Regular Paper

On-demand Suspicious Host Isolation Adopting Software Defined Network Approach on a Computer Security Incident Response

Motoyuki Ohmori[1,a] Masayuki Higashino[1,b] Toshiya Kawato[1,†1,c] Satoshi Fujio[1,d]
Kiyoyuki Nakashima[1,e]

Received: June 25, 2018, Accepted: December 4, 2018

Abstract: Computer security has been getting more attention because a computer security incident may cause great damage on an organization. A quick and correct response against an incident is then important. One of the first possible responses is then locating and isolating a suspicious host. This isolation typically requires a manual operation that may cause a mistake or long delay. In order to solve these issues, this paper proposes a novel system to locate and isolate a suspicious host on an incident response adopting the Software Defined Network (SDN) approach. This SDN approach allows the proposed system to locate and isolate a suspicious host on-demand in a network that comprises different switches and routers of different makers. The proposed system then requires no host authentication configured, no IP address allocation/assignment database, no network topology map and no switch port list in advance. The proposed system, therefore, can reduce human manual operations. This paper then presents that human manual operations actually induce longer delays, more than 3 minutes on average, and also cause mistakes. This paper also presents that the proposed system can locate and isolate a suspicious host within 10 seconds right after an IP address of a suspicious host is given.

Keywords: Computer security, Incident response, On-demand host isolation, Software Defined Network

1. Introduction

Computer security has been getting more attention because a computer security incident may cause great damage on an organization. Since it is difficult to avoid all incidents to happen, a proper and quick response against an incident is important in order to mitigate or minimize damage. The first possible response against an incident is to isolate a *suspicious* host that is observed to behave to compromise security, e.g., communicate with a malicious host such as a Command and Control (C&C) server and The Onion Router (Tor) [1]. This isolation may be initiated as follows. In many cases, a malicious communication is detected by an external organization such as Japan Security Operation Center (JSOC) [2] operated by LAC Co., Ltd., National Institute of Informatics Security Operation Collaboration Services, the so-called NII-SOCS, operated by National Institute of Informatics (NII) [3], government organizations or others. An organization then firstly recognizes a computer security event after receiving an alert of a suspicious communication from an external organization. The organization then makes a triage decision whether the event should be handled as an incident or not. If the event is considered as an incident, the organization then initiates an incident response. An operator in the organization then manually locates and isolates a suspicious host. These location and isolation apparently rely on human operations and operator capability. These location and isolation also load more operations on an operator, and may induce a mistake or a longer delay on an incident response. For example, it may require many operations to build a host database, which includes an IP address allocation/assignment database, a network topology map and so on. It may be also difficult to maintain and keep the database up-to-date. An operator may make a mistake, e.g., isolating an unsuspicious host. In addition, an operator may forget to share information such as who did what operation and when. This unshared information confuse other operators, e.g., operators other than an operator, who isolates a suspicious host, cannot revert the isolation on a recovery of an incident. On the other hand, a contact person, who is in charge of a management of a suspicious host, may be unable to be immediately contacted due to a business trip or day off. The suspicious host may be then unable to be located, and it may take more than hours to isolate the suspicious host. In order to avoid these mistakes or longer delay, dependencies on human operations must be excluded as much as possible toward the end of the era that relies on human operator's ad-hoc solutions.

To this end, this paper proposes a novel system to locate and isolate a suspicious host on an incident response. This system requires no host authentication, no IP address allocation/assignment database, no network topology map and no switch port list in advance. This system then takes a Software Defined Network (SDN) like approach. To be more specific, this

[1] Tottori University, Tottori 680–8550 Japan
[†1] Presently with National Institute of Technology, Yonago College
[a] ohmori@tottori-u.ac.jp
[b] higashino@tottori-u.ac.jp
[c] t.kawato@tottori-u.ac.jp
[d] s-fujio@tottori-u.ac.jp
[e] nakashima@tottori-u.ac.jp

system employs the Link Layer Discovery Protocol (LLDP) [4], which is also employed by a major SDN protocol, OpenFlow [5], or a similar protocol, Cisco Discovery Protocol (CDP), to automatically compute a network topology. This SDN-like approach allows an *on-demand* host isolation in a network where there are different switches and routers of different makers and multiple accounts are defined for access controls. This SDN-like approach can also reduce a control traffic, CPU load on a switch and storage consumption. An operator then needs to be given an IP address of a suspicious host, and just executes a script to isolate a suspicious host. After an isolation finishes, this system automatically reports its result, i.e., a successful finish or finish with an error, to all operators involved. This system also presents an operator with how to revert an isolation, and an operator just executes a script presented by this system on a recovery. In addition, this system supports a host that frequently moves and its IP address changes. This system also guards against mistakes that wrongly isolate a non-suspicious host or network.

The contributions of this paper can be summarized as follows:

- an operator can manually locate a suspicious host within approximately 3 minutes on average right after an IP address of the host is given,
- the time and a correctness to manually locate and isolate a suspicious host heavily depend upon operator capability,
- the proposed system can locate and isolate a suspicious host within 10 seconds,
- the proposed system can locate and isolate a suspicious host within 10 minutes in an actual environment right after an organization recognizes an event,
- on-demand SDN-like approach requires no host database, and can reduce a control traffic, CPU load on a switch and storage consumption to maintain a location of a host in comparison with a traditional approach using Simple Network Management Protocol (SNMP) [6],
- the proposed system can support a network, especially where network equipment is being replaced, that comprises different switches and routers of different makers while other existing systems cannot,
- the proposed system supports both of LLDP and CDP to find a neighboring router or switch especially for long-term network equipment replacement while other existing systems do not,
- the proposed system supports a router or switch implementing neither of LLDP nor CDP while other existing systems assume they are implemented,
- the proposed system can isolate a suspicious host that frequently moves and connects to a different switch,
- 15 ways to isolate a suspicious host are presented and discussed, and it is not enough for a recent malware such as *WannaCry* to just filter out a traffic to/from the Internet.

The rest of this paper is organized as follows. Section 2 defines and clarifies terminologies used in this paper. Section 3 states problems more in detail that motivate us to propose an on-demand suspicious host isolating system. Section 4 proposes an on-demand suspicious host isolating system. Section 5 then presents our first prototype implementation. Section 6 evaluates the proposed system. Section 7 discusses various case studies on isolating a suspicious host. Section 8 refers to related work, and clarifies differences between this paper. Section 9 finally concludes this paper.

2. Terminology

This section defines terminologies in this paper for clarification as follows.

- Event: an observed anomalous behavior. An event can also be an incident.
- Triage: making a decision whether an event should be handled as an incident or not.
- Incident: a special event confirmed to compromise security. An incident may cause a significant disruption of business.
- Incident response: an initial technical countermeasure against an incident. An incident response in this paper refers to locating and isolating a suspicious host from a network, and other responses are out of scope of this paper.
- Switch: a network switch. A switch in this paper refers to a layer-2 switch only and not a layer-3 switch for simplicity.
- Router: a network router. A router in this paper includes a layer-3 switch.
- SDN: Software Defined Network. A general programmable network, not limited to OpenFlow [5].
- Host database: A database comprises an IP address allocation/assignment database, a network topology map, switch port lists, and must be transversely referred by persons involved in an incident.
- IP address allocation: allocating an IP address block to a department or laboratory.
- IP address assignment: choosing an IP address from an allocated IP address block, and assigning the IP address to a host.

3. Motivation

This section states problems in locating and isolating a suspicious host, which motivate authors to propose the system and are solved in this paper.

3.1 Dependency on Operator Capability

It heavily depends on operator capability to manually locate and isolate a suspicious host. For example, an operator in an organization may manually locate and isolate a suspicious host as follows:

(1) identify a department using an IP address of a suspicious host from an IP address allocation/assignment database,
(2) locate a switch and port accommodating the suspicious host from a network topology map, Address Resolution Protocol (ARP) [7] address table or MAC address table, and
(3) shut down the port or filter out the MAC address.

Regarding (1), an operator who works longer for an organization may memorize an allocation of an IP address block to a department, and the operator can identify the department faster than other operators. Similarly, regarding (2), the operator can locate a switch and port faster than other operators. Regardless of years of continuous employment of an operator, each operator may work in a different place. An operator may then know switches well that are installed in nearer places where the operator works while the operator may not know other switches installed in other places. In other words, each operator has different knowledge about each switch. In addition, some operators may not know how to locate and isolate a suspicious host. Regarding (3), again, some operators may not know how to shut down a port or filter out a MAC address, and how to decide which operation is appropriate.

Nobody makes no mistakes, and an operator sometimes makes a mistake. On a response against an incident, an operator may then make a mistake with higher probability than on a usual operation since the operator is rushed to more quickly respond.

3.2 Quick Response and Human Operation Delay

A quicker response against an incident is better because a quick response may avoid compromising security and reduce operations. For example, a quick host isolation may avoid compromising confidential information. A quick host isolation also reduces operations to check if confidential information is compromised or not. If a suspicious host is not quickly isolated from a network, the host may continue to initiate new communications. In order to make sure that confidential information is not compromised, all communications must be investigated. A quicker host isolation, therefore, can reduce more operations.

A quick host isolation can also avoid a pandemic or epidemic of a malware. For example, *WannaCry* exploits a vulnerability of Server Message Block (SMB) protocol [8], and spreads a malware into other hosts within the same network. In order to avoid secondary infections within the same network, a quick host isolation is necessary.

In addition, a quick response is necessary for a mobile host. A suspicious host may leave a network before the host is located. A quick host locating is necessary.

On the other hand, a manual human operation requires more delay than an automated operation in general. The authors, indeed, have experienced that it took more than 10 minutes to locate and isolate a suspicious host. In order to reduce a delay, an automated operation would be better.

3.3 Building and Maintaining Host Database

Regarding locating a host in a network, it might be considered to build a *host database*. A host database usually comprises an IP address allocation/assignment database, a network topology map and switch port lists. It is, however, difficult to build a host database that persons involved in an incident can transversely refer over an organization. For example, each department builds own IP address allocation database in the authors' organization, and the database cannot be referred by others. An IP address assignment database may be then built by each laboratory, and cannot be shared among persons involved. In addition, an unified format for these databases is not defined, and its format may depend upon each person who is in charge of managing a database in each laboratory. Under these circumstances, it is difficult for authors to immediately build a host database.

Regarding a host database, it may be also difficult to keep the database up-to-date. This is because it is no incentive for a user to update a database. In authors organization, it is not technically prohibited to assign an IP address that a department or laboratory does not authorize. A user can then intentionally or unintentionally assign an unauthorized IP address to a host, and a host can then communicate. It can be said that a host database may not reflect an actual IP address assignment, and may be useless for a quick response against an incident. In order to solve these issues, one may be able to technically prohibit a host assigned to an unauthorized IP address from communicating. It may be, however, not feasible because a host authentication must be deployed at all ports of all switches in an organization.

3.4 Building and Maintaining Switch Port List

Regarding locating a host in a network, a *switch port list* should be maintained. In authors' environment, there are more than 10,000 ports of about 300 edge switches, there is no switch port list that records which port is connected to which host. We have been trying to build a switch port list from scratch, have not yet finished. Even after finishing it, it may be difficult for us to maintain the switch port list up-to-date. It would be better to have an on-demand way to locate a suspicious host.

3.5 Inefficiency of Traditional Methods

There are commercial systems to locate and isolate a suspicious host [9]. They, however, employ a traditional method using SNMP and periodically poll a ARP table and MAC address table from a router and switch, respectively. This method is apparently not scalable in terms of the number of network equipments. For example, a polling interval of SNMP is usually 5 minutes. Each polling is done for each switch and router in a network. In authors' environment, there are more than 300 network edge switches. In addition, commercial systems need a large amount of storages [9], and this may not be scalable. Moreover, authors have experienced that a core router stalls and cannot forward any IP traffic when not so many SNMP packets are received [10]. As described above, traditional methods using SNMP should be avoided.

3.6 Different Login Methods for Network Equipment

In authors' environment, there are different switches and routers of different makers because different vendors installed. Similarly, there are also different login methods for switches and routers. Some switches can be logged in using telnet while others can be using Secure SHell (SSH) or both. In addition, we have multiple accounts for network equipment. For example, core, departmental backbone and edge switches need different accounts for security policy, respectively. In this environment, it is very difficult for operators to quickly and appropriately login and operate a switch or router.

4. On-demand Suspicious Host Isolation

This section proposes the on-demand suspicious host isolating system for an incident response adopting SDN-like approach. The proposed system automatically locates and isolates a suspicious host from a network. The proposed system comprises

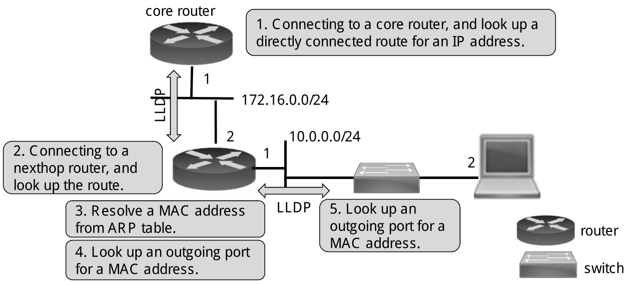
core router
1. Connecting to a core router, and look up a directly connected route for an IP address.
1
LLDP
172.16.0.0/24
2
10.0.0.0/24
1
2
2. Connecting to a nexthop router, and look up the route.
3. Resolve a MAC address from ARP table.
4. Look up an outgoing port for a MAC address.
LLDP
5. Look up an outgoing port for a MAC address.
router
switch

Fig. 1 Overview of the on-demand suspicious host isolating system.

three subsystems; SDN controller, host locating and host isolating subsystems. This section then introduces assumptions of the proposed system. This section then defines each requirement for each subsystem. This section finally presents each behavior of each subsystem.

4.1 Overview

Figure 1 depicts an overview of the on-demand suspicious host isolating system. Since OpenFlow switches are not so common, all switches and core routers are traditional, and accessed by telnet or SSH. In Fig. 1, all switches and routers basically implement LLDP [4] or CDP. These switches and routers can then find a neighboring switch using LLDP or CDP. On the other hand, few switches or routers may not implement both of LLDP and CDP, and the number of such switches or routers are very small. For these switches or routers, a neighboring switch or router can be statically defined at an SDN controller of the proposed system. When an IP address of a suspicious host is given, an SDN controller connects to a pre-defined core router, and locates a port of a switch to which the suspicious host is connected as shown in Fig. 1. The SDN controller then requests the switch to shut down a port or filter out the MAC address of the suspicious host.

4.2 Assumptions

4.2.1 Event

This paper focuses on an event of a suspicious communication that an external organization alerts. Other events are out of scope of this paper, and future works. This paper then assumes that an external organization gives following information of an event:

(1) an IP address of a suspicious host,
(2) an IP address of a corresponding host, and
(3) a time when a suspicious communication is observed.

4.2.2 Network Configuration

This paper assumes a network such that:

(1) a target network is a campus network of a university,
(2) the number of hosts are more than 6,000,
(3) the number of routers and switches are more than 300,
(4) almost all routers or switches support LLDP or CDP,
(5) one or more switches are installed between a router and edge network equipment of a user or department,
(6) a network is divided into public space segments where hosts may frequently move around and private space segments where hosts rarely move,
(7) an IP address is assigned by DHCP in the public space segments, and the DHCP lease time must be longer than ARP expiry time and must be longer than a supported alert delay, say one hour or longer, in order to properly locate a suspicious host, and
(8) a Virtual Routing Forwarding (VRF) or a routing domain is given in advance, and no IP address is duplicately assigned to different hosts in an organization network.

4.2.3 Command Line Interface of Network Equipment

This paper assumes that a Command Line Interface (CLI) of network equipment:

(1) has the same command and output when a maker, product, and firmware version is the same, and
(2) has the similar command and output when a maker and product is the same and the firmware version is different.

The firmware that has drastic changes for the same maker and product is out of scope of this paper because it needs more efforts to support a different product and firmware.

4.3 Requirements

4.3.1 Requirements for SDN Controller

The SDN controller is in charge of providing interfaces to control all switches and routers with the host locating system and host isolating system. The SDN controller must:

(1) be given no information about a maker of a switch to be connected,
(2) support different switches and routers of different makers,
(3) support different account information, i.e., different user names and passwords, for switches or routers, and
(4) support multiple access methods to a switch or router such as telnet and SSH.

4.3.2 Requirements for Host Locating System

The host locating system is in charge of *locating* a suspicious host. When an event is alerted and considered as an incident, the host locating system must:

(1) require no IP address allocation/assignment database, no host location database and no network topology map in advance,
(2) require an IP address of one of routers in an organization network,
(3) require an IP address of the suspicious host,
(4) require a Route Distinguisher (RD) or name of VRF if and only if necessary, and
(5) produce *location information* of the suspicious host for the host isolating system.

4.3.3 Requirements for Host Isolating System

The host isolating system is in charge of immediately isolating a suspicious host from a network in an organization. The host isolating system must:

(1) minimize the number of isolated hosts when the suspicious host is isolated,
(2) require *location information* of the suspicious host that the host locating system produces,
(3) isolate the suspicious host from a network,
(4) notify an operator of how to revert the isolation, and
(5) support a dry run in which the suspicious host is not actually isolated.

4.4 SDN Controller

In this paper, the SDN controller just provides interfaces to control a switch and router with the host locating and isolating systems. The SDN controller has functions described in the following sections.

4.4.1 Static Neighbor

As described before, there are few switches or routers that implement neither of LLDP nor CDP. In addition, some switches such as AlaxalA AX3800 series in actual environment do not support LLDP if they are in a *stack* configuration. To make matters worse, AlaxalA AX2530 and AX8600 series implement different LLDP versions and do not work well by default. In order to support these cases, the SDN controller supports to statically define a neighboring switch or router and its port like below.

```
'core-router_TenGigabitEthernet 1/2' =>
    { name: 'ESXi-switch',
      ia:   '192.168.0.1' }
```

This static neighbor definition represents that “ESXi-switch” whose IP address is 192.168.0.1 is connected to “TenGigabitEthernet 1/2” of “core router.”

4.4.2 Multiple Transports and Accounts

The SDN controller currently supports telnet and SSH. The SDN controller assumes that it is unknown in advance if a switch or router accepts telnet or SSH or both. The SDN controller then supports to try both of telnet and SSH. This nature requires no network topology map and no switch port list, and keeps administrators away from maintaining these map and list.

The SDN controller never requires a tuple of a switch or router, login password and privilege access password, so called *enable password*. The SDN controller then automatically tries these passwords one by one. This nature also enables administrators to be free from maintaining a switch and router password list.

4.4.3 Maker and Product Detection

The SDN controller supports different makers and different products as described later. To this end, the SDN controller can detect a maker and product when connecting to a switch or router. A maker of a switch or router can be detected by:

- CLI prompt format,
- an error message of a CLI command, and
- a characteristic use of an escape sequence.

Regarding an example of Section 4.4.3, an AlaxalA edge switch requests a cursor position using CSI escape sequence, DSR (0x1b[6n), right after SSH authentication succeeds. The AlaxalA edge switch waits for a reply for the request before the switch sends a CLI prompt. The SDN controller must then reply to the request, and can detect the maker.

Once a maker is detected, the SDN controller can easily detect a product by using CLI command of the maker.

4.4.4 Different Makers and Products Support

The SDN controller supports different makers and products since it is common to install switches and routers of different makers and products in actual environment. Even different products of the same maker may have a different CLI command for the same purpose. The SDN controller then supports a different CLI command per product, and provides a common notation to the host locating system and isolating system as below.

```
host-locate <IP address>
host-isolate <switch IP address> <port number>
```

4.5 Host Locating System

The host locating system dynamically locates a suspicious host, i.e., the host locating system locates which port on which switch the suspicious host is connected to. The host locating system requires only an IP address of the suspicious host, an IP address of a router and RD or name of VRF if necessary, and does not require a pre-defined host database. This nature reduces a load on an operator in an organization to build or periodically update a host database. This nature can then locate even a host that is not registered to such host database. The host locating system is given an IP address of one of routers and VRF in an organization network, and then locates a suspicious host as follows.

(1) connect to a router, which is given in advance,
(2) look up a route for an IP address of the suspicious host and VRF,
(3) connect to the nexthop router of the route if the route is not *directly connected*,
(4) repeat (2) and (3) until a *directly connected* route is found, i.e., locate a router that has a *directly connected* route for an IP address of the suspicious host and VRF,
(5) identify a VLAN for the IP address at the router,
(6) locate a *directly connected* router for the IP address on the VRF,
(7) resolve a MAC address of the suspicious host from an Address Resolution Protocol (ARP) [7] table,
(8) identify a port on which the MAC address is seen in a MAC address forwarding table,
(9) discover a neighboring switch on the port,
(10)repeat from (8) to (9) until a neighboring switch is not found,
(11)finally locate a port on an edge switch accommodating the MAC address, and
(12)produce *location information* of the suspicious host.

One can see more detailed pseudo code in **Fig. 2**. As shown in Fig. 2, note that there is a special case where a departmental router is installed and routes are directed to the departmental router, i.e., an organization-wide administrator cannot operate the departmental router, and a MAC address of the actual suspicious host cannot be resolved. In this case, a MAC address of the departmental router should be resolved and the departmental router should be isolated. This allows an organization to flexibly design an organization network.

4.6 Host Isolating System

The host isolating system enables to immediately isolate a suspicious host from a network in an organization. There may be multiple methods to isolate a suspicious host as discussed later. This paper here proposes two methods as follows.

- Shutting down a port on an edge switch: This method is intuitively easy to understand for a human operator, and feasible to implement on almost all products of a switch. This method can then confine a suspicious host. This method, however, may collaterally isolate another unsuspicious host

```
def locate_host(core_router, gip, time)
    # resolve an internal IP address and VRF.
    (ip, vrf) = resolve_local_ip_address(gip, time)

    # find a directly connected router.
    router = core_router
    while router do
        route = router.lookup_route(ip, vrf)
        if route.is_directly_connected?
            break
        end
        # we cannot control user's or
        # departmental router.
        if not route.nexthop.is_ours?
            ip = route.nexthop.ip_address
            break
        end
        router = route.nexthop
    end

    vlan = route.vlan
    mac = router.resolve_mac_address(ip, vlan)

    # locate an edge switch and port.
    sw = router
    while sw do
        port = sw.mac_address_table(vlan, mac)
        neighbor = port.get_neighbor
        if neighbor.nil?
            break
        end
        sw = neighbor
    done
    return sw, port
done
```

Fig. 2 A pseudo code to locate a suspicious host.

that is accommodated to the same port on the same switch. This method cannot follow a mobile suspicious host that moves around a network. This method is then adopted to a suspicious host on a private space segment where a host rarely moves.

- Filtering out a MAC address of a suspicious host at a router: This method can follow a mobile suspicious host that moves around a network. This method is then adopted to a host on a public space segment such as a lecture room and wireless network where a host frequently moves.

The host isolating system then operates as follows:

(1) connect to a router or switch that the host locating system gives,

(2) shut down a port or filter out a MAC address,

(3) send an e-mail of a result of shutting down or filtering out to all operators given in advance.

5. Implementation

This section presents our first prototype implementation of the SDN controller, the host locating system and the host isolating system. We have implemented the SDN controller, the host locating and isolating system as scripts written in Ruby, and they are combined into one script.

5.1 SDN controller

The SDN controller currently supports CLIs of Cisco Catalyst 6500, 3560, 2960, AlaxalA AX8600, AX3800, AX2530, AX2200, AX620, NEC IX, Paloalto PA-5220, PA-3020, PA-850 and Aruba WLC. In many cases, a pager of a CLI bothers the SDN controller, and the SDN controller then disables a pager right after being logged in. *Out-of-band messages* such as logging messages also bother the SDN controller. For example, when a link on a port gets up, the logging message will show up in a CLI output. This kind of message is unexpected, and bothers the SDN controller. The SDN controller then disables logging messages to a CLI except for AlaxalA AX2200 because AX2200 cannot disable the logging messages in a CLI console.

5.2 Host Locating System

The host locating system currently supports the case where all IP addresses are unique in a network even the host locating system itself supports multiple routing tables using VRF. The host locating system then supports all types of routes of routing protocols implemented on a core router. The host locating system then chooses the longest matching prefix to locate a host. The host locating system also supports to statically define a preferred nexthop as a tie breaker.

5.3 Host Isolating System

Our implementation currently supports only two types of isolating methods: port shutdown at an edge switch and a MAC address filtering at a router. Regarding a MAC address filtering, there is an implementation limitation that requires all VLANs are known and given in advance. This is because almost all switch or router can filter a MAC address per VLAN. It may be technically possible to filter out a MAC address on all VLANs. This means that the proposed system of the host isolating system of a MAC address filtering requires that all VLANs are known and given in advance.

6. Evaluations

This section evaluates the proposed system, and then presents how the host locating and isolating system avoid an error and reduce a delay by a human operation.

6.1 Human Operations versus Automated Operations

This section presents that a human operation is slower and more dangerous than an automated operation to locate a host in authors' actual campus network.

In our campus network, there were 315 switches and routers. These were installed in three main campuses: Koyama, Yonago and Hamasaka, and three branch campuses: Hiruzen, the kindergarten and the special support education school. There was only one router in each campus, and all routers were directly connected by VLAN services and so on. Each router in each campus accommodated switches in each campus in star topology, and there might be several intermediate switches between a campus core router and an edge switch. The proposed system firstly connected to the core router in Koyama campus, connected to the other router in the other campus if necessary, and then connected

Table 1 Times required to locate hosts.

			traditional manual operations[*1] (min.:sec.)								the proposed system (sec.)		
IP address	routers	switches	A	B	C	D	E	mean	median	SD	mean	median	SD
IP_1	2	4	4:20	3:30	4:00	4:00	14:50	6:08	4:00	4:52	4.311	4.324	0.112
IP_2	1	5	2:30	5:00	8:00	3:00	12:00	5:30	4:00	3:44	8.247	8.312	0.146
IP_3	3	1	2:30	3:30[*2][*3]	4:00	0:40	15:10	5:10	3:30	5:44	2.121	2.155	0.095
IP_4	2	2	2:20	4:00	2:50	2:30	8:30	4:02	2:50	2:35	4.420	4.381	0.147
IP_5	3	1	3:30	2:00[*2]	2:20	2:00	3:50[*5]	2:44	2:20	0:52	2.373	2.392	0.090
IP_6	1	2	8:50[*4]	5:00	10:00[*4]	3:30[*4]	18:50[*5]	8:50	8:50	5:12	5.329	5.352	0.160
IP_7	1	3	2:00	4:30	2:00	3:30	11:10	4:38	3:30	3:48	7.047	6.990	0.219
IP_8	2	3	1:40	3:30	2:00	5:00	7:00	3:50	3:00	2:13	3.626	3.645	0.124
IP_9	1	3	13:20	3:30	5:50	3:00	11:10	7:22	5:50	4:39	7.030	7.083	0.252
IP_{10}	2	3	2:50	3:00	2:20	8:00	8:20	4:54	3:00	3:00	3.583	3.591	0.097
IP_{11}	1	3	2:50	2:30	1:50	3:00	16:00	5:14	2:50	6:02	6.915	6.905	0.232
IP_{12}	2	4	2:20	2:00	1:40	3:30	15:40	5:02	2:20	5:59	4.162	4.205	0.132
		mean	4:05	3:30	3:34	3:33	11:42	5:17	-	-	4.930	-	-
		median	2:40	3:30	2:35	3:30	11:35	-	3:30	-	-	4.381	-
		SD	3:20	1:10	2:23	1:47	4:08	-	-	4:13	-	-	1.911

to a neighboring switch one by one. All routers and switches were made by Cisco Systems, Inc. except for several switches, and these Cisco switches could be accessible using telnet. Five different passwords are configured for a login and privilege access to a switch and router in order to separate an access privilege for each kind of switch or router:

- 1 switch for a special network dedicated for a TV conference application,
- 4 switches for VMWare ESXi,
- core routers in Koyama and Yonago,
- 6 departmental backbone switches in Koyama, and
- other edge routers or switches.

We currently have 9 technical staffs who are in charge of operating our campus network. All technical staffs have been working for our organization for more than 5 years, and memorize a certain degree of a network topology. Technical staffs then measured the time required to locate each host by himself or herself under below conditions:

(1) 12 IP addresses of hosts are given,
(2) each IP address is on a different network segment, i.e., a different broadcast domain,
(3) some IP addresses are from a different campus,
(4) each technical staff may measure whenever he or she is available from June 15th 2017 to June 19th 2017,
(5) each technical staff may obtain information associated with a given IP address from a database such as a department, geographical location, building and so on,
(6) each technical staff should not practice in advance,
(7) each technical staff should locate a switch and port to which a host given an IP address is connected, and
(8) each technical staff should measure the times only by himself or herself without any advice in advance.

Table 1 then shows the results of the measurements. In Table 1, *Routers* is the number of core routers that the proposed system traversed including the first Koyama campus core router. *Switches* is the number of switches from the core router in the campus to the edge switches directly accommodating a host of a given IP address.

As shown in Table 1, there were some human operation errors that wrongly located a switch or port. Some operators then required advice to locate a switch, or could not locate a switch or port. It can, therefore, be said that it depends on an operator's skill to locate a host.

Table 1 also shows that a time to locate a host depends on an operator and switch as each standard deviation (SD) of required times is larger. This indicates that an operator may know a switch well but not other switches. Again, it can, therefore, be said that it depends on an operator's skill to locate a host.

In case of the proposed system, each SD for each IP address to be located is small, and each time depends on the number of switches and routers. We can say that the proposed system can exclude dependencies on a human operation.

Table 1 also shows that a human operator requires 3 minutes on average while the proposed system requires 10 seconds at maximum. We can also say that the proposed system can reduce operation delays.

Regarding the proposed automated system, a required time is not correlated to the number of routers but the number of switches. This indicates that the proposed method to find a neighboring switch needs more time than one to find a directly connected neighboring router.

6.2 Different Accounts, Protocols and Login Delays

As described in the previous section, a required time to locate a host is not correlated to the number of routers but the number of switches. In order to identify the cause, this section presents how long it takes to log in a switch or router when multiple accounts are given for telnet and SSH. A target switch was, Cisco Catalyst 2960C Software (C2960c405-UNIVERSALK9-M), Version 15.2(2)E4, RELEASE SOFTWARE (fc2). Regarding SSH, SSH protocol version was 2, and its RSA key was 2,048 bits long. The number of passwords was changed from 1 to 4 as shown as No. in **Table 2**. All logins were tried 10 times, and their mean, median times and SD were computed as shown in Table 2.

Interestingly, telnet is faster than SSH, and the required times for telnet login are not correlated to the number of passwords if the number is lower than 4. This indicates that telnet quickly ac-

[*1] Less than 10 seconds rounds down to zero second.
[*2] A MAC address was not resolved before an advice was given.
[*3] A wrong port was located.
[*4] A wrong switch was located.
[*5] A switch could not be located.

cepts a next password right after wrong password is input. The required time for telnet login dramatically bumps up if the number of passwords are 4 because Cisco closes a telnet connection right after authentication fails 3 times. In this case, a new telnet connection should be then opened.

On the other hand, the required times for SSH login are relatively slow. The required times are then correlated to the number of passwords. This is because our implementation drops a SSH connection for each password.

As a result, we could say that telnet may be better for a switch or router login if the network is enough to be secure.

In addition, a required time is, however, not simply correlated to the number of switches. This might result from a fact that there were firmware version or CPU difference of a router or switch.

6.3 False Positive Alert Prevention

This paper focuses on an event of a suspicious communication that an external organization alerts. The alert may be sometimes incorrect. It would be better to avoid unnecessary host isolation as a host isolation may have a serious impact on one's regular work. To this end, we have implemented a very simple safeguard hook. The simple safeguard is not to shut down a port whose link is a 10GbE because a 10GbE link is adopted to only one of network interfaces of a backbone switches, routers and servers that run VMware ESXi accommodating more than 200 Virtual Machines (VMs) in authors' environment.

Even the simple safeguard actually worked well for authors. On January 1st 2018, a commercial SOC alerted us that a suspicious server in our network was maliciously attacking other hosts in our network. We were firstly suspecting that the suspicious server was compromised and then attacking other hosts. The proposed system was then trying to shut down a port to the suspicious host. The port was, however, a 10GbE link, and then the safeguard worked to prevent the proposed system from shutting down the port. The suspicious host was actually checking vulnerabilities of other hosts, and these behaviors were intended. The alert was, therefore, false positive. The suspicious host was a VM and resident on VMWare ESXi. If the link had been shut down, more than half of all VMs would have been isolated from our network, and our regular works might have almost stopped.

Table 2 Times required to login a switch.

	telnet			SSH		
No.	mean	median	SD	mean	median	SD
1	0.757	0.751	0.0234	5.628	5.608	0.0365
2	0.779	0.747	0.0629	14.952	14.922	0.0688
3	0.759	0.752	0.0209	24.210	24.208	0.0241
4	3.271	3.259	0.0300	33.494	33.485	0.0340

7. Discussions

This section discusses issues to locate and isolate a suspicious host that can be seen especially in authors' environment.

7.1 Reactive versus Proactive

This paper proposes an on-demand suspicious host isolation adopting SDN-like approach, which is, so to speak, *reactive* because it requires no host database or network topology map in advance. This *reactive* approach can reduce a control traffic, CPU load on a switch and storage consumption. This may, however, be unable to handle an incident that occurs outside of business hours and a suspicious host move. Even within business hours, it may be difficult for a small organization like us to handle the case that a suspicious host moves because no staff is available. In our experience, we would say to require more than an hour to handle an incident. A *proactive* approach may be then necessary, and should be more lightweight than a traditional method using SNMP. We are now considering to adopt an SDN-like approach to this issue again, and this is our future work.

7.2 Host Isolating Methods

There are multiple methods to isolate a suspicious host from a network. We found that methods can be classified from the viewpoint of a type, method, place, supporting a mobile host, containment, collaterally isolating an unsuspicious host and feasibility as shown in **Table 3**. As shown in Table 3, each method has pros and cons. We then found that there is no one best method that can be adopted to all cases. For example, an authentication seems to be the best method. It may, however, be difficult to employ an authentication on all ports on all switches because some hosts still do not implement an authentication. A MAC address filtering at an edge switch then seems to be better method. A MAC address filtering cannot, however, be implemented at an edge switch in some cases because some switches cannot simultaneously implement a MAC address and IP or UDP/TCP filtering. These switches, hence, cannot simultaneously implement

Table 3 Pros and cons of host isolating methods.

Type	Method	Place	Mobile	Containment	Collateral	Feasibility
authentication	authentication	auth. server	good	good	good	poor
physical operation	plug off a cable	edge	poor	good	fair	good
shut down	port	edge	poor	good	fair	good
	VLAN (L2)	edge	poor	good	fair	fair
		router	fair	fair	poor	good
	VLAN (L3)	router	fair	good	fair	good
filter	MAC address	edge (port)	poor	good	good	poor
		edge (FIB)	poor	good	good	fair
		router	good	fair	good	good
	IP address	edge (port)	poor	good	good	fair
		router	fair	fair	good	good
		exit firewall	fair	poor	good	good
	UDP/TCP port	edge (port)	poor	fair	good	fair
		router	fair	fair	good	good
		exit firewall	poor	poor	good	good

a MAC address filtering and Web authentication because a Web authentication generally requires UDP/TCP filtering to allow a DHCP communication before an authentication. Web authentication may be required in many cases today, and a MAC address filtering at an edge switch may, therefore, not be feasible.

8. Related Work

Nagai et al. investigated and reported differences between ISMSs in national universities in Japan [11]. They also presented their own incident management system using trac [12]. They then reported that their system could record information of only about a half of all security events because some of those events were reported or discussed in meetings and their data was never input to the system.

Hasegawa et al. proposes the countermeasure support system against incidents caused by targeted attacks [13]. Their system automatically suggests 9 types of traffic filtering to an operator in accordance with a severity of an incident. They, however, consider only a traffic filtering across VLANs at a core router, and do not consider a traffic filtering within a VLAN. Their system then cannot avoid a sort of a malware, e.g., *WannaCry*, to spread within the same VLAN. Their system also assumes that a network configuration is given in advance. In addition, they do not consider a mobile host that moves around in an organization. These are different from our proposal.

AlaxalA Networks Corporation has released AX-Security-Controller (AX-SC) [9] on 2017 that can also isolate a suspicious host. AX-SC, however, employs a traditional method using SNMP to locate a host and periodically poll a MAC address table from a switch. AX-SC, therefore, produces more control traffic and requires more loads on a switch than our proposal. AX-SC also requires more storage or memory space for a database to locate a host than our proposal. For example, AX-SC requires 8 GB memory at least while our proposal in our evaluation environment can run just with 2 GB memory. AX-SC is now trying to support LLDP to dynamically find neighboring switches. There are, however, many limitations. For example, stack configurations of switches are not supported. Some LLDP versions implemented on AlaxalA switches are not supported. CDP is not supported while our proposal can support. AX-SC cannot find switches that are separated by a router, i.e., AX-SC should be installed in each campus network. AX-SC also cannot support a suspicious host that moves frequently, and should disconnect a suspicious host for each switch. In addition, AX-SC cannot support network equipment produced by other than AlaxalA Networks Corporation while our proposal can also support multiple makers. AX-SC cannot support host isolations for some models such AX8600 and AX3630 that are even produced by AlaxalA Networks Corporation while our proposal supports them. AX-SC cannot then handle the case where there is a router operated only by a user or a department between a suspicious host and a switch as described in Section 4.2.

Tokyo University of Agriculture and Technology has also implemented automated suspicious host isolation [14], [15]. They, however, adopted, AX-SC, and there are many limitations and drawbacks as described above.

There are also many security or network vendors such as Kaspersky, F-Secure, Symantec, TrendMicro, Paloalto, FireEye, Fortigate and Cisco that provide systems to isolate a suspicious host. Their systems, however, assume that all network equipments are produced by the same maker, and seem to employ a traditional method using SNMP.

9. Concluding Remarks

This paper has proposed the on-demand suspicious host isolating system using SDN-like approach. Our proposal requires no host database, network topology map and switch port list in advance, and isolates a suspicious host on-demand. Our system has appeared to be able to locate and isolate a suspicious host within 10 seconds right after an IP address of a suspicious host is given. Right after an external organization alerts an event and we recognize, we have been able to isolate a suspicious host within 10 minutes. We are now considering to identify a responsible person, e.g., a user, of the suspicious host when locating the host. We are also considering to develop an efficient host location recording system for a past incident, and it is future work.

References

[1] Tor Project: Tor Project: Anonymity Online (2002), available from ⟨https://www.torproject.org/⟩ (accessed 2017-06-03).

[2] LAC Co., Ltd.: Japan Security Operation Center (JSOC®) — Services and Products — LAC Co., Ltd. (1995), available from ⟨https://www.lac.co.jp/english/service/operation/jsoc.html⟩ (accessed 2017-05-26).

[3] National Institute of Informatics: National Institute of Informatics (2007), available from ⟨http://www.nii.ac.jp/⟩ (accessed 2017-05-26).

[4] IEEE Std. 802.1ab-2004: *Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks: Station and Media Access Control Connectivity Discovery* (2004).

[5] Open Networking Foundation: Home - Open Networking Foundation (2017), available from ⟨https://www.opennetworking.org/⟩ (accessed 2017-05-26).

[6] Case, J., Fedor, M., Schoffstall, M. and Davin, J.: Simple Network Management Protocol (SNMP), RFC 1157 (Historic) (1990).

[7] Plummer, D.: Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware, RFC 826 (Standard) (1982). Updated by RFCs 5227, 5494.

[8] Microsoft: [MS-SMB]: Server Message Block (SMB) Protocol (2017), available from ⟨https://msdn.microsoft.com/en-us/library/cc246231.aspx⟩ (accessed 2017-06-25).

[9] ALAXALA Networks Corporation: AX-Security-Controller (2017), available from ⟨http://www.alaxala.com/jp/news/press/2017/20170601.html⟩ (accessed 2017-06-03).

[10] OHMORI, M.: On a SNMP DoS Attack against Vulnerable Architecture of Network Equipment, *SIG Technical Reports*, Vol.2016-IOT-33, No.4, pp.1–4 (2016).

[11] Nagai, Y., Tadamura, K. and Ogawara, K.: Considering Incident Management Systems in Some National Universities, *SIG Technical Reports*, Vol.2014-IS-127, No.7, pp.1–7 (2014).

[12] Software, E.: The Trac Project (2003), available from ⟨https://trac.edgewall.org/⟩ (accessed 2017-05-19).

[13] Hasegawa, H., Yamaguchi, Y., Shimada, H. and Takakura, H.: A Countermeasure Support System against Incidents caused by Targeted Attacks, *Journal of Information Processing*, Vol.57, No.3, pp.836–848 (2016).

[14] Tsujisawa, T., Sakurada, T., Segawa, H., Kawamura, Y., Mishima, K. and Hagiwara, Y.: A challenge for full deployment both 802.1x authentication and an automatically isolation function in a campus network, Task and correspondence in the operation, *Journal for Academic Computing and Networking*, Vol.22, No.1, pp.36–43 (2018).

[15] ALAXALA Networks Corporation: Case Study: Tokyo University of Agriculture and Technology (2017), available from ⟨https://www.alaxala.com/jp/introduce/case36/⟩ (accessed 2018-10-01).

Motoyuki Ohmori was born in 1976. He received his B.S. and M.S. degrees in Computer Science and Communication Engineering from Kyushu University in 1999 and 2001, respectively. He joined Information Processing Society of Japan in 2001. He had been a lecturer at Chikushi Jogakuen University since 2004. He has been an associate professor at Tottori University since 2013. His research interest includes network architecture, multicasting, routing, mobile networking and energy efficient network operation. He is a member of IPSJ, IEICE, JSSST, IEEE CS/ComSoc and ACM.

Masayuki Higashino was born in 1983. He received his B.Eng., M.Eng. and D.Eng. degrees from Tottori University, Japan, in 2008, 2010 and 2013, respectively. He has been an assistant professor at Tottori University since 2015. His research interest includes mobile agent systems, distributed systems and network architecture. He is a member of IPSJ, JSSST, IEEE CS/ComSoc and ACM.

Toshiya Kawato was born in 1989. He graduated from National Institute of Technology, Yonago College and received his B.S. degree from National Institution for Academic Degrees and University Evaluation in 2012. He had been a technical staff at Tottori University since 2012. He has been an assistant professor at National Institute of Technology, Yonago College since 2018. He has been a graduate student of Tottori University for his doctoral degree. His research interest includes distributed systems. He is a student member of IPSJ.

Satoshi Fujio was born in 1984. He received his B.S. degree in Department of Information and Knowledge Engineering from Tottori University in 2007. He has been a technical staff at Tottori University since 2007.

Kiyoyuki Nakashima was born in 1973. He received his diploma in Online information department from Hiroshima Electronics College in 1995. He has been a technical staff at Tottori University since 2009.