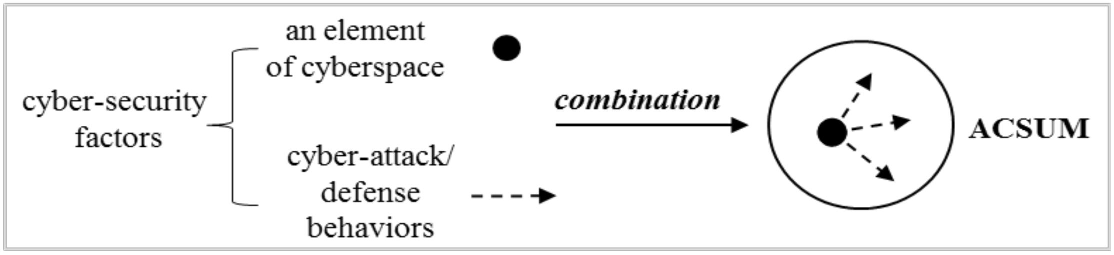


Fig. 4. Components of ACSUM

In order to make use of ACSUMs as building blocks in a simulation, an ACSUM should be able to change its state during the simulation. That is, changes of state of an ACSUM can be primitive factors that make a cyber-security model change its state. We represent ACSUMs using DEVS formalism as follows:

$$M_{ACSUM} = \{X_{ACSUM}, Y_{ACSUM}, S_{ACSUM}, \delta_{ext_{ACSUM}}, \lambda_{ACSUM}\}, \tag{3}$$

where X_{ACSUM} is a set of input events of ACSUM, Y_{ACSUM} is a set of output events of ACSUM, S_{ACSUM} is a set of states, $\delta_{ext_{ACSUM}}$ is an external transition function where S_{ACSUM} x $X_{ACSUM} \rightarrow S_{ACSUM}$, and λ_{ACSUM} is an output function. Although the ACSUM does not include the time advance function in the specification, it has a processing time as a state variable. The time advance function is considered in ACSIM that we finally run on a DEVS simulation engine.

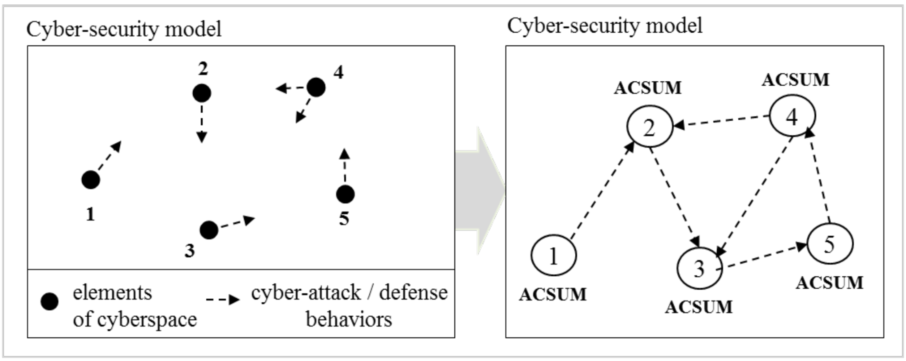


Fig. 5. The concept of coupling relations among ACSUMs

As shown in **Fig. 5**, each arrow indicates the effect of the behavior, and that implies a relation between two ACSUMs. The following cases create coupling relations: 1) a behavior of an ACSUM targets another ACSUM, 2) sharing an element of cyberspace among ACSUMs. In the case that input/output data is transmitted among ACSUMs, the data must be able to include information about the source, destination, and content. Accordingly, we define that a message format of an ACSUM consists of "source ACSUM ID," "destination ACSUM ID,", and "content," The "content" field can contain any exchanging messages including a worm file.

3.2 Modeling of ACSIM

Each ACSUM works as a building block of cyber-security simulations. On the other hand, there should be models for letting the ACSUMs operated in a discrete-event simulation environment. We transform ACSUMs into ACSIMs for this purpose. In order to develop a flexible cyber-security simulation model considering various security issues, we position ACSUMs into several groups according to various simulation purposes, as shown in **Fig. 6**. Each group is modeled into one ACSIM. By coupling ACSIMs, we can make a complete cyber-security simulation model. In the process of modeling an ACSIM, the concept of abstraction is required. This concept allows us to develop simulation models considering various security viewpoints.

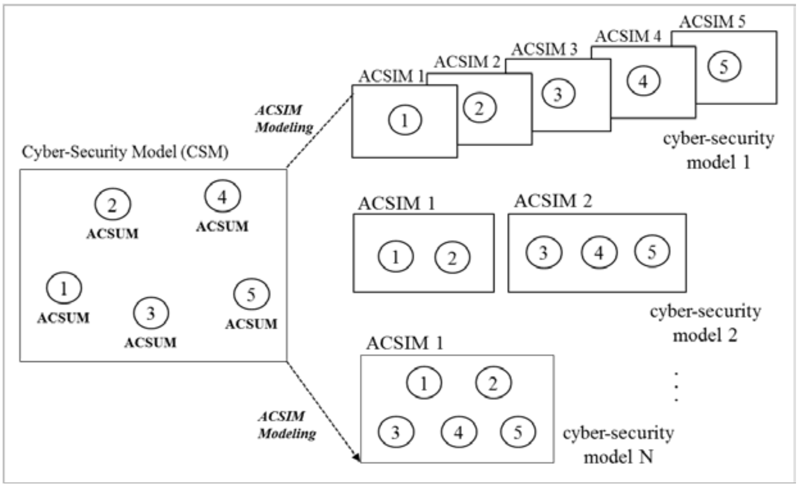
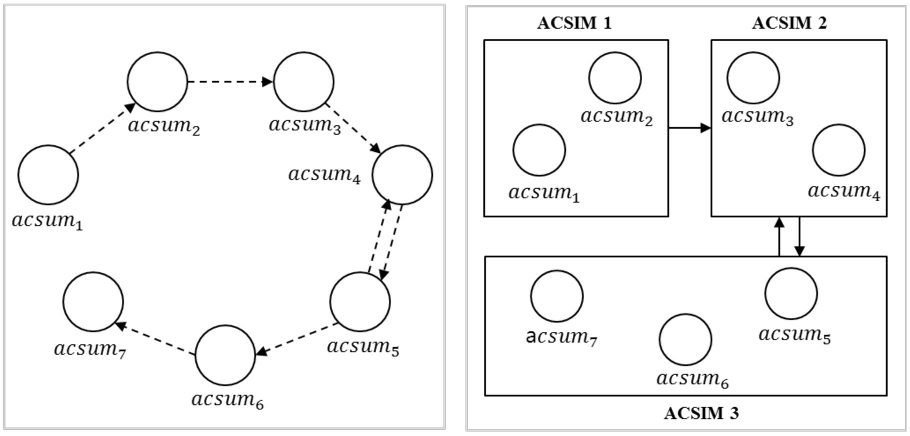


Fig. 6. Development of various ACSIMs with combinations of ACSUMs

After determining the ACSUMs that belong to an ACSIM, we develop each ACSIM by extracting and combining key state variables from the ACSUMs based on a simulation purpose. Since a funcdamantal cyber-attack scenario should not be changed in the ACSIM modeling step, ACSIMs should be able to maintain coupling relations among ACSUMs as shown **Fig. 7**.



(a) ACSUM-level modeling (b) ACSIM-level modeling

Fig. 7. An example of developing ACSIMs containing coupling relations among ACSUMs

Considering the above example, we can model a cyber-security model (CSM) in phases as shown in **Fig. 8**.

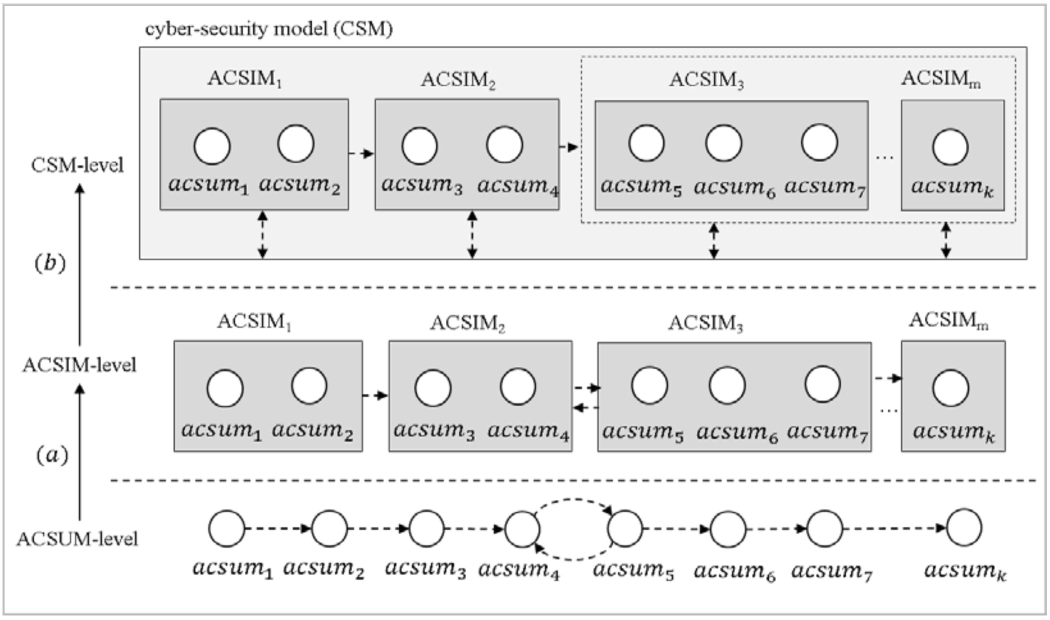


Fig. 8. A process of developing a cyber-security model in phases

In **Fig. 8(a)**, an ACSIM is modeled with a partitioned set of ACSUMs. Let the sets of ACSUMs and ACSIMs to be U and D, respectively. Given $U = \{u \mid u \subseteq \{acsum_1, acsum_2, \ldots acsum_k\}, u \neq ø\}$ and $D = \{d \mid d \in \{ACSIM_1, ACSIM_2, \ldots, ACSIM_m\}\}$ where both k and m are nature numbers, an abstraction function f is defined by $f(u) = d$. The specification of d is as shown in Equation (4).

$$d = <X_d, Y_d, S_d, \delta ext_d, \delta int_d, \lambda_d, ta_d> \tag{4}$$

X_d and Y_d are the sets of inputs and outputs through interfaces of d, where d is one of the ACSIMs. S_d is a set of states, δext_d is an external transition function, and δint_d is an internal transition function of d. λ_d is an output function and ta is a time advance function. Each ACSIM advances its time by calculating the processing times of ACSUMs belonging to the ACSIM.

We can develop the CSM by coupling ACSIMs. The CSM consists of more than one d or more than one coupled model that is already composed of more than one d, as shown in **Fig. 8(b)**. Accordingly, we can finally model the CSM in phases as shown in **Fig. 8**, and the complete structure of the model is shown in **Fig. 9.** The CSM is specified as shown in Equation (2).

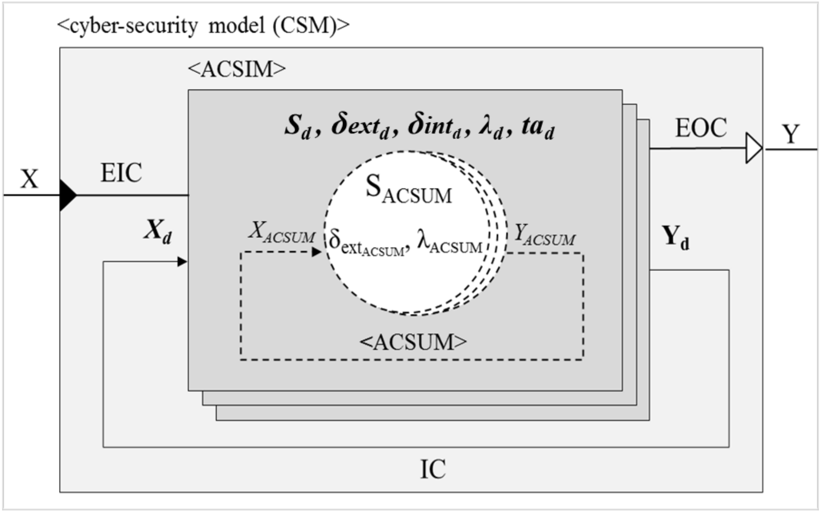

<cyber-security model (CSM)>
<ACSIM>
S_d, δext_d, δint_d, λ_d, ta_d
EOC
Y
X
EIC
S_{ACSUM}
X_d
X_{ACSUM}
δext_{ACSUM}, λ_{ACSUM}
Y_{ACSUM}
Y_d
<ACSUM>
IC

Fig. 9. The complete structure of a cyber-security model based on DEVS

In **Fig. 9**, X and Y are sets of inputs and outputs of a CSM, respectively. EIC is a set of connections between inputs of the CSM and ACSIMs, EOC is a set of connections between outputs of ACSIMs and CSM, and IC is a set of connections between inputs and outputs of ACSIMs.

In the modeling of ACSIMs, the location of the ACSUMs determines whether ACSIM I/O events occur. For example, **Fig. 10(a)** is a case in which two ACSUMs belong to different ACSIMs. In this case, if acsum_2 sends a message to acsum_3, an I/O event between ACSIM_1 and ACSIM_2 occurs. **Fig. 10(b)** is a case in which two ACSUMs belong to the same ACSIM. In this case, ACSIM_1 processes an I/O event between acsum_2 and acsum_3 internally, so that no I/O event occurs.

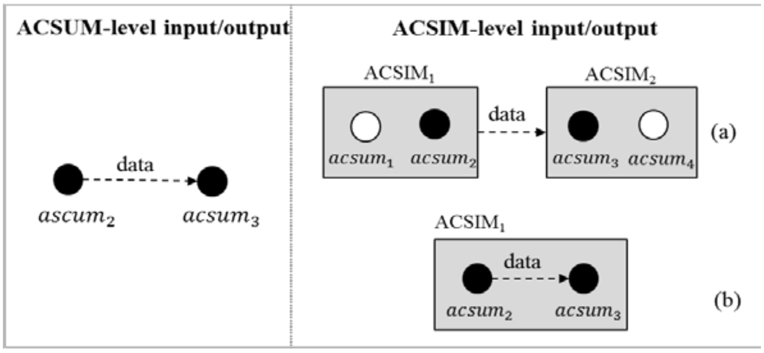


Fig. 10. I/O event of ACSIMs based on the location of the ACSUMs

4. Modeling of a Worm Simulation from a Security Perspective

This section describes a process of modeling worm simulations under three scenarios that have different security purposes. In addition, we model and simulate the three scenarios to verify the models' design and execution. We consider a fundamental behavior of a worm attack. Not only traditional worm attacks (e.g. Code red[20]) but also recent worm attacks (e.g. WannaCry[21]) have the characteristics of propagating themselves through the network. The infected host scans vulnerable hosts and transmits a worm file to the target host. When the target host executes the file, the host is infected and keeps propagating the worm following the fundamental behavior.

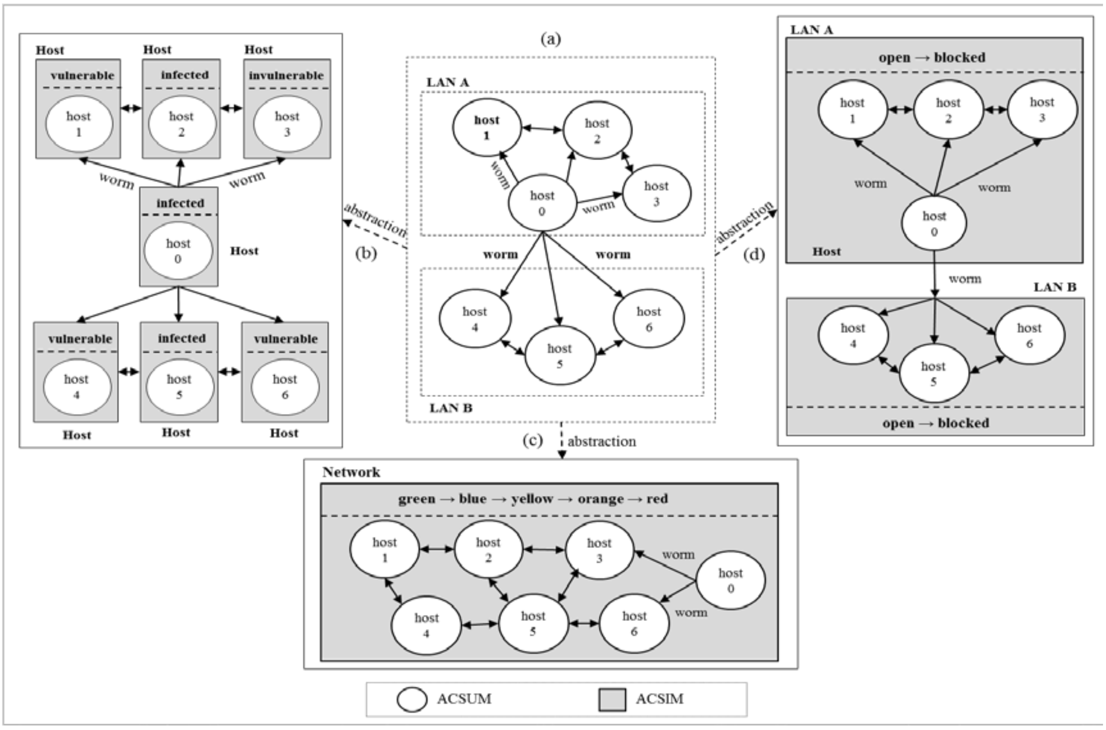


Fig. 11. Example scenarios(b, c, d) of modeling various forms of ACSIMs of a fundamental worm propagation process(a) from a different security perspective

In **Fig. 11(a)**, host0, which has been initially infected by a worm, propagates the worm file to the other hosts in the internal network (LAN A) and the external network (LAN B). Each host is modelled as ACSUM, which has the fundamental behavior. A set of ACSUMs is as follows:

$$\text{ACSUM} = \{\text{host1(system)}, \text{host2(system)}, ..., \text{host\#(system}\} \qquad (5)$$

Because each host targets each other, a worm attack corresponds to attacks involving a bidirectional relation. A set of coupling relations are defined as follows:

$$CR_{\text{ACSUM}} = \{(\text{host0}, \text{host0}), (\text{host0}, \text{host1}), (\text{host0}, \text{host2}), ..., (\text{host2}, \text{host0}), (\text{host1}, \text{host0}), ..., (\text{host\#}, \text{host\#})\} \qquad (6)$$

Each ACSUM is modeled after designing coupling relations. First, we determine the initial states of all the ACSUMs and then define the ACSUM behaviors as a set of states for each ACSUM. In the worm simulation, a host performs three functions: worm execution, scanning the other hosts, and worm transmission [22]. The set of states for each ACSUM is defined as follows:

$$S_{\text{host\#}} = \{s_0, \text{execute}, \text{scan}, \text{transmit}\}\text{, where } s_0 \text{ is the initial state} \qquad (7)$$

Next, ACSIMs are modelled based on the simulation purposes. We develop three scenarios of ACSIMs by abstracting security issues. The purpose of Scenario 1 and 2 is to observe the status of the simulation model; the purpose of Scenario 3 is to control network groups according to security policies.

- Scenario 1 - observing the state of each host **(Fig. 11(b))**

This scenario is to observe the changing state of each host like numerous previous worm simulation studies. We can create an ACSIM from one ACSUM. The ACSIM can have states such as ‘vulnerable,’ ‘infected,’ and ‘invulnerable.’ In order for the ACSIM to abstract the three states, the key state variables from the ACSUMs must be extracted in advance. Vulnerabilities and attack behaviors from the ACSUMs can be key state variables. If one ACSUM is not vulnerable to a worm, the state of this ACSIM will always be ‘invulnerable.’ If an ACSUM is vulnerable to the worm, however, the state of the ACSIM will be different depending on the state of the ACSUM. If the state of the ACSUM is ‘execute,’ we can abstract, to the ACSIM, a state of ‘vulnerable’ because the host will be infected when the worm file has executed successfully. If the state of the ACSUM is one of the others, we can abstract the state ‘infected’ to the ACSIM.

- Scenario 2 - observing an entire network to alert Internet users in order to prepare for the worms **(Fig. 11(c))**

This scenarios is to observe an entire network when a worm is introduced. In this case, all hosts are developed into one ACSIM by abstracting the statuses of multiple ACSUMs. As states of the ACSIM, there can be five abstracted states: ‘green,’ ‘blue,’ ‘yellow,’ ‘orange,’ and ‘red.’ We can determine the state by observing the behaviors of host and calculating the number of behaviors of ‘scan’ and ‘transmit’ of ACSUMs in the network.

- Scenario 3 – controlling a local network when a worm is introduced **(Fig. 11(d))**

This scenarios is to control LANs according to security policies. In this case, each LAN is modeled into one ACSIM with states that can be ‘open’ or ‘blocked.’ If the number of behaviors of ‘scan’ and ‘transmit’ of hosts is more than a threshold of the security policies, a countermeasure that blocks outbound traffic will be applied.

As shown by these three scenarios of ACSIM modeling of a worm simulation, we are able to develop a basic worm propagation scenario that is modeled with combinations of ACSUMs into different types of ACSIMs including hosts, local networks, and an entire network. As a result, we are able to model and simulate the worm simulation considering management groups, and to apply countermeasures and evaluate the effects.

5 Simulation Results

We implement three simulation models using DEVSJAVA [23], which is a simulation software used for executing DEVS models. In our simulation, a scenario of **Fig. 11(a)** consists of 100 hosts and the percentage of invulnerable hosts on the network is set from 20 to 30.

Our basic worm propagation scenario comprising low-level behaviors is used for the following three simulations. Because the three simulations are modeled for different purposes from various viewpoints, we can observe and obtain different simulation results from each of them. **Table 1** shows the first scenario and describes how the states of ACSUMs are abstracted to that of ACSIMs. This scenario can be also used in numerous studies that observe the state of hosts to find infected host.

Table 1. (Scenario 1) Abstracted states of ACSIMs obtained from ACSUMs

Type of Model	State of a host		
ACSUM	Execute	Scan	Transmit
ACSIM	Vulnerable	Infected	

Table 2 presents the relation between security levels and the rate of infected hosts in Scenario 2.

Table 2. (Scenario 2) Security levels according to the rate of infected host in the network

Security level	The rate of infected host
Green	0~5%
Blue	5%~20%
Yellow	20%~40%
Orange	40%~70%
Red	70%~100%

Fig. 12 compares Scenario 1 and 2. Both scenarios are simulated based on the basic worm propagation scenario. However, because Scenario 2 is modeled to trace the state of the network rather than the state of the hosts, the changing security levels can be observed. The 'yellow' level only has a short time to stay because the worms spread very rapidly. In the graph, the greater the slope of Scenario 1, faster the change in security level.

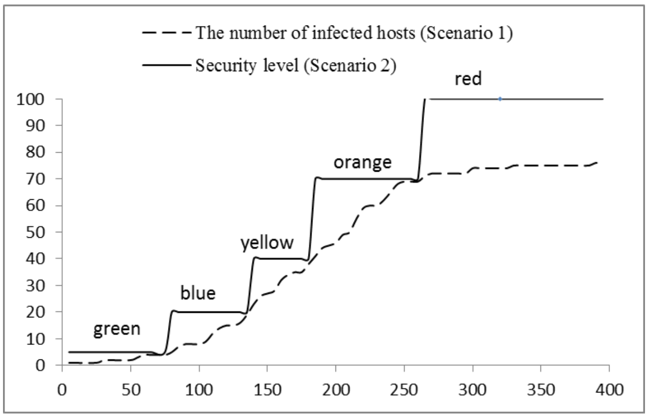
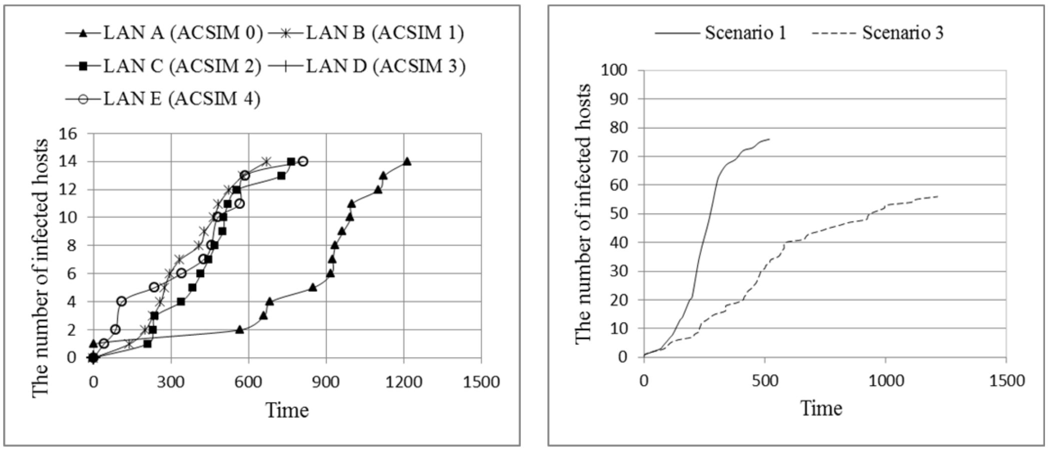


Fig. 12. Scenario 1 vs Scenario 2



(a) Scenario 3 - The number of infected hosts of each LAN

(b) Scenario 3 vs Scenario 1

Fig. 13. Simulation results of Scenario 3

In the Scenario 3, there are five LANs in the network, with each LAN comprising 20 hosts. In this scenario, each LAN has a countermeasure that blocks outbound traffic when three hosts are infected with worms. Even though outbound traffic of a LAN is blocked, a worm propagation will continue until all vulnerable hosts on the LAN are infected. **Fig. 13(a)** shows a simulation result for Scenario 3. The number of infected hosts on LAN D is zero during the simulation. This is because all the LANs except LAN D blocked their outbound traffic, in accordance with security policies, before the vulnerable hosts on LAN D could be scanned. **Table 3** shows simulation results of the blocking time for each LAN. **Fig. 13(b)** compares Scenario 1 and 3. In the case of Scenario 3, the total number of infected hosts on the entire network is less than that in the case of Scenario 1 because of the countermeasures, even though the total number of vulnerable hosts on the entire network is the same. In addition, it is clear that the velocity of Scenario 3 is also lower than that of Scenario 1.

Table 3. (Scenario 3) A simulation result of blocking time of outbout traffic

LAN	The changes of state	Time
LAN A	open → blocked	660
LAN B	open → blocked	230
LAN C	open → blocked	240
LAN D	open→ open	-
LAN E	open → blocked	110

6. Conclusion

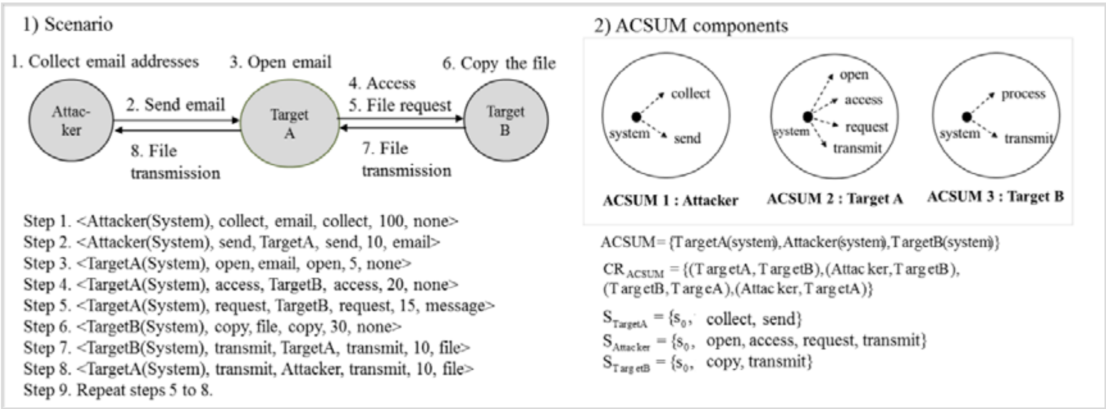
In order to simulate cyber-security problems from viewpoints of security administrators, we have proposed a modeling method for developing simulation models using basic components and composite models. We have developed the two types of models, the ACSUM and the ACSIM.

The ACSUM, a basic component, consists of an element of cyberspace and its attack or defense behaviors. By combining pre-defined ACSUMs, we can develop various cyber-attack scenarios, which progress based on the interactions among the elements. To help modelers develop ACSUMs easily, we suggested a scenario-based modeling method for ACSUMs.

The ACSUMs can be transformed into multiple composite models, the ACSIMs, after considering security issues. In the security field, the concept of abstraction, which is a process that classifies the objects to be managed into specific groups based on security policies, is usually employed to apply countermeasures. Accordingly, we suggested methods for classifying ACSUMs into specific groups using abstraction and developed each group into an ACSIM based on DEVS formalism, a specification method for discrete event systems.

As a case study, we have developed various worm simulation models considering countermeasures and observation group units, such as hosts, local networks, and entire network, using our modeling methods. We then have simulated and analyzed them to verify the models' design and execution. Through the simulation, our model is useful for security administrators by enabling them to simulate their security issues from various viewpoints considering their security policies. Since ACSUMs can also represent other elements of cyberspace and their behaviors, we can model other cyber-attacks using our modeling method.

For example, we can model APT attacks based on our method. **Fig. 14** shows the modeling of ACSUMs for an APT attack based on an example scenario usually used in APT attacks.



Step 1. <Attacker(System), collect, email, collect, 100, none>
Step 2. <Attacker(System), send, TargetA, send, 10, email>
Step 3. <TargetA(System), open, email, open, 5, none>
Step 4. <TargetA(System), access, TargetB, access, 20, none>
Step 5. <TargetA(System), request, TargetB, request, 15, message>
Step 6. <TargetB(System), copy, file, copy, 30, none>
Step 7. <TargetB(System), transmit, TargetA, transmit, 10, file>
Step 8. <TargetA(System), transmit, Attacker, transmit, 10, file>
Step 9. Repeat steps 5 to 8.

$ACSUM = \{TargetA(system), Attacker(system), TargetB(system)\}$

$CR_{ACSUM} = \{(TargetA, TargetB), (Attacker, TargetB), (TargetB, TargeA), (Attacker, TargetA)\}$

$S_{TargetA} = \{s_0, \text{ collect, send}\}$
$S_{Attacker} = \{s_0, \text{ open, access, request, transmit}\}$
$S_{TargetB} = \{s_0, \text{ copy, transmit}\}$

Fig. 14. ACSUM modeling for an APT attack

If we develop all the possible components of an ACSUM in advance, in addition, we can make it possible to develop a wide range of cyber-security issues into ACSUMs by combining them, as shown in Fig. 15.

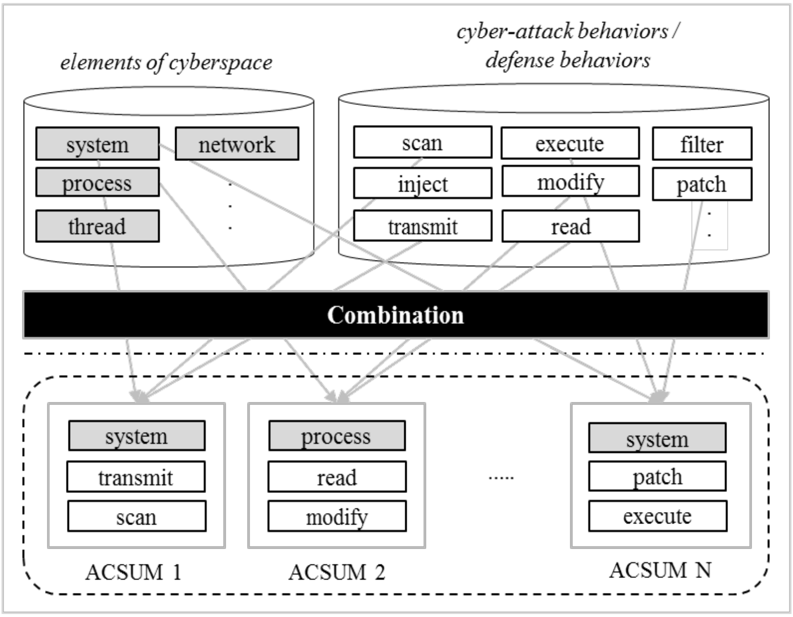


Fig. 15. Development of ACSUMs by combination of pre-extracted elements of cyberspace and behaviors of cyber-attacks and defenses

For the extraction of the behaviors of cyber-attacks and defenses, we can analyze CAPEC (Common Attack Pattern Enumeration and Classification) [24] that defines 400 types of attack patterns. By incorporating, in advance, all possible behaviors from CAPEC, we can develop various cyber-attacks using combinations of the extracted behaviors.

This study can be used to develop a cyber-security simulator that supports various cyber-security simulations. In the simulator, ACSUMs can be used as building blocks for the cyber-security simulation, and subsequently can be transformed into a simulation model using the ACSIM modeling method. As future work, we will implement the simulator, and develop a database that involves cyber-attack behaviors and defense behaviors that are extracted from CAPEC. Moreover, we will develop various ACSUMs in advance so that modelers can easily model various cyber-security issues by combining the pre-developed ACSUMs.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2062654). This research was also supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2018R1D1A1B07050543)

References

[1] Hutchins, Eric M., Michael J. Cloppert, and Rohan M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, 1(1), 80-106, 2011. Article (CrossRef Link)

[2] COHEN, Fred, "Simulating cyber attacks, defences, and consequences," *Computers & Security*, 18(6), 479-518, 1999. Article (CrossRef Link)

[3] NICOL, David M, "Modeling and simulation in security evaluation," *IEEE security & privacy*, 3(5), 71-74, 2005. Article (CrossRef Link)

[4] Saunders, John H., "The Case for Modeling and Simulation of Information Security," in *Proc. of Computer Security Institute Conference*. Article (CrossRef Link)

[5] LAW, Averill M.; KELTON, W. David; KELTON, W. David, *Simulation modeling and analysis*, New York: McGraw-Hill, 2000.

[6] SUNG, Chang Ho; MOON, Il-Chul; KIM, Tag Gon, "Collaborative work in domain-specific discrete event simulation software development: Fleet anti-air defense simulation software," in *Proc. of 2010 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises. IEEE*, pp. 160-165, 2010. Article (CrossRef Link)

[7] Valentin, Edwin C., and Alexander Verbraeck, "Requirements for domain specific discrete event simulation environments," in *Proc. of Simulation Conference, 2005 Proceedings of the Winter. IEEE*, 2005. Article (CrossRef Link)

[8] Arena Simulation Software. https://www.arenasimulation.com

[9] OPNET Network Simulator. http://opnetprojects.com/opnet-network-simulator

[10] NS-3 Network Simulator. https://www.nsnam.org/

[11] Zeigler, Bernard P., Herbert Praehofer, and Tag Gon Kim, *Theory of modeling and simulation. 2nd edition*, Academic Press, 2000.

[12] Schneier, Bruce, "Attack trees," *Dr. Dobb's journal*, 24(12), 21-29, 1999. Article (CrossRef Link)

[13] Bistarelli, Stefano, Fabio Fioravanti, and Pamela Peretti, "Defense trees for economic evaluation of security investments," in *Proc. of Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on. IEEE*, 2006. Article (CrossRef Link)

[14] Sharif, Monirul I., George F. Riley, and Wenke Lee, "On the Need for Packet–Level Details in Worm Simulations," pp. 1-9. Article (CrossRef Link)

[15] Lin, Siming, and Xueqi Cheng, “NSME: A Framework for Network Worm Modeling and Simulation,” *Network Control and Engineering for Qos, Security and Mobility, V. Springer US*, 201-212, 2006. Article (CrossRef Link)

[16] Ting Chen, Xiao-song Zhang, and Yue Wu, “FPM: FOUR-factors Propagation Model for passive P2P worms,” *Future Generation Computer Systems*, vol. 36, pp. 133-141, 2014. Article (CrossRef Link)

[17] Min-huan, Huang, et al, “Research on Technologies of Building Experimental Environment for Network Worm Simulation,” in *Proc. of Parallel and Distributed Systems (ICPADS), 2009 15th International Conference on. IEEE*, 2009. Article (CrossRef Link)

[18] Lu, Kejie, et al, “Robust and efficient detection of DDoS attacks for large-scale internet,” *Computer Networks*, 51(18), 5036-5056, 2007. Article (CrossRef Link)

[19] Du, Ping, and Akihiro Nakao, “OverCourt: DDoS mitigation through credit-based traffic segregation and path migration,” *Computer Communications*, 33(18), 2164-2175, 2010. Article (CrossRef Link)

[20] MOORE, David; SHANNON, Colleen; CLAFFY, K, “Code-Red: a case study on the spread and victims of an Internet worm,” in *Proc. of the 2nd ACM SIGCOMM Workshop on Internet measurment*, 273-284, 2002. Article (CrossRef Link)

[21] MOHURLE, Savita; PATIL, Manisha, “A brief study of wannacry threat: Ransomware attack 2017,” *International Journal of Advanced Research in Computer Science*, 8(5), 2017. Article (CrossRef Link)

[22] Jiyeon KIM and Hyung-Jong Kim, “Defining Security Primitives for Eliciting Flexible Attack Scenarios Through CAPEC Analysis,” *Information Security Applications. WISA 2014. Lecture Notes in Computer Science, Springer, Cham*, vol. 8909, 370-382, 2015. Article (CrossRef Link)

[23] H. Sarjoughian, “Introduction to DEVS modeling &simulation with JAVA: Developing component-based simulation models,” *Arizona State University*, 2005. Article (CrossRef Link)

[24] MITRE, “Common Attack Pattern Enumeration and Classification,”. http://capec.mitre.org

[25] Pudar, Srdjan, Govindarasu Manimaran, and Chen-Ching Liu, “PENET: A practical method and tool for integrated modeling of security attacks and countermeasures,” *Computers &Security*, 28(8), 754-771, 2009. Article (CrossRef Link)

[26] Blanchet, Bruno, and Ben Smyth, “ProVerif 1.85: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial,” *National Institute for Research in Computer Science and Control*, 2011. Article (CrossRef Link)

[27] Moore, David, and Colleen Shannon, “Code-Red: a case study on the spread and victims of an Internet worm,” in *Proc. of the 2nd ACM SIGCOMM Workshop on Internet measurment. ACM*, 2002. Article (CrossRef Link)

[28] Moore, David, et al, “Inside the slammer worm,” *Security &Privacy, IEEE*, 1(4), 33-39, 2003. Article (CrossRef Link)

[29] Moore, David, et al, “Internet quarantine: Requirements for containing self-propagating code,” in *Proc. of INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, 2003. Article (CrossRef Link)

[30] Ingalls, Ricki G, “Introduction to simulation,” in *Proc. of the 40th Conference on Winter Simulation. Winter Simulation Conference*, 2008. Article (CrossRef Link)

[31] Whitley, John N., et al, “Attribution of attack trees,” *Computers &Electrical Engineering*, 37(4), 624-628, 2011. Article (CrossRef Link)

[32] Saini, Vineet, Qiang Duan, and Vamsi Paruchuri, “Threat modeling using attack trees,” *Journal of Computing Sciences in Colleges*, 23(4), 124-131, 2008. Article (CrossRef Link)

[33] Jiyeon Kim and Hyung-Jong Kim, "Poster: Modeling of APT Attacks through Transforming Attack Scenarios into DEVS Models," *IEEE Security & Privacy*, 2015. Article (CrossRef Link)

[34] Lee, Jang-Se, et al, "Linux-Based system modelling for cyber-attack simulation," *Artificial Intelligence and Simulation. Springer Berlin Heidelberg*, 585-596, 2005. Article (CrossRef Link)

[35] Douligeris, Christos, and Aikaterini Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks*, 44(5), 643-666, 2004. Article (CrossRef Link)

[36] Kaplan, Bonnie, and Joseph A. Maxwell, "Qualitative research methods for evaluating computer information systems," *Evaluating the Organizational Impact of Healthcare Information Systems*, 30-55, 2005. Article (CrossRef Link)

[37] Kuhl, Michael E., et al, "Cyber attack modeling and simulation for network security analysis," in *Proc. of the 39th Conference on Winter Simulation: 40 years! The best is yet to come. IEEE Press*, 2007. Article (CrossRef Link)

[38] Guizani, Mohsen, et al, *Network Modeling and Simulation: A Practical Perspective*, Wiley, 2010.

[39] Aslan A., Stephen C, "Learning is Change in Knowledge: Knowledge-based Security for Dynamic Policies," in *Proc. of 2012 IEEE 25th Computer Security Foundations Symposium*, 308-322, 2012. Article (CrossRef Link)

Jiyeon Kim received her BSc and PhD degrees in information security engineering from Seoul Women's University in 2007 and 2013, respectively. She was a postdoctoral research associate in the department of electrical and computer engineering at Carnegie Mellon University from 2014 to 2017. She is a teaching professor at Seoul Women's University since 2019. Her research interests include cybersecurity, network security, cloud and IoT security, artificial intelligence, and M&S(modeling and simulation) methodology.

Hyung-Jong Kim has been a faculty member of Seoul Women's University since Mar. 2007. He worked as a principal researcher of Korea Information Security Agency (KISA) from 2001 to 2007. He received his information engineering B.S. (1996) degree in Sungkyunkwan university, Korea. Also, he received his M.S. (1998) and Ph. D. (2001) degree in electrical computer engineering department of Sungkyunkwan university. He worked in the CyLab Korea at CMU (Carnegie Mellon University) as a visiting scholar from 2004 to 2006. His research interests include cloud computing security, VoIP security, privacy protection and simulation modeling methodology.