Construction of Secure Internal Network with Communication Classifying System Using Multiple Judgment Methods

Hirokazu Hasegawa

Information Security Office,
Nagoya University
Nagoya, Japan
Email: hasegawa@icts.nagoya-u.ac.jp

Yuya Sato

ZOZO Technologies, Inc.
Tokyo, Japan
Email: yuya.sato@zozo.com

Hiroki Takakura

Center for Cybersecurity
Research and Development,
National Institute of Informatics
Tokyo, Japan
Email: takakura@nii.ac.jp

***Abstract*—Recent sophistication of cyber attacks targeting organizations such as companies, governments, and so on, have made the complete protection of our network very difficult. However, with the conventional measures including intrusion detection systems or firewalls, our network is not completely safe from intrusion because the dedicated malwares can slip through such measures. Thus, the separated network is one of the most effective countermeasures. In the separated network, an organization's internal network is divided into multiple segments, and fine access control among separated segments is conducted. To support a separated network construction, an automated ACL generation system has been previously proposed because the separated network is difficult to construct. However, this method focuses on the business continuity of the organization, and ACL will unconditionally permit the communication of a section where traffic is observed to maintain business continuity. Therefore, we have proposed a communication classifying system to judge the necessity of communication and its protocol by a two-step investigation. First, the system judges the consistency of the communication permitted by conventional systems. Second, if inconsistent communication is detected, the system judges the validity of the communication by checking the waiting state of its destination terminal. However, the system misjudges the necessity of communication in several conditions. In this paper, to resolve the misjudgment of the conventional communication classifying system, we improve it to conduct statistical analysis as a third investigation. In the experiment, the proposed system detected and terminated unintended communication between clients and servers. Thus, the proposed system outperformed the conventional communication classifying system.**

***Keywords*—Targeted Attacks; Network Separation; Access Control; Statistical Analysis.**

I. Introduction

This paper is follow up of our previous paper "An Evaluation on Feasibility of a Communication Classifying System" already published in the proceedings of SECURWARE 2019 [1].

Recently, cyber attacks targeting organizations such as specific companies or governments have frequently occurred. Such attacks are called targeted attacks, and attackers have specific purposes, e.g., information theft and sabotage activities. In contrast to conventional indiscriminate attacks committed for the fun of a solo attacker, targeted attacks are conducted by multiple attackers belonging to well-funded crime groups for money making. In order to reach the goal of the attack, attackers use sophisticated methods and continue the attack persistently. For example, they carefully investigate the target and prepare dedicated malwares against the target including zero-day attacks.

Generally, organizations have applied several cybersecurity measures. For example, firewalls and intrusion detection systems are located on the border between the internet and the internal network to prevent intrusion by malwares. In many cases, such countermeasures use pattern matching technology with known malicious information and probably cannot detect unknown attacks such as zero-day attacks. Therefore, when sophisticated attacks slip through these countermeasures, we cannot prevent their invasion.

Therefore, the sophistication of cyber attacks has made it very difficult to protect our network from the intrusion of malwares completely. Against such a situation, recent countermeasures have been focusing on after the intrusion of malwares. The goal of such countermeasures is the mitigation of damages by the attacks, e.g., preventing information leakage and ceasing file destruction activities [2].

A separated network is one of the effective countermeasures [3], and our research has been focusing on it. It divides the organization's internal network into multiple segments and performs fine access control among the divided segments. In the conventional network structure, only a single segment without any access control is deployed, that is, all terminals are connected to the segment, and they can directly communicate with all others. In contrast to such a traditional structure, the separated network restricts communication in the internal network, and we can prevent unintended communication caused by malwares, e.g., lateral movement. In addition, when we detect malwares, it can minimize the harmful effect on business continuity because we can isolate only the infected segment or terminals.

It is difficult to construct and maintain the separated network because the border among segments and its access controls must be determined using various information concerning networks, human resources, business contents, and so on. Moreover, the change in human resources or business contents should be followed to maintain access control. Therefore, the separated network is not cost-effective, and many organizations still use traditional structures in the internal network.

In our earlier work, we proposed several systems to solve such problems and to support constructing a separated network. In our research, we assume a general organization's network is divided into several segments based on the department. Our goal is to construct a separated network by applying fine access controls permitting only necessary communication to network equipment that is the border of each segment. A necessary communication is defined as legitimate communication required for the works of users. For example, when a user needs to access the file in server A, the communication between the user and server A is defined as necessary. On the other hand, when a user never uses server B, the communication between the user and server B is defined as unnecessary.

An automated ACL generation system is our first work [4], and the system generates ACL automatically based on the user's access authority against files or directories in the servers and on existing communication in the network. We call this system "AAGS (Automated ACL Generation System)" in this paper. Although it reduces the burden of separated network construction by the administrators, the generated ACL allows overly permission and prohibition of the communication.

AAGS may permit several unnecessary communications. To avoid such overly permission, we need a detailed judgment method for the necessity of communication. Thus, we proposed a communication classifying system [5] to avoid overly permission. In this paper, we call the system a Communication Classifying System "(CCS)". The system carefully investigates existing communication and evaluates the consistency and the validity of the observed communication by checking the state of its destination terminal.

Here, we improved and implemented the CCS, and we verified its feasibility in our experimental network [1]. As a result of the experiment, we found several problems. These problems make CCS misjudge the necessity of communication in several cases.

Moreover, we improve the CCS to conduct a three-step investigation for verification of communication necessity. We deploy a statistical analysis for the third investigation to solve the problem of the CCS. Therefore, we implemented an extended CCS and experiment to evaluate the feasibility of the system.

The rest of this paper is organized as follows: In Section II, we introduce researches against targeted attacks and related works. Section III presents the proposal system. The implementation of the proposed system explained in Section IV. In Section V, we describe the evaluation of the proposal system. Finally, we conclude this paper in Section VI.

II. Related Works

In this section, we introduce related works for mitigating targeted cyber attacks and our previous research.

A. Research for Preventing Malware Activities

Many researchers have done several works to prevent+ malware activities in internal networks. Alessandro et al. proposed a method for modeling communication patterns of malwares that perform lateral movement [6]. However, it is not cost-effective to employ this method because we need to install a communication analysis tool on all terminals in the network. In the separated network, the spread of infection can be suppressed without installing special tools on the terminal because the ACL limits the communication area of malwares.

Also, several methods to construct the separated network have been widely studied. Watanabe et al. proposed a VLAN (Virtual Local Area Network) configuration method [7]. This method monitors traffic in the network and generates a network design using the monitoring information. When a certain amount of traffic exceeding the threshold among terminals is observed, a new VLAN concerning the terminals is generated. Because the terminals are frequently communicating with each other, it is effective from the viewpoint of the amount of traffic volume. However, if the VLAN is generated, the infected terminal is in the VLAN, it cannot prevent malware activities in that VLAN. According to [8] [9] [10] in supporting VLAN construction, the works do not pay attention to constructing moderate access controls among VLANs because such works only focus on the network efficiency. Besides the above researches, several products, e.g., "VLAN.Config" [11], for constructing VLAN automatically are difficult to generate ACL for the constructed network.

On the other hand, some researchers focus on the segmentation of the internal network for security measures. Mujib et al. constructed a micro-segmentation environment by using Cisco Application Centric Infrastructure (ACI) and evaluated the effectiveness of micro-segmentation against cyber attacks [12]. The result shows that the micro-segmentation is effective against cyber attacks. However, it is not shown what criteria should be used to construct micro-segmentation in a real environment network. Wagner et al. proposed a semi-automated network segmentation construction method in [13]. Moreover, in [14], they proposed a fully-automated network segmentation generation method focusing on security, cost, mission performance. Their proposals are effective methods for constructing a segmented network, however, their methods are based on simulation using network environmental data and attack threat data. The preparation of such data is not cost-effective. Our method is more effective from the viewpoint of cost and user convenience because it is based on the real network traffic data and coordinates the access control to the user's real traffic.

From the viewpoint of traffic investigation of the internal network, there are many researches for malware activity detection [15] [16]. However, recently the encryption of communication is often conducted, and malwares also encrypt their communication for avoiding detection. There are many researches for decryption of communication for detecting malware communication [17], however, it includes the problem of privacy. In our research, the proposed system can treat observed traffic regardless it is encrypted or not.

B. Automated ACL Generation System (AAGS)

This research focuses on the separated network, and we proposed several systems that support the separated network construction. An AAGS [4] evaluates the necessity of communication sections based on two criteria, i.e., access authority of a user to files or directories in servers and existing communication in an internal network.

Generally, the access authority of files or directories is strictly managed. For example, although all members can access public information, confidential information is managed for access by the only concerned person(s). Thus, the communication section between a user and a server is unnecessary if the user has no access authority to all files and directories in the server. Many organizations apply directory service for managing access authority, therefore, the system gathers information on access authorities by analyzing the information in the directory server in a network and evaluates the necessity of communication sections. By using the result of the evaluation, AAGS generates ACL automatically.

However, there are various types of necessary communication in a network except for file access communication. The system judges such legitimate communication as unnecessary if it evaluates based only on file access authority. To avoid such a situation and secure business continuity, AAGS analyzes the mirrored packets of the internal network. Before applying the ACL generated based on file access authorities, the system revises it by using mirrored packets. Even if communication was previously judged as unnecessary, its new observation calls reevaluation, and the communication is judged as necessary. Based on the judgment, the system regenerates ACL to permit all of the necessary communication sections. Thus, AAGS supports us to construct the separated network easily by applying the generated ACL.

C. Problems of the AAGS

Although AAGS can reduce the burden of administrators in constructing or managing the separated network, the generated ACL is not properly described because of the following two reasons.

First, AAGS judges all communications observed in the network as necessary even if they occur unintentionally, and it permits all of such communications. Therefore, generated ACL may include overly permission of unnecessary communication sections.

Second, the ACL only is based on the pair of source and destination IP addresses. Once the system judged the communication section to be allowed, all communication protocols on the pair are permitted.

III. Communication Classifying System (CCS)

To solve the problems of AAGS, we proposed a CCS [5] that improves the preciseness of ACL generated by AAGS.

A. System Overview

Previously, we made CCS conduct a two-step investigation. First, CCS investigates the consistency between the communication observed in the network and the reason AAGS permitted such communication section, i.e., user's file access authority or communication observation. When the observed communication does not relate to file sharing even though the file access authority is the reason permitting the communication section, such communication does not have consistency. If a communication lacks consistency, CCS performs the additional investigation. Because a legitimate communication assumes that its destination terminals listen to the appropriate port, the system performs a port scan to identify the listening port and then analyzes the correlation between the observed communication protocols and listening ports of destination terminals.

These investigations make CCS possible to detect illegal communication. Finally, the system generates a new ACL described by the sets of source and destination IP addresses and destination port to permit only legitimate communication and prohibiting unnecessary communication.

However, we noticed that the CCS makes misjudgments in several conditions. For example, the previous experiment made a client conduct illegal SMB communication to a file server. Although the client has no access authority to the server, CCS judged such communication as necessary. CCS judged such communication as necessary. Such misjudgments occur in the condition that a destination server provides the same service as illegal communication for specific users. For example, as shown in Figure 1, we assume a case that the server provides web services against only regular staff excluding part-time staff by using the authentication function. In this time, the server listens 80/tcp port for using the HTTP protocol. If a part-time staff accidentally communicates with the server via the HTTP protocol, the web service cannot be used because this server cannot be authenticated. In other words, this accidentally occurred communication is an unnecessary one. However, CCS verified such communication as a necessary one as the destination server opens the corresponding service port of the communication.

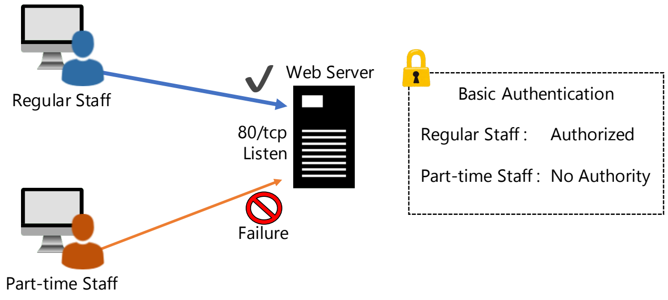


Fig. 1. Example of a condition in which judgment fails.

Moreover, this paper extends the CCS to conduct a three-step investigation by combining the conventional methods and the new statistical judgment method.

B. Assumption in CCS

We proposed CCS to complement our previous AAGS. CCS assumes that the network is roughly divided into several segments, and ACL generated by AAGS is applied to the network. The applied ACL is stored in a database (ACL DB) by AAGS.

ACL DB is extended by adding three new columns. First, we added "Permitted Reason" to register the reason why the communication is permitted, i.e., directory service information, or/and communication analysis. AAGS uses the extended versions of DB so that the ACL describes permitted communication sections, e.g., source IP addresses, destination IP addresses, and Permitted Reason. The remaining two columns are "Destination Port" and "Status". However, AAGS ignores these two columns as empty fields.

When CCS analyzes the communication section, it registers "analyzed" to the Status field of such a communication section. If there is only one record for the pair of source IP address and destination IP address, and such a record's Status field is empty, it is the first time for the proposed system to analyze that communication section. If "analyzed" has been registered to the Status field of a communication section, CCS omits the analysis of the communication section.

In addition to the above case where AAGS or CCS permitted the communication section, we assume the other case that different ways permit the communication section. For example, an administrator can permit any protocols manually. Furthermore, the research "Dynamic Access Control System" [18] associates with CCS to permit communication overly prohibited by AAGS. In these cases, the permitted communication section has not yet been analyzed by CCS. To distinguish the not analyzed communication, CCS assumes that "not_analyzed" is registered to Status filed of the communication section not permitted by AAGS or CCS. If the Status field is "not_analyzed", CCS analyzes the communication protocols in the section.

In this paper, to simplify the discussion, we assume that all terminals are statically assigned IP addresses, and such assignment information is managed in a directory server. However, our method can be easily applied to the environments that employ dynamically IP address assignment methods, e.g., DHCP. We can control the connected device's communication by identifying the device's user with any authentication method, e.g., IEEE 802.1X. For example, we can assign the appropriate VLAN that the user should belong to, or update ACL based on the assigned IP address.

C. The architecture of CCS

Figure 2 shows the architecture of CCS. The system consists of six modules and the extended database in the AAGS. The details of each module are described below.

1) Traffic Collector: This module receives all mirrored packets generated in the internal network. This paper assumes that the collection period of mirrored packets for investigation is statically defined in advance, e.g., 1 day, 1 hour, 10 minutes, and so on. After mirrored packets collection, the module generates a list of packet information including sets of source IP address, destination IP address, and destination port from the collected packet. The generated list of packet information is sent to the Consistency Judgment module.

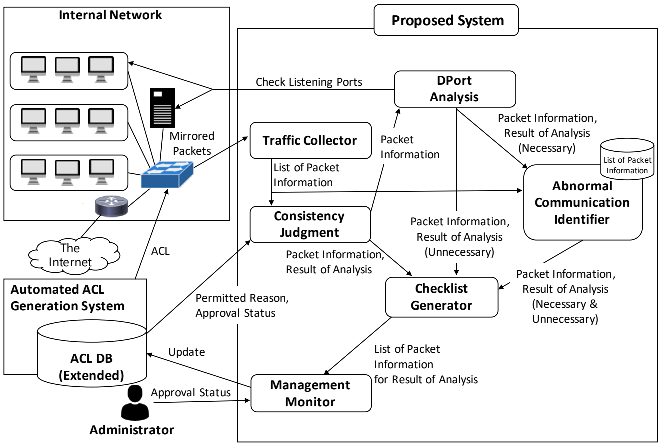

Internal Network
Proposed System
Check Listening Ports
DPort Analysis
Mirrored Packets
Traffic Collector
Packet Information
Packet Information, Result of Analysis (Necessary)
List of Packet Information
List of Packet Information
Abnormal Communication Identifier
Consistency Judgment
Packet Information, Result of Analysis (Unnecessary)
The Internet
ACL
Packet Information, Result of Analysis
Packet Information, Result of Analysis (Necessary & Unnecessary)
Automated ACL Generation System
Permitted Reason, Approval Status
Checklist Generator
ACL DB (Extended)
Update
List of Packet Information for Result of Analysis
Approval Status
Management Monitor
Administrator

Fig. 2. The architecture of a communication classifying system.

2) Consistency Judgment: First, when a list of packet information is received, this module searches ACL DB records for each communication section by specifying each pair of source and destination IP addresses. When the status field is empty, the Consistency Judgment module analyzes all protocols captured in that communication section.

After extracting the subject of the communication for investigation, the Consistency Judgment module judges the consistency of such communication. The module finds the permitted reason for such communication by checking ACL DB. As shown in Table I, there are six combinations of a collected packet and communication reasons. In the table, CA denotes communication analysis. Because AAGS checks the necessity of the file-sharing communication by using a Directory Service Information (DSI), the Consistency Judgment module classifies the captured communication as SMB protocol or Other Protocols. In this paper, we assume that only SMB is used as a file-sharing communication protocol. SMB uses multiple ports and protocols, e.g., 139/tcp and 445/tcp. To simplify the discussion, we express these sets of all ports by using the term "SMB protocol".

TABLE I
COMBINATIONS OF PERMITTED REASON AND COLLECTED PACKET.

Collected Packet	Permitted Reason		
	DSI	DSI+CA	CA
SMB	1	2	3
Other Protocol	4	5	6

For the SMB protocol, combinations 1 and 2 of Table I have consistency. To permit these communications, the Consistency Judgment module sends this Packet Information to the Check List Generator module. On the other hand, communication lacks consistency in combination 3 because communication of SMB protocol was observed even if there was no access authority by DSI. However, file sharing may be conducted among user's terminals directly without management by the directory server. In order not to prohibit such communication, the Consistency Judgment module sends this Packet Information to the DPort Analysis module for further verification.

Apart from SMB, only combination 4 lacks consistency in other protocols. The Packet Information of such communication is sent to the Checklist Generator module to prohibit such communication. Though combinations 5 and 6 have consistency, the module cannot determine the sameness of the communication protocol collected by the Traffic Collector and AAGS. The Packet Information of such communication is sent to the DPort Analysis module, which conducts a detailed investigation.

3) DPort Analysis: This module analyzes the normality of communication. We assume that the destination terminal listens to the correct port of service for communication. In such an assumption, the module judges the normality of communication using the current stand-by states of destination terminals. There are several ways to specify the listening ports of terminals. Thus, we adopt port-scanning against destination terminals in this paper.

Based on the result of port-scanning, when the destination port of communication listened to the destination terminal, the DPort Analysis verifies that communication is necessary. When the judgment result is necessary, Packet Information and the result are sent to the Abnormal Communication Identifier module.

On the other hand, if the destination port is blocked, the communication is judged as unnecessary. Thus, the judgment result is sent to the Checklist Generator module with its packet information.

4) Abnormal Communication Identifier: Abnormal Communication Identifier module finds out abnormal communication from the Packet Information judged as necessary communication by the DPort Analysis module. In this paper, we define abnormal communication as the unintentional user's communication against a server that provides service for limited users.

To find the abnormal communication, we use statistical analysis. The module receives a list of packet information from the Traffic Collector module and stores it. Against the stored information, the module statistical analysis and finds abnormal communication.

As a basic idea, we assume that abnormal communication is extremely less than legitimate communication. For example, as shown in Figure 3, there is a web server that is utilized for managers. Here, communication via HTTP protocol is allowed between the server and the managers. On the other hand, communication accidentally conducted by part-time staff is very few compared to legitimate communication. The difference in the amount of communication between legitimate and anomalous communication can make a significant difference. Therefore, we intend to judge such very few volume communications as abnormal communication.

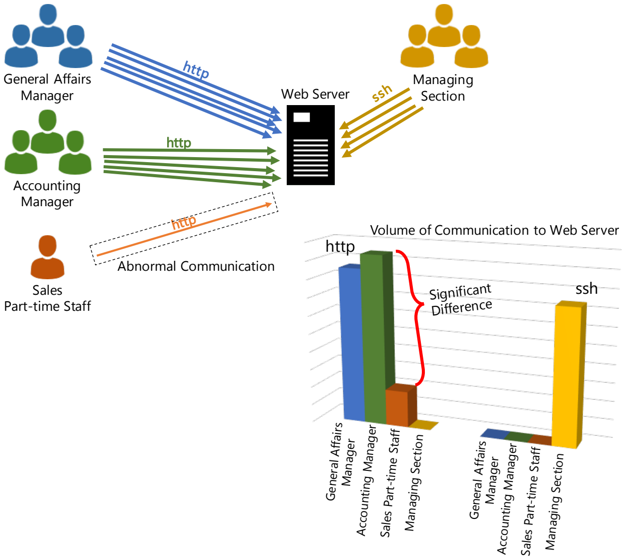


Fig. 3. The basic assumption of statistical analysis.

However, such a tendency of the communication volume depends on the communication protocol. When the web server shown in Figure 3 provides SSH for the system managing section, a lot of SSH, different from HTTP communication, is conducted only between the web server and the managing section.

From the viewpoint of this idea, we classify the communication based on the destination IP address and port number. When the Abnormal Communication Identifier receives the packet information to be verified from the DPort Analysis module, it extracts all the packet information with the same destination IP address and destination port number as the packet information to be verified. The module also counts the number of packets for each source IP address from all extracted packet information. We utilize these counted number of packets as data for statistical analysis.

Because various communication is allowed in the organization's network, and traffic volume depends on the communication content, we assumed that the data might contain an extremely large or small value. For such data, it is not appropriate to use non-robust statistics such as mean and standard deviation. In this paper, we adopt an outlier test using a quartile and an interquartile range (IQR) to consider robustness because a quartile and IQR are less affected by the size of the data values.

A quartile is a quantile dividing the data sorted in ascending order of value into four equal parts. The second quartile (Q_2) is the median, and it divides data into two equal parts. The first quartile (Q_1) is the median of data smaller than Q_2. It divides whole data into the lowest 25%. The third quartile (Q_3) is the median of data bigger than Q_2. It divides whole data into the lowest 75%. By using Q_1 and Q_3, we obtain IQR.

$$\mathrm{IQR} = Q_3 - Q_1 \tag{1}$$

By using a quartile and IQR, we set a threshold for the outlier test. In this paper, we apply a report of the tabulating statistical survey [19]. In [19], Noro and Wada pointed out that we can properly detect the outlier by using a threshold based on quartile and IQR even when the data does not follow a normal distribution. They used equation (2) for setting a lower limit of an appropriate range of data.

$$\mathrm{L} = Q_1 - 1.724 \times \mathrm{IQR} \tag{2}$$

However, the number "1.724" in the equation can be arbitrarily set according to the length of the tail of the distribution of the actual data. For simplicity of discussion, this section follows [19] and uses "1.724" without change.

In the communication data including extremely large or small values, IQR becomes so large that equation (2) cannot derive the threshold correctly. To solve such a problem, we convert all data in logarithm with base e in advance.

Because a threshold that shows a negative value cannot be properly treated, it is replaced with a positive value using an exponential conversion. Therefore, we earn a threshold by using equation (3) shown below.

$$\text{Threshold} = \begin{cases} L & (L >= 0) \\ e^L & (otherwise) \end{cases} \tag{3}$$

The Abnormal Communication Identifier module determines the communication is abnormal if the count is less than the threshold. If the abnormal communication is detected, the judgment result of this communication is changed as unnecessary. Finally, the results and Packet Information are sent to the Checklist Generator module.

5) Checklist Generator: This module receives the packet information and judgment results from the Consistency Judgment module or DPort Analysis module. The Checklist Generator module combines the packet information and analysis results and generates a checklist from this information for administrators. The generated checklist of the packet information is sent to the Management Monitor module.

6) Management Monitor: Lists of the packet information and judgment results are sent from the Checklist Generator module to the Management Monitor module. This module presents the received lists to the administrators. Administrators check the list and authorize the permission or prohibition of the communication section. Finally, the module updates the ACL DB to register the authorized packet information as "analyzed" value in the status field. After updating the ACL DB, the ACL Applier in AAGS applies it to the network.

IV. Implementation of the Proposed System

This section describes the implementation of the proposed system. The basic structure of the modules and the data flow among modules are shown in Figure 4. In the proposed system, the Traffic Collector module, the Consistency Judgment module, the DPort Analysis module, and the Abnormal Communication Identifier module run as batch processing written with Python. The list of observed packet information, in which the Abnormal Communication Identifier stores designed using MySQL database [20]. We adopt Node.js [21] as a Web server including the Checklist Generator module and the Management Monitor. Also, we designed an API server by using FastAPI [22] for smooth data exchanges between each module and the ACL DB or between the Abnormal Communication Identifier module and the list of packet information.

In this paper, we implemented ACL DB and ACL Applier that are included in AAGS. In addition to the list of packet information, we used MySQL for ACL DB. By using the Software Designed Networking (SDN) technique, we realized the ACL applier. We assume that Open vSwitch [23] (OvS) is used as a network switch, and we adopted Trema [24] as the SDN controller that instructs the OvS to control packets in the network.

Moreover, all of these modules run on Docker [25], which manages applications using a container type virtual environment.

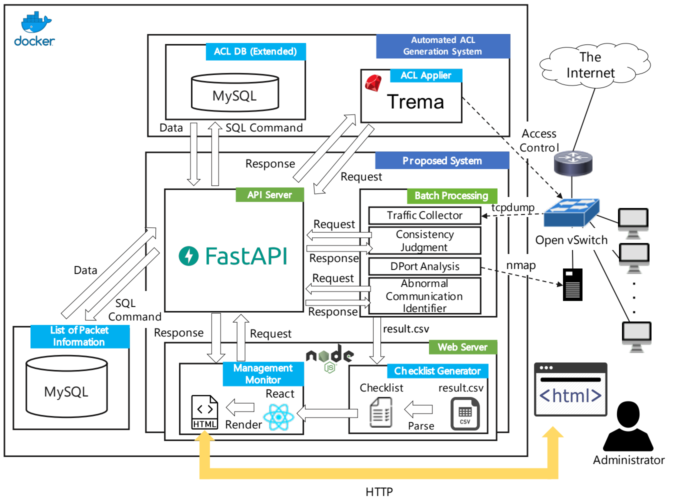


Fig. 4. System configuration diagram.

A. Traffic Collector

This module receives mirrored packets from the OvS and generates a list of packet information. We configure the OvS in advance to generate mirrors of all packets in the network and send them to the Traffic Collector. The Traffic Collector executes the `tcpdump` command and captures all of the mirrored packets sent from OvS for a collection period. In the experiment mentioned in Section V, we set a collection period as 1 hour or 30 minutes or 10 minutes.

The captured packets are saved as pcap files, and this module extracted sets of source IP address, destination IP address, and destination port for each packet from the saved pcap file by using dpkt [26], which is a module of Python. Finally, this module sends the extracted set as packet information to the Consistency Judgment module and Abnormal Communication Identifier module.

B. Consistency Judgment

After receiving the list of packet information, this module sends a request to the API server to search the record of the communication section in the ACL DB corresponding to each packet information. This module also checks the destination ports of each packet information and classifies them into SMB or other ports.

Further, the module compares such destination ports and the result of the record search, and it judges the consistency of the communication. When the module decides whether the observed communication is necessary or not, it sends the packet information of that communication section with the judgment results to the Checklist Generator module. On the other hand, if the module determines that the detailed analysis is necessary, it sends the packet information to the DPort Analysis module.

C. DPort Analysis

This module judges the normality of the communication included in packet information sent from the Consistency Judgment module. To assess the listening ports of destination terminals, it uses the `nmap` command. Here, we use the `-S` optional command of nmap to spoof the source IP address of the observed communication.

Based on the results of nmap, if the proper service port of packet information is listening at the destination terminal, the module judges this communication as rightful and necessary. Otherwise, the communication is judged as unnecessary.

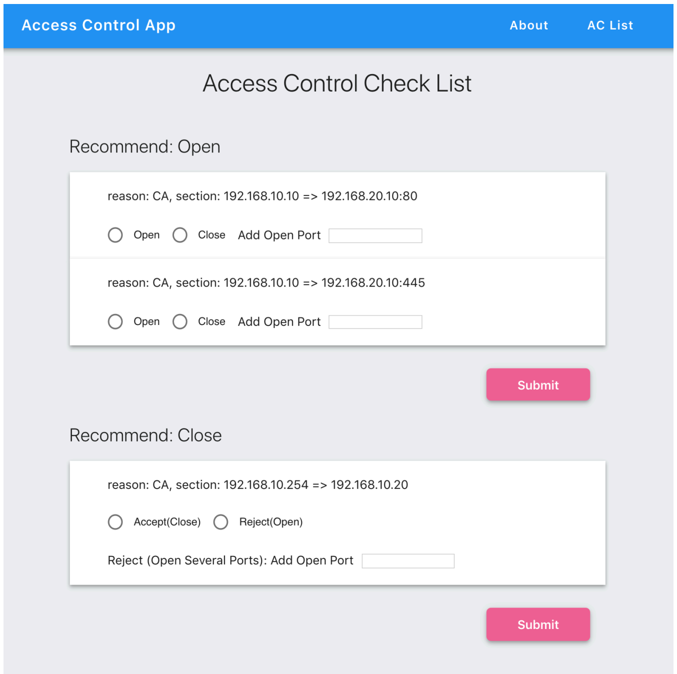
Moreover, if DPort Analysis determines the communication as unnecessary, the module sends the packet information and its judgment results to the Checklist Generator module, otherwise, communication is judged as necessary. Further, the packet information and its judgment results are sent to the Abnormal Communication Identifier module.

D. Abnormal Communication Identifier

This module receives all the list of packet information from the Traffic Collector module, and the module sends a request to the API server to store received packet information in the database. When the packet information and the judgment result are sent from the DPort Analysis module, the Abnormal Communication Identifier module sends a request to the API server to search the packets that have the same destination IP address and the same destination port of received packet information. By using receiving packet information and packet information found in the list of the packet information database, the module conducts the statistical analysis. Finally, the module sends packet information and judgment result to the Checklist Generator module.

E. Checklist Generator and Management Monitor

The Checklist Generator module receives the packet information and its judgment results from the Consistency Judgment module and the DPort Analysis module. The Checklist Generator combines these pieces of information about the packet and generates the checklist of the packet information.



Access Control Check List

Recommend: Open

reason: CA, section: 192.168.10.10 => 192.168.20.10:80

○ Open ○ Close Add Open Port

reason: CA, section: 192.168.10.10 => 192.168.20.10:445

○ Open ○ Close Add Open Port

Recommend: Close

reason: CA, section: 192.168.10.254 => 192.168.10.20

○ Accept(Close) ○ Reject(Open)

Reject (Open Several Ports): Add Open Port

Fig. 5. Sample of management monitor web page.

The generated list of packet information is sent to the Management Monitor. Then, based on the list, a html page is generated as an interface for administrators by using React [27]. Figure 5 shows a sample of the generated Web page for administrators.

On this screen, there are two sections. The first section is "Recommend: Open". The communication sections displayed in this section is judged as necessary. If the administrator judges it as appropriate, it can be authorized by selecting the "Open" button. However, only the displayed ports are judged necessary by the system, and all of the other ports not displayed will be prohibited. When administrators want to permit several ports in addition to the system recommendation, they can insert such ports into the "Add Open Port" form. Otherwise, they use the "Close" button to prohibit the displayed communication.

The second section is "Recommend: Close". The system judged communication displayed in this section as unnecessary. If the administrator selects the "Accept (Close)" button, all communication in this section is prohibited. On the other hand, when the "Reject (Open)" is selected, the ACL permits all communication in this section. Also, if the administrator wants to permit several ports in this section, such ports can be inserted into the "Add Open Port" form.

Finally, this module updates the ACL DB using the API server when the "Submit" button is clicked. As mentioned in the next Subsection IV-F, the ACL DB stores only permitted communication sections. In case of that all analyzed communication is judged as still permitted, the system updated the status field of the flow_list table about such communication section as analyzed. If only several ports are permitted, in addition to the above update, those ports are inserted into the dst_port field.

On the other hand, if all protocols in the communication section are judged as unnecessary, the module updates the ACL DB to delete any record of such a communication section in the section_list table.

F. ACL DB (Extended)

As described in Section III-C, we extended ACL DB. ACL DB consists of two tables, "section_list" and "flow_list" shown in Table II. The section_list table consists of four columns: "id", "src_ip", "dst_ip", and reason. The src_ip and the dst_ip store the source IP address and destination IP address of the communication section permitted by AAGS respectively. The reason column stores the permitted reason.

TABLE II
ACL DB (EXTENDED) TABLE SCHEMA.

Table Name	Column	Data Type	Example
section_list	id	Integer	3
	src_ip	String	192.168.10.10
	dst_ip	String	192.168.20.20
	reason	String	CA
flow_list	section_id	Integer	3
	dst_port	Integer	443
	status	String	analyzed

The flow_list table consists of three columns: "section_id", "dst_port", and "status". The value of section_id is corresponding to the id of the section_list table. Permitted destination ports in the communication section are stored in the dst_port column. If the communication section is permitted without analysis by CCS, "not_analyzed" is stored in the status column. After analysis by CCS, the value of status is updated to "analyzed".

G. ACL Applier

We use the SDN technique to implement the ACL Applier. The OvS is operating as a core switch in the network. We use Trema as an OpenFlow controller to apply the contents of ACL DB to the network.

V. Evaluation Experiment

We applied the implemented system to our prototype network and conducted an evaluation experiment on it.

A. Experimental Conditions

1) Network Structure: Here, we conducted experiments in our prototype network to verify the various situations by extending our prototype network, which is used for evaluation of the CCS [1]. As shown in Figure 6, the network has one sever segment and four client segments.

Each segment has seven or four Windows 10 PCs, and all PCs are assigned static IP addresses. Besides, two Windows Server 2019 terminals are located in the server segment. One of these servers works as a file server, and another server works as an active directory, which also has the role of DNS in this organization. The file server permits access only from the user whose position is the manager.

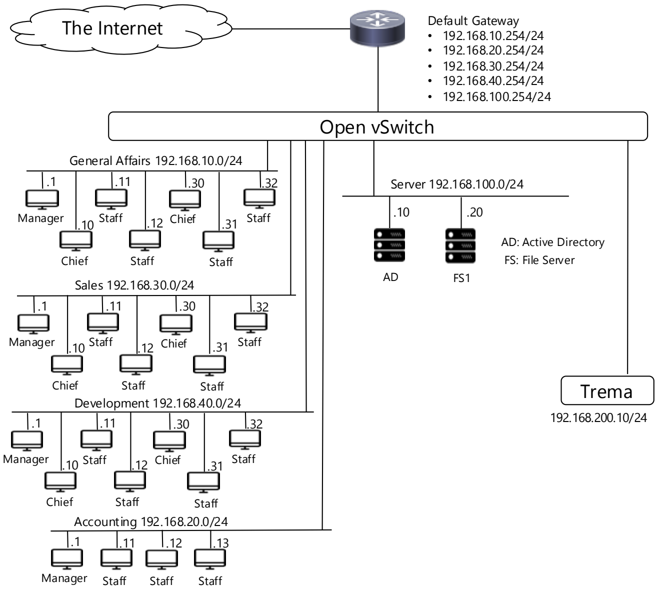


Fig. 6. Prototype network architecture.

2) Access Controls: The AAGS generated the ACL, and we prepared the ACL shown in Table III. We configured Trema to permit only the communication listed in Table III in addition to the communications between the default gateway and all the terminals. Open vSwitch, controlled by Trema, performs the access control.

TABLE III
LIST OF COMMUNICATION SECTIONS PERMITTED BY AAGS.

Source IP Address	Destination IP Address	Permitted Reason
192.168.10.1	192.168.100.20	DSI
192.168.20.1	192.168.100.20	DSI+CA
192.168.20.11	192.168.100.20	CA
192.168.30.1	192.168.100.20	DSI+CA
192.168.40.1	192.168.100.20	DSI

Because the managers have access authority to the file server, 192.168.10.1, 192.168.20.1, 192.168.30.1, and 192.168.40.1 are permitted to communicate with 192.168.100.20, and we insert "DSI" as Permitted Reason in the records of these communication sections.

In addition to the file access communication, unintended communication from 192.168.20.1 and 192.168.30.1 to 192.168.100.20 are conducted, and "CA" is added to Permitted Reason of that section. Similarly, communication between 192.168.20.11 and 192.168.100.20 is permitted because of unintended communication, and "CA" is registered as its Permitted Reason.

B. Experiment 1: Judgments of All Communication

To evaluate the effectiveness of CCS, we run CCS to collect and judge all communication in the prototype network. The experiment was performed according to the following procedure.

Step 1: Run the proposed system and start to collect mirrored packets in the network. In this experiment, we set the collection period to be 1 hour.

Step 2: In the collection period, terminals, 192.168.10.1, 192.168.20.1, 192.168.30.1, and 192.168.40.1, access the file server using the SMB protocol. Also, the terminal of 192.168.20.11 that has no access authority tried SMB protocol communication with the file server. Though the file server does not provide HTTP service, HTTP protocol communication to the file server is conducted by terminals 192.168.20.1 and 192.168.30.1. In addition, all nine client terminals access external sites on the Internet that are assuming the activities of the organization.

Step 3: After 1 hour, the collection period ends, and the captured packets are analyzed by the proposed system. Based on the analysis result, the system generates the checklist and prepares the Web page.

Step 4: We check the result of the analysis by the proposed system on the Web page and authorize them.

Step 5: Finally, the system applies the authorized ACL to the internal network.

C. Results of Experiment

The result of the analysis using the proposed system is shown in Table IV. The legitimate SMB communication from 192.168.10.1, 192.168.20.1, 192.168.30.1, and 192.168.40.1 to the file server (192.168.100.20) is correctly judged as necessary. Also, the system judges the DNS protocol communication as necessary. Moreover, the system judges several communication sections including the unintended HTTP communication and high port number communication that seems to be returned packets as unnecessary.

Thus, the above results are approximately the same as the previous CCS's results. In this evaluation, we focus on communication from 192.168.20.11 to 192.168.100.20. The previous CCS judges the communication as necessary because the destination port has listened. On the other hand, the extended CCS judged such illegal communication as unnecessary as the "Outlier" is shown in the result of the analysis.

D. Experiment 2: Using Several Patterns of SMB Communication

In the experiment using all communication, the proposed system found abnormal communication correctly. To verify the credibility of the statistical judgment method, we further conducted another experiment. In this experiment, as same as experiment 1, four legitimate client terminals and one illegal client terminal tried to communicate with the file server. However, we conducted several patterns of experiments with different access counts or observation times. We only focus on these file-sharing communications and show the detailed process of the statistical analysis.

We generated different four access patterns shown in Table V. First, in pattern 1, legitimate clients frequently communicate with the file server, and the illegal client also conducts communication most frequently. We set the observation time as 30 minutes. In pattern 2, legitimate clients conduct communication as same as pattern 1. In contrast to pattern 1, the illegal client tried to communicate only once in 30 minutes. We set different observation time in pattern 3. Similar to pattern 2, the illegal client tried to communicate only once in 10 minutes. Finally, in pattern 4, all terminals randomly communicate with the file server at the same time. However, in all patterns, all terminals share different files with the file server. Even if all file-sharing communication is conducted at the same time, traffic volumes of each communication are different because of the file size.

TABLE IV
ANALYSIS RESULT BY OUR PROPOSED SYSTEM.

Internal Network Communication that Occurred			Result of Analysis
Source IP Address	Destination IP Address	Destination Port	
192.168.10.1	192.168.100.20	445	Open
192.168.20.1	192.168.100.20	445	Open
192.168.20.11	192.168.100.20	445	Outlier
192.168.30.1	192.168.100.20	445	Open
192.168.40.1	192.168.100.20	445	Open
192.168.10.1	192.168.100.10	53	Open
192.168.10.10	192.168.100.10	53	Open
192.168.20.1	192.168.100.10	53	Open
∼	∼	53	Open
192.168.20.1	192.168.100.20	80	Close
192.168.30.1	192.168.100.20	80	Close
192.168.100.20	192.168.10.1	63221	Close
192.168.100.20	192.168.20.1	59012	Close
192.168.100.20	192.168.20.11	55658	Close
192.168.100.20	192.168.30.1	52796	Close
192.168.100.20	192.168.40.1	51166	Close
192.168.100.10	192.168.10.1	63205	Close
192.168.100.10	192.168.10.10	65180	Close
192.168.100.10	192.168.10.11	61426	Close
∼	∼	∼	Close

In Table VI, the upper rows of each pattern show the number of observed packets, and the data for the statistical analysis that is generated by converting the number of observed packets in logarithm with base e is shown in the lower row.

Table VII shows the results of the statistical analysis. In patterns 1, 2, and 4, illegal communication's data for statistical analysis is judged as outlier correctly. On the other hand, only in pattern 3, the illegal communication's data (4.190) exceeds the threshold (3.221), and no outlier value was detected.

E. Discussion

From the results of experiment 1 and experiment 2, we found that the proposed system correctly judged the illegal SMB communication from 192.168.20.11 to 192.168.100.20 as unnecessary. Therefore, the judgment accuracy of the extended CCS is improved compared to the previous CCS that previously judged the communication as necessary.

In the case of pattern 1 in experiment 2, we expected that to conduct judgment correctly might be difficult because the illegal terminal generated communication most frequently in all terminals. However, correct judgment was conducted by the proposed system. As shown in Table VI, the observed number of packets for each terminal is different. The variation in the number of packets occurred because the traffic volume depends on the size of sharing files. Also, only traffic of protocol negotiation occurs in illegal communication. In other words,

TABLE V
ACCESS PATTERNS.

	192.168.10.1	192.168.20.1	192.168.30.1	192.168.40.1	192.168.20.11
Pattern 1 (30m)	Once / 20s	Once / 30s	Once / 15s	Once / 25s	Once / 5s
Pattern 2 (30m)	Once / 20s	Once / 30s	Once / 15s	Once / 25s	Once / 30m
Pattern 3 (10m)	Once / 20s	Once / 6s	Once / 7s	Once / 27s	Once / 10m
Pattern 4 (30m)	Random (All terminals communicate at the same time)				

TABLE VI
NUMBER OF OBSERVED PACKETS AND DATA USED FOR STATISTICAL ANALYSIS.

	192.168.10.1	192.168.20.1	192.168.30.1	192.168.40.1	192.168.20.11
Pattern 1 (30m)	1779 Packets	1242 Packets	1758 Packets	3236 Packets	521 Packets
	7.484	7.124	7.472	8.082	6.256
Pattern 2 (30m)	5942 Packets	1458 Packets	789 Packets	3104 Packets	48 Packets
	8.690	7.285	6.671	8.040	3.871
Pattern 3 (10m)	414 Packets	2691 Packets	1147 Packets	2075 Packets	66 Packets
	6.026	7.898	7.045	7.638	4.190
Pattern 4 (30m)	3544 Packets	3552 Packets	3230 Packets	5780 Packets	654 Packets
	8.173	8.175	8.080	8.662	6.483

TABLE VII
RESULT OF STATISTICAL ANALYSIS.

	Q_1	Q_3	IQR	Threshold	Outlier
Pattern 1 (30m)	7.124	7.484	0.360	6.500	6.256
Pattern 2 (30m)	6.671	8.040	1.369	4.311	3.871
Pattern 3 (10m)	6.026	7.638	1.612	3.247	N/A
Pattern 4 (30m)	8.080	8.175	0.095	7.916	6.483

no actual file-sharing communication has occurred between 192.168.20.11 and 192.168.100.20, and such existence of the file-sharing in a series of communication makes a significant difference to detect as the outlier.

In pattern 2 and pattern 3, illegal communication was conducted only once in the experiment, and no communication was detected as an outlier in pattern 3. In pattern 3, we purposely set the collection period of the packet as a short time. As a possible reason for false negative judgment, therefore, the observation time of pattern 3 was too short to collect enough data for statistical analysis. To verify this reason, when we calculate with the tripled number of legitimate packets assuming the collection period is 30 minutes, the threshold becomes "4.345", and we can correctly judge the data of the illegal communication "4.190" as the outlier. Besides, the pattern 2, which applied observation time as 30 minutes, although the communication was conducted in similar trends with pattern 3, illegal communication was detected as an outlier correctly.

In pattern 4, though all terminals conducted communication with the same number of times, the difference occurs in the number of packets for the same reason as pattern 1. Hence, this method can judge the necessity of communication correctly.

In summarize, when sufficient data is obtained by observing the packets for a certain period, we can detect illegal communication that is misjudged by the previous CCS by using statistical analysis, and the proposed system judges communication correctly. Thus, the proposed system can make a judgment with higher accuracy than the previous CCS.

However, in these experiments, we did not conduct parameter tuning for deriving threshold, and we can detect outlier correctly as explained in Table VII. Also, the problem of lack of data due to short observation time can be solved by adjusting the parameter. For example, to increase the value of the parameter, we change the parameter from "1.724" to "1.100". Following this change, each threshold of experiment 2 varies as shown in Table VIII. In this situation, the proposed system detects outlier correctly in all patterns with no false positive.

TABLE VIII
PARAMETER TUNING OF THRESHOLD.

	Threshold (1.724)	Threshold (1.100)	Outlier
Pattern 1 (30m)	6.500	6.728	6.256
Pattern 2 (30m)	4.311	5.165	3.871
Pattern 3 (10m)	3.247	4.253	4.190
Pattern 4 (30m)	7.916	7.976	6.483

In the result of experiment 1, the system displayed a lot of communication judgment between all client terminals, and the router, which is the default gateway of each segment. All these communications are like returned packets. We should not prohibit the returned packets, so these communications should be ignored by the system. We have already pointed out this problem in [1] and listed it as future work to distinguish whether high port communication is legitimate or not.

In the proposed system, the Abnormal Communication Identifier module has all the list of the packet information, and we consider that it can solve the problem. In this paper, we focus on the statistical analysis, and we designed the proposed system in which only communication judged as necessary by the DPort Analysis module is sent to the Abnormal Communication Identifier module to simplify the discussion of the module.

To distinguish the legitimate returned packets and illegal packets, we change the DPort Analysis module to send all results to the Abnormal Communication Identifier module.

When the Abnormal Communication Identifier module receives judgment targets from the DPort Analysis module, it first checks the judgment results. If the judgment result is necessary, it conducts statistical analysis as explained in Section III-C. Otherwise, no statistical analysis should be conducted and put it in the list of judgment targets. Finally, it conducts analysis using the following procedure.

Step 1: Receive all judgment targets that are judged as "Unnecessary" by the DPort Analysis module.

Step 2: Check the destination port number. If it is the well known port or registered port, the target is judged as unnecessary. If it is the dynamic/private port, go to step 3.

Step 3: Check whether there is the outward communication that has the same number in source port as the destination port number of judgment target. If there is no such communication, the target is judged as unnecessary. If there is such communication, go to step 4.

Step 4: Finally, check the found outward communication. If it is judged as necessary, the target is judged as necessary. If it is judged as unnecessary, the target is judged as unnecessary.

VI. Conclusion

In this paper, we extended our previous CCS to solve the problem of it. The previous CCS misjudges abnormal communication as a necessary communication when the destination terminal listens the communication's destination port. We assumed that it is possible to distinguish legitimate communication and abnormal communication by statistical analysis of network traffic volume. Therefore, we adopt a statistical analysis for indicating abnormal communication, which is misjudged as a necessary communication by the previous CCS. We extended the CCS to perform statistical analysis in addition to the previous CCS's analysis. We implemented extended CCS and applied it to a prototype network. In the experiment, the system judged the necessity of communication observed in the network correctly, and we confirmed that the previous CCS problem was solved. As a result, we confirmed the feasibility of the proposed system.

However, because we adopted a statistical analysis, we need a mirrored packet to ensure the significant difference of packet volume between legitimate communication and illegal communication exists. Also, we need to set an appropriate packet collection period according to the traffic volume in the organization's network.

In the experiment, we applied the SDN technique for constructing a network. Our proposal dynamically changes ACL according to the observed traffic, therefore, SDN is one of the best ways to implement the separated network with our method. Constructing an organization network with SDN network equipment is not cost-effective. However, recently SDN has become a more common technology. Recently, there are many kinds of research focusing on SDN technology [28] [29]. Therefore, it can be an expected improvement in the cost of SDN in the near future.

As future works, we will extend the system to apply more complicated environments. Nowadays, the COVID-19 dramatically changes human's work style, and a lot of organizations all over the world adopt working from home. In this situation, many clients connect to the internal network resources from the outside network via VPN and so on. To maintain the security of the organization's network, we have to take into account such outside network devices for constructing a secure internal network.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Number JP19K20268. The authors would like to thank Enago (www.enago.jp) for the English language review.

References

[1] Y. Sato, H. Hasegawa, and H. Takakura, "An Evaluation on Feasibility of a Communication Classifying System," *Proceedings of The Thirteenth International Conference on Emerging Security Information, Systems and Technologies*, pp. 9–15, 2019.

[2] P. Cichonski, T. Miller, T. Grance, and K. Scarfone, "Computer Security Incident Handling Guide," *NIST Special Publication 800-61 Revision 2*, 2012, NIST SP800-61 Rev.2.

[3] J. Information-technology Promotion Agency, "Design and Operational Guide to Protect against "Advanced Persistent Threats" Revised 2nd edition," 2011, URL: https://www.ipa.go.jp/files/000017299.pdf [accessed: 2020-12-08].

[4] H. Hasegawa, Y. Yamaguchi, H. Shimada, and H. Takakura, "An Automated ACL Generation System using Directory Service Information and Network Traffic Data (in japanese)," *The IEICE Transactions on Information and Systems (Japanese Edition)*, vol. J100–D, no. 3, pp. 353–364, 2017.

[5] Y. Sato, H. Hasegawa, and H. Takakura, "Construction of Secure Internal Networks with Communication Classifying System," *Proceedings of the 5th International Conference on Information Systems Security and Privacy*, vol. 1, pp. 552–557, 2019.

[6] G. Alessandro, P. Giovanni, C. Alberto, and B. Giuseppe, "Advanced widespread behavioral probes against lateral movements," *International Journal for Information Security Research*, vol. 6, pp. 651–659, 2016.

[7] T. Watanabe, T. Kitazaki, T. Ideguchi, and Y. Murata, "A Proposal of Dinamic VLAN Configuration with Traffic Analyzation and Its Evaluation Using a Computer Simulation (in Japanese)," *IPSJ Journal*, vol. 46, no. 9, pp. 2196–2204, 2005.

[8] A. K. Nayak, A. Reimers, N. Feamster, and R. Clark, "Resonance: Dynamic Access Control for Enterprise Networks," *Proceedings of the 1st ACM SIGCOMM 2009 Workshop on Research on Enterprise Networking*, pp. 11–18, 2009.

[9] T. Miyamoto, T. Tamura, R. Suzuki, H. Hiraoka, H. Matsuo, and et al., "VLAN Management System on Large-scale Network (in Japanese)," *Transactions of Information Processing Society of Japan, IPSJ Journal*, vol. 41, no. 12, pp. 3234–3244, 2000.

[10] N. Gude, T. Koponen, J. Pettit, B. Pfaff, and M. Casado, "Nox: Towards an operating system for networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 3, pp. 105–110, 2008.

[11] "VLAN .Config," 2019, URL: http://www.iiga.jp/solution/config/vlan.html [accessed: 2020-12-08].

[12] M. Mujib and R. F. Sari, "Design of implementation of a zero trust approach to network micro-segmentation," *International Journal of Advanced Science and Technology*, vol. 29, no. 7 Special Issue, pp. 3501–3510, 2020.

[13] N. Wagner, C. Ş. Şahin, M. Winterrose, J. Riordan, J. Pena, D. Hanson, and W. W. Streilein, "Towards automated cyber decision support: A case study on network segmentation for security," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2016, pp. 1–10.

[14] N. Wagner, C. Ş. Şahin, J. Pena, and W. W. Streilein, "Automatic generation of cyber architectures optimized for security, cost, and mission performance: A nature-inspired approach," in *Advances in Nature-Inspired Computing and Applications*. Springer, 2019, pp. 1–25.

[15] M. Akbanov, V. G. Vassilakis, and M. D. Logothetis, "Ransomware detection and mitigation using software-defined networking: The case of wannacry," *Computers & Electrical Engineering*, vol. 76, pp. 111–121, 2019.

[16] A. Bohara, M. A. Noureddine, A. Fawaz, and W. H. Sanders, "An unsupervised multi-detector approach for identifying malicious lateral movement," in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 2017, pp. 224–233.

[17] T. Radivilova, L. Kirichenko, D. Ageyev, M. Tawalbeh, and V. Bulakh, "Decrypting ssl/tls traffic for hidden threats detection," in *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. IEEE, 2018, pp. 143–146.

[18] S. Nakamura, H. Hasegawa, Y. Tateiwa, H. Takakura, Y. Kim, and et al., "A Proposal of Dynamic Access Control with SDN for Practical Network Separation," *IEICE Technical Report*, vol. 117, no. 299, pp. 65–69, 2017.

[19] T. Noro and K. Wada, "A univariate outlier detection manual for tabulating statistical survey (in japanese)," *Research memoir of the statistics*, no. 72, pp. 41–53, 2015.

[20] "MySQL," 2019, URL: https://www.mysql.com [accessed: 2020-12-08].

[21] "Node.js," 2019, URL: https://nodejs.org/ [accessed: 2020-12-08].

[22] "FastAPI," 2019, URL: https://fastapi.tiangolo.com [accessed: 2020-12-08].

[23] "Open vSwitch," 2019, URL: https://www.openvswitch.org [accessed: 2020-12-08].

[24] "Trema," 2019, URL: https://trema.github.io/trema [accessed: 2020-12-08].

[25] "Docker," 2019, URL: https://www.docker.com [accessed: 2020-12-08].

[26] "dpkt," 2019, URL: https://dpkt.readthedocs.io/en/latest/ [accessed: 2020-12-08].

[27] "React," 2019, URL: https://reactjs.org [accessed: 2020-12-08].

[28] F. Kuliesius and V. Dangovas, "Sdn enhanced campus network authentication and access control system," in *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2016, pp. 894–899.

[29] F. Nife, Z. Kotulski, and O. Reyad, "New sdn-oriented distributed network security system," *Appl. Math. Inf. Sci*, vol. 12, no. 4, pp. 673–683, 2018.