Electronic Preprint for Journal of Information Processing Vol.27

Regular Paper

On Automation and Orchestration of an Initial Computer Security Incident Response by Introducing Centralized Incident Tracking System

Motoyuki Ohmori[1,a)]

Received: December 10, 2018, Accepted: June 11, 2019

Abstract: A critical computer security incident may cause great damage to an organization for example by a confidential data breach or malware pandemic. In order to avoid or mitigate such damage, a quick and accurate response against a computer security incident is becoming more important. In order to realize these quickness and accuracy, this paper presents the Incident Tracking System (ITS) that orchestrates several information systems and automates an initial incident response. The ITS automatically locates and isolates a suspicious host, and sends a mail notification to the person in charge of handling an incident. The ITS can also identify or suggest a user of the suspicious host by network authentication logs or other service logs.

Keywords: computer security, incident response, network operation, automation and orchestration

1. Introduction

Computer security has been becoming more important because a computer security incident may cause great damage to an organization. Since it is difficult to prevent all incidents from happening, a proper and quick response to an incident is important in order to mitigate or minimize damage. To this end, it is now becoming common that an organization creates a Computer Security Incident Response Team (CSIRT).

In many cases, a malicious communication is detected by an external organization such as the Japan Security Operation Center (JSOC) [1] operated by LAC Co., Ltd., or the National Institute of Informatics Security Operation Collaboration Services, the so-called NII-SOCS, operated by National Institute of Informatics (NII) [2], government organizations or others. A CSIRT in an organization then firstly recognizes a computer security event after receiving an alert of a suspicious communication from an external organization. The CSIRT 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 CSIRT then initiates an incident response.

In order to mitigate or avoid damage to an organization caused by an incident, a quick and proper initial response to an incident is important. A quick initial response can reduce the possibility of data breach itself, also may reduce an operation to investigate the data breach. An effective initial response may be able to avoid misoperation and therefore retain this availability. It may, however, be difficult to make an initial response quicker and more proper.

To this end, we propose to automate and orchestrate an initial incident response using the centralized Incident Tracking System (ITS). An initial incident response with this system indicates to isolate a suspicious host from a network. All processes of an initial incident response are basically automated, and are recorded on ITS as an issue or ticket. ITS also enables persons involved in an incident to share necessary information in order to make an initial incident response more effective. ITS then provides *workflow* that navigates a person in charge to intuitively operate.

Contributions of this paper can be summarized as follows:

- automated and orchestrated initial incident response can dramatically reduce the time required to isolate a host and send an alert mail,
- automated host isolation can avoid misoperation caused by a false-positive report from a Security Operation Center (SOC),
- status of a ticket of an incident on ITS can be combined with *handling*, *uncritical*, *ball*, i.e., who is in charge of, and *done*,
- this combined status can navigate CSIRT members to easily and intuitively change Finite State Machine (FSM) of an incident on ITS,
- *workflow* works well for ITS, and
- most of many fields are unnecessary to input in many security events because most of security events are not critical security incident and they should be hidden if unnecessary.

The rest of this paper is organized as follows. Section 2 presents automation and orchestration of an initial incident response centralizing ITS. Section 3 presents how faster automated and orchestrated incident response is in comparison with a manual incident response. Section 4 discusses operational issues regarding an incident handling. Section 5 refers to related work. Section 6 finally concludes this paper.

[1] Tottori University, Tottori 680–8550, Japan
[a)] ohmori@tottori-u.ac.jp

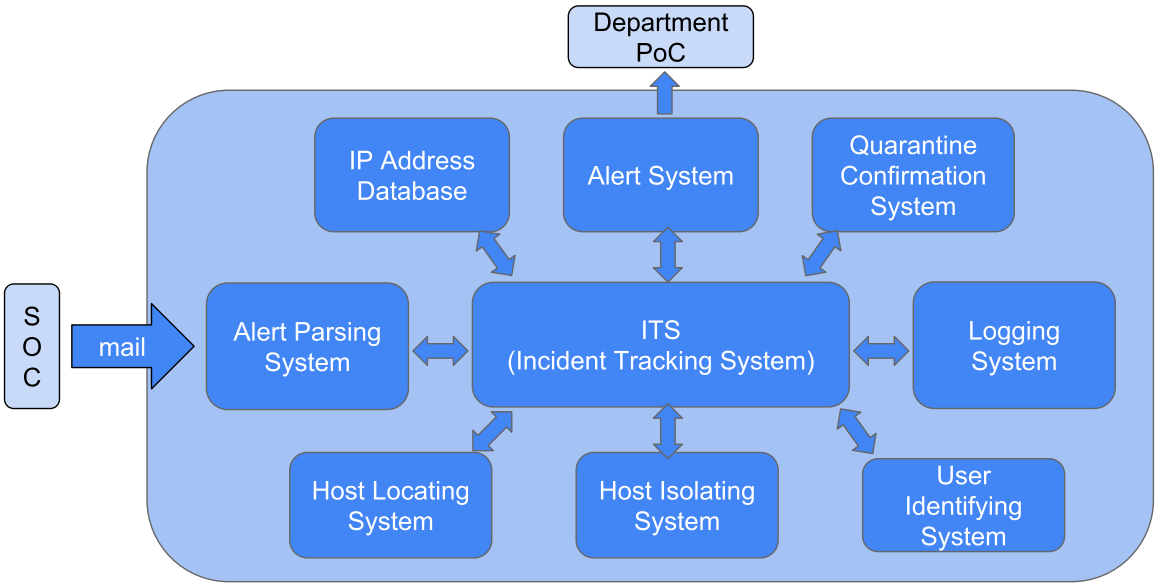


Fig. 1 Overview of components of automation and orchestration of an initial computer security incident response.

2. Automation and Orchestration of an Initial Computer Security Incident Response

This section presents the automated and orchestrated initial incident response system. This section firstly overviews the components of automation and orchestration of an initial incident response. This section then presents each component in detail.

2.1 Overview

Figure 1 depicts components of an automated and orchestrated initial incident response system. As shown in Fig. 1, we assume that an external SOC sends a mail in fixed format indicating an incident. Other notifications from a SOC such as a telephone or a mail written in free format are out-of-scope of this paper. The system is composed of 9 components, and they are described in the following sections.

An initial incident response is automatically done, and then CSIRT members manually investigate an incident by utilizing ITS as follows:

(1) logging system always stores required logs,
(2) a SOC find a security incident and send an alert mail,
(3) alert parsing system receives and parses the alert mail, creates a ticket of the incident on ITS,
(4) ITS then asks quarantine confirmation system if downloaded malware is already quarantined or not,
(5) quarantine confirmation system then investigates anti-virus software logs and returns its result to ITS,
(6) ITS then asks host locating system to locate a suspicious host,
(7) ITS then asks host isolating system to isolate the suspicious host,
(8) ITS obtains Point of Contact (PoC) information by querying IP address database,
(9) ITS obtains a possible user of the suspicious host by querying user identifying system,
(10) ITS requests alert system to send an alert mail to PoC,
(11) ITS notifies CSIRT of above incident handlings by mails and the initial incident response finishes,
(12) CSIRT members manually start to physically identify the suspicious host and plug off the LAN cable,
(13) CSIRT members then investigate the suspicious host and investigate what happened if required,
(14) CSIRT members then check if there is a possibility of information compromise or not, and
(15) CSIRT members finally summarize the incident and close the ticket on ITS.

2.2 Logging System

The logging system holds information required for an incident response. Log messages, however, tends to be a large amount. For example, a firewall log consumes about 13 GB per day when the log is stored as a text file. A log stored in a text file is, however, not useful for searching purposes because keywords for a search are not indexed. Database is appropriate for a search. Database, however, requires more storage space. It can be said that a text file is suitable for the long term while database is suitable for a short-term search.

We have then implemented two types of logging system as follows:

- mongoDB holds recent two or three month log
- file holds raw syslog messages.

2.3 Alert Parsing System

The alert parsing system polls a mail box and parses an alert mail sent from a SOC. The alert parsing system assumes that:

(1) a SOC can be identified by a source mail address in an alert mail,
(2) an alert mail includes an identifier of an alert (e.g., incident or ticket ID), and
(3) a SOC must provide required information as listed below to identify a suspicious communication in any way, e.g., in a message body of a mail or in a SOC portal site:

- a source IP address,
- a destination IP address,
- a source TCP/UDP port number,
- a destination TCP/UDP port number, and
- time of suspicious communications.

The alert parsing system then has modularity regarding a SOC, and each parser for each SOC can be easily defined as a module. A core part of the alert parsing system identifies a SOC by a source mail address, and passes contents of a mail to a parser of the SOC. Each parser of each SOC must return the required information as described above. If a SOC changes a mail format or a user interface, a parser for the SOC must also follow the changes. The change is necessary only in the parser and does not affect a parser for other SOC.

We have implemented modules for WideAngle operated by NTT Communications Corporation and NII-SOCS operated by NII. WideAngle is a commercial SOC service while NII-SOCS is a collaboration services for national universities in Japan. In the case of WideAngle, an alert mail can be in the fixed format that includes:

- a source IP address,
- a destination IP address,
- a source TCP/UDP port number,
- a destination TCP/UDP port number,
- time of suspicious communications,
- severity of a security event, and
- brief description of a security event.

the WideAngle module of the alert parsing system parses the above information for other systems. Generally speaking, in the case of a commercial SOC service, traffic is monitored in an internal network. CSIRT members can then identify a suspicious host by given internal private IP addresses and port numbers. If NAT or NAPT is employed and a SOC service can give only translated outer global IP addresses and port numbers, an associated internal private IP address and port numbers can be obtained by querying a NAT/NAPT box or logging system that stores traffic logs. CSIRT members can then identify a suspicious host even if NAT or NAPT is employed.

In case of NII-SOCS, an alert mail can also be in the fixed format but that includes only:

- an IP address of a suspicious host,
- time of suspicious communications, and
- alarm name.

the NII-SOCS module of alert parsing system parses the above information for other systems. These might be, however, insufficient because a suspicious flow cannot be identified when NAT or NAPT is employed. In order to identify a flow, the NII-SOCS module accesses a portal site of an organization of NII-SOCS. the NII-SOCS module then obtains the necessary information for alert parsing system.

The alert parsing system then creates a ticket of an incident on ITS, and registers an original alert mail and the required information.

Note that all alerts from WideAngle are always treated as an incident and not as false positive detection for now because WideAngle alerts only actual C&C server communications as critical and their alerts are accurate. On the other hand, alerts from NII-SOCS are always examined by the quarantine confirmation system and CSIRT members, and are never automatically treated as an incident because NII-SOCS alerts are rarely critical. More accurate false positive detection is for future work.

2.4 Host Locating System

The host locating system dynamically locates a suspicious host; the suspicious host is connected to which port on which switch. 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 requires a pre-defined host database. This characteristic reduces the load on an operator in an organization to build or periodically update a host database. This characteristic can then locate even a host that is not registered to such host database. The host locating system has two operational modes: *on-demand* and *proactive*.

2.4.1 On-demand Host Locating

The host locating system is given an IP address of one of the 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) [3] 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.

The on-demand host locating system is described more in detail [4].

2.4.2 Proactive Host Locating

The host locating system *proactively* stores ARP table entries in each core router. All hosts are then usually authenticated by one of IEEE802.1x, Web authentication and MAC address authentication. These authentication logs are stored in the logging system. It can be considered to be difficult to deploy network authentications to all network equipment. In this case, MAC address authentication can be configured to authenticate all MAC addresses where it is difficult to deploy network authentications. We, Tottori University, actually enable IEEE802.1x, Web authentication and MAC address authentication in all network switches in our university. The host locating system then locates a suspi-

cious host from network authentication log.

2.5 Host Isolating System

The host isolating system enables CSIRT members 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 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 host locating systems gives,
(2) stop further process if a port or a MAC address is listed in white list,
(3) shut down a port or filter out a MAC address,
(4) send an e-mail of a result of shutting down or filtering out to all operators given in advance, and
(5) register its content to ITS.

Note that the host isolating system does nothing, i.e., does not isolate a suspicious host, if an alerted malware is already quarantined on the host as described in Section 2.8.

2.6 IP Address Database

The IP address database holds information about IP address allocations:

- IP address prefix,
- network media (i.e., wired or wireless),
- campus,
- network segmentation type (i.e., research network, educational network, secretariat network and so on),
- Point of Contact (PoC),
- department or division,
- section, and
- remark.

2.7 Alert System

The alert system automatically sends an alert mail to departmental PoC in accordance with information given by the alert parsing system and IP address database. An alert mail format is in fixed format, and it can be easily modified by editing a text template file.

2.8 Quarantine Confirmation System

The quarantine confirmation system determines if an alerted malware is already quarantined on a suspicious host or not. If the malware is already quarantined, it is unnecessary to isolate the suspicious host anymore. The quarantine confirmation system can then avoid unnecessary host isolation, and mitigate reduction in availability. We have implemented the quarantine confirmation system as follows. We deliver VirusBuster Corporate Edition to our members. In VirusBuster Corporate Edition, there is a central server that collect all logs and quarantined malware. These logs can then be forwarded to another server using syslog protocol. We then have these logs in mongoDB and files as described above. In these logs, a host is identified by MAC address or host name. When NAT or NAPT is not employed in a room of our member, a host can be then identified by MAC address. We can then search for a log that indicates a reported malware is already quarantined.

2.9 User Identifying System

When NAT or NAPT is employed in a room of our member and there are multiple hosts in the room, it is difficult to identify a suspicious host. The user identifying system can then suggest who may be a user that uses a suspicious host. A suspicious host may be able to be identified by investigating the user hosts. To this end, we utilize authentication logs of the following other systems:

- Shibboleth IdP,
- dovecot, and
- groupware.

These logs are held in the logging server as described above. When a suspicious host is not identified by a network authentication, the user identifying system searches for a login record from the logging system. The user identifying system then suggests possible users at the time when suspicious communication is detected. When multiple users are found, all of them are suggested by registering users to ITS.

2.10 Incident Tracking System

ITS is in charge of sharing information among CSIRT, recording actions that CSIRT takes and observed phenomenon, and making an incident trackable. ITS must be able to:

(1) share information among CSIRT members involved in a security incident response,
(2) issue a ticket for an incident,
(3) differentiate *open* and *closed* issues.
(4) associate similar incidents with a ticket,
(5) register CSIRT member in advance,
(6) notify CSIRT involved of updates of an incident,
(7) upload a file for an incident,
(8) automatically produce a final report of an incident, and
(9) automatically produce a summary of incidents during specified duration.

ITS can then be built using an exiting Bug Tracking System (BTS) or issue tracking system [5], [6], [7]. ITS, however, needs to assign an incident to a group of CSIRT members while BTS usually assigns to one person. ITS is very different from BST or issue tracking system in this point. In this paper, we use Redmine [5] as ITS.

2.10.1 Status of a Ticket

This section presents what problems we faced regarding the status of a ticket, and how we have solved them.

We firstly faced the problem that CSIRT members did not *close a ticket* even after the incident handling was over. From the point of view of a software developer, it is extremely common to close a ticket after a bug or problem is solved. Most of CSIRT members, unfortunately, were not experienced in developing an information system from scratch in a real environment or in commercial use. They were, hence, not accustomed to close a ticket. They could not then close a ticket although our incident handling manual said to close a ticket after the incident handling finished.

We secondly faced the problem that it was unclear who was a person in charge and who should have been currently responsible to take an action. For example, let us assume that an external organization notifies us of a suspicious communication. In this case, we need to compute a private IP address of a suspicious host from the notified global IP addresses and port numbers because we adopt NAT or NAPT for all hosts in our campus network. In our organization, CSIRT is responsible for computing a private IP address from the global IP addresses and port numbers. This computation can be done as follows:

(1) identifying a flow by the global IP addresses and port numbers from traffic log stored in NAT/NAPT box or logging system, and
(2) obtaining a private IP address from traffic log of the matching flow.

It was, however, difficult for CSIRT to notice at a glance whether this computation was required or not. We had then to introduce a new input field, *ball*, that indicated who, i.e., CSIRT, a department or a user, was in charge of an incident. This field was, however, not always updated because the person in charge could not notice that he or she should have updated the field. Even if the field was properly updated, almost all CSIRT members did not check to see a *ball* field, and did not join an incident handling.

We thirdly faced the problem that CSIRT member could not understand when they could close a ticket. Redmine unfortunately cannot define a detailed condition onto each field by default when a ticket can be closed. Even if such a detailed condition can be defined, it would be complicated and difficult for CSIRT members to understand which field should have what value.

We have then solved these problems using *workflow* in Redmine. In order to adopt *workflow*, we firstly have modified and defined the *status* of Redmine as below.

$$\begin{aligned} \mathit{status} &::= \mathit{type}\ \text{"("}\ \mathit{ball}\ \text{")"} \\ \mathit{type} &::= \mathit{handling} \mid \mathit{uncritical} \\ \mathit{ball} &::= \text{"CSIRT"} \mid \text{"department"} \mid \text{"user"} \mid \text{"done"} \end{aligned}$$

As can be seen in the above definition, we have combined status with *handling*, *uncritical*, *ball* and *done*, i.e., finished status. We have actually defined the status of a ticket as shown in **Table 1** in our Redmine. We have then instructed CSIRT members to go toward *done* state.

Table 1 Status of a ticket.

Status
identification (CSIRT)
awaiting identification (department)
data breach investigation (CSIRT)
data breach investigation finished (CSIRT)
awaiting final report (department)
awaiting OS re-installation (student)
false positive (done)
uncritical (done)
confirmation operation (done)
the same host as other incident (done)
out of scope of CSIRT (done)
finished (done)

2.10.2 FSM for ITS

Using *combined status* as defined in Section 2.10.1, we define FSM of our ITS as shown in **Fig. 2**. In Fig. 2, each box and arrow represent the status and an event, respectively. Blue boxes represent *open* status. On the other hand, green boxes represent *closed* status. All green status except for *finished (done)* can be moved from all status. As shown in Fig. 2, all events change status toward the *closed* status, and there is no event that goes back towards initial status. In addition, all blue boxes have two or less arrows, that is, there are only two choices at maximum when the status is changed except for *closed* status. As can be also seen in Fig. 2, a lesser critical incident requires lesser status changes. While a really critical incident rarely happens, false positive detection often occurs in our environment. This characteristic decreases operations that CSIRT member must do on an incident handling. We have then implemented this FSM in Redmine using *workflow*.

2.10.3 Input Fields of a Ticket

When an incident is handled, there are many things to interview, clarify and record. We define then information that ITS should hold as shown in **Table 2**. Note that boolean is not used in order to allow empty even though Redmine has a value type of boolean. Boolean values are listed as *list* in Table 2.

As shown in Table 2, there are currently 49 fields defined in our ITS while there is no unnecessary field. We faced the problem that it was difficult for the CSIRT member to find out which field should have been input. Even though there is no unnecessary field, all fields are not *always* necessary. For example, let us assume that a PC gets infected with malware, and the PC does not contain any confidential information. In this case, CSIRT members do not need to preserve all data stored in the PC for digital forensic since there is no possibility of data breach. CSIRT members do not then need input fields regarding digital forensic. As described above, it depends upon status which field should be input or not.

In order to reduce fields which are displayed in front of the CSIRT member, we have utilized privilege control of Remine. **Figure 3** shows our privilege control for each status and each field. In Fig. 3, "*" represents required field, and "-" represents read only field which is hidden. A blank means that the field is displayed on the status.

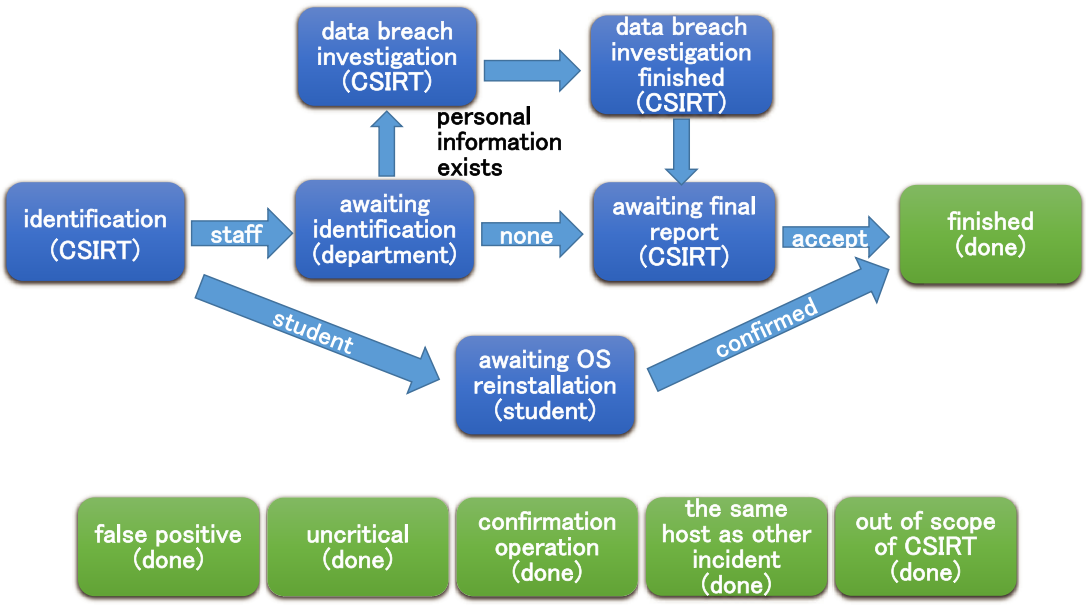


Fig. 2 FSM on ITS.

3. Evaluation

3.1 Shortened Time for an Initial Incident Response

This section presents whether automation and orchestration of an initial incident response can shorten the time for an initial incident response.

Table 4 shows actual times required for manual initial incident responses in Tottori University since January 2017 before January 2018. Table 4 shows only critical incidents that required host isolations and at least one of isolating time or alert mail sent time was recorded. In Table 4, **x** indicates unrecorded time or seconds. Note that dates of incidents are also omitted for anonimity in Table 4. Also note that there actually were incidents that took more than a few days to isolate a suspicious host before January 2018. Those incidents were, however, omitted here because their records were insufficient, and the total time for an incident response could never be calculated.

As shown in Table 4, a manual incident response required at least 6 minutes. 6 minutes were the minimum and such fast handling was only the incident no. 196, and the others tooks more than 15 minutes. As shown in Table 4, incidents happening outside office hours, no. 257 and 289, took more than 30 minutes. Especially, in the case of incident no. 289, it took more than four hours to isolate a suspicious host and send an alert mail. Incidents happening outside office hours might not be manually handled longer especially during weekend or long vacations. It can be said that an incident happening outside office hours is an issue of a manual incident response.

As shown in Table 4, sending an alert mail took more than 10 minutes. In the case of the incident no. 257, it took more than 20 minutes. These longer times might result from searching an IP address from an IP address allocation list, finding a mail address of a PoC, and making a mail message.

On the other hand, **Table 5** shows actual times required for orchestrated initial incident responses in Tottori University since January 2018. As shown in Table 5, all initial incident responses were finished in 40 seconds. As shown in Table 5, sending an alert mail was relatively fast and finished within 1 second. Even incident no. 303 happening outside office hours was handled within 17 seconds. This delay was relatively faster than the manual incident response that required more than four hours in incident no. 289 in Table 4.

Interestingly, in the case of incident no. 302, the host isolation was automatically canceled. Our host isolating system implementation was programmed a *safeguard* not to isolate a host that connected to a 10 GbE link because it would be a VMWare ESXi server. In case of no. 302, this safeguard worked well, and avoided other Virtual Machines (VMs) residing on the same ESXi to be isolated. The suspicious host was actually a vulnerability scanning server, and the server was accessing to servers in our university. Its behavior might look like an attacker. If we had manually handled this incident, we might have isolated the host without any thought or investigation. It can be then said that an automated operation may be able to avoid misoperation resulting from a false positive report.

3.2 Saved Time of CSIRT Members

This section presents how long the proposed system can save CSIRT members time.

Let us take a look at the number of actual alerts from SOCs. **Table 6** shows alerts from SOCs in Tottori University from fiscal years 2014 to 2018. Note that Table 6 does not include alerts that were intentionally generated while CSIRT members investigated the suspicious host. Also note that Table 6 may include alerts that were duplicatedly generated for one suspicious host when the suspicious host had multiple network interfaces. In Table 6, a column of *incident* is the number of alerts that were treated as incident.

As shown in Table 6, there are many alerts especially since 2017 even though most of them are false positve detection. Re-

Table 2 Input fields of a ticket on ITS.

Field	Value Type	Description
ID *1*2	integer	monotonically increasing number.
created time *1*2	timestamp	created time.
updated time *1*2	timestamp	last updated time.
subject *2	text	a subject of an incident: suspicious malware infection, and so on.
description *2	long text	a description of an incident that SOC firstly reports.
priority *2	list	priority of this incident: low, medium, high, very high, extremely high.
a person in charge *2	list	a person in charge in CSIRT.
status *2	list	status of an incident defined in Table 1.
detection	list	detecting institute: commercial SOC, NII-SOCS, MEXT, police, user, CSIRT and other.
type	list	types of incidents: security, physical and contents.
threat	list	threat type: malware, phishing, XSS, defacing, unauthorized access, mail sending miss, DoS, account data breach.
malware name	text	a malware name.
malware type	list	types defined in STIX: adware, backdoor, bot, dropper, exploit-kit, key logger, ransomware, remote-access-trojan, resource-exploitation, rogues-security-software, rootkit, screen-capture, spyware, trojan, virus and warm.
external corresponding IP address	IP address	an IP address of a corresponding host.
internal global IP address	IP address	a global IP address of a suspicious host.
internal private IP address	IP address	a private IP address of a suspicious host.
MAC address	MAC address	a MAC address of a suspicious host
network category	list	a type of a network: education, research, secretariat, guest and other.
LAN type	list	types of media:, wireless or wired.
start time	timestamp	the time when malicious communication is started.
end time	timestamp	the end time when malicious communication is finished.
communication block	list	unapplied, firewall (IP address filtering), core switch (MAC address or IP address filtering), edge switch (port shutdown, MAC address or IP address filtering), wired or wireless LAN authentication (MAC address), wireless LAN controller (MAC address) and released.
host isolation	list	status of a suspicious host isolation: locating or isolating a host, recoverying from isolation and unapplied.
department	list	a department that the network belongs to.
division or section	text	a devision or section that the network belongs to.
user type	list	staff, student or other.
user ID	text	user ID of staff or student.
personal information	list	a suspicious host contains personal information or not.
encryption	list	confidential data is encrypted or not.
data breach	list	data breach is possible or impossible.
SOC ticket number	text	SOC ticket number.
SOC ticket status	text	open, SOC investigating, wating for SOC response, CSIRT investigating, closed, and so on.
SOC notification time	timestamp	the time when a SOC notifies.
OS and version	text	OS and its version of a suspicious host.
security software	text	security software name and version.
personal information types and amount	long text	personal information types such as phone number, name, e-mail address and etc. and theirs amount.
communication log investigation	list	done or not.
identifying infection source	list	done or not.
specimen collection	list	done or not.
static analysis	list	done or not.
dynamic analysis	list	done or not.
obtaining file list	list	done or not.
obtaining start up list	list	done or not.
obtaining task list	list	done or not.
obtaining task scheduling list	list	done or not.
obtaining registry	list	done or not.
forensic	list	done, deleted or not.
countermeasures to prevent recurrence	long text	a description of a countermeasures.
abstract	long text	a brief description of an incident to explain to board members.

garding NII-SOCS alerts, all alerts were false positive detection. In case of NII-SOCS, it is very difficult for CISRT members to manually obtain the required information from a portal site of NII-SOCS. In order to login to the portal site, CSIRT members are requested a two-factor authentication. CSIRT members need to prepare NII-SOCS client certificate and input their own user names and passwords dedicated for NII-SOCS. For security reasons, it may not be suitable to install NII-SOCS client certificates to all hosts that a CSIRT member may use for an incident response. CSIRT members then need to move to a room where there is a host that NII-SOCS client certificates are installed, and it may require a longer time. It actually required more than several minutes to manually obtain the required information. The proposed system can automatically obtain the required information. We also have to register all NII-SOCS alerts to ITS as a record, and it requires much time. NII-SOCS alerts give only a global IP address, and require to identify a private IP address from a firewall. The proposed system can automatically do these operations. NII-SOCS alerts are also sent even if a downloaded malware is quarantined by anti-vrus software. The proposed system automatically checks to see if a downloaded malware is quarantined or not. In 2017, we had 20 alerts from NII-SOCS, and that

*1 automatically generated.
*2 Redmine built-in field.

Table 3 Visibility and permissions of input fields of a ticket on ITS.

Field	Identification (CSIRT)	Awaiting identification (department)	Data breach investigation (CSIRT)	Data breach investigation finished (CSIRT)	Awaiting final report (department)	Awaiting OS re-installation (student)	Abnormal (done)	Finished (done)
ID *	*	*	*	*	*	*	*	*
created time *	*	*	*	*	*	*	*	*
updated time *	*	*	*	*	*	*	*	*
subject *	*	*	*	*	*	*	*	*
description *	*	*	*	*	*	*	*	*
priority *	*	*	*	*	*	*	*	*
a person in charge		*	*	*	*	*	*	*
detection	-	-	-	-	-	-		-
type	-	-	-	-	-	-		-
threat								
malware name								
malware type								
external corresponding IP address								
internal global IP address								
internal private IP address		*	*	*				*
MAC address		*	*	*				*
network category	*	*	*	*	*			*
LAN type	*	*	*	*	*			*
start time								
end time								
communication block	*	*	*	*	*			*
host isolation		*	*	*	*			*
department	*	*	*	*	*			*
division or section		*	*	*	*			*
user type		*	*	*	*			*
user ID		*	*	*	*			*
personal information		*	*	*	*		*	*
encryption		*	*	*	*		*	*
data breach			*	*	*		*	*
SOC ticket number								
SOC incident ID								
SOC ticket status								
SOC notification time								
OS and version			*	*				
security software			*	*				
personal information types and amount	-	-	*	*	-	-	-	-
communication log	-	-		*	-	-	-	-
investigation				*	-	-	-	-
identifying infection source	-	-		*	-	-	-	-
specimen collection	-	-		*	-	-	-	-
static analysis	-	-		*	-	-	-	-
dynamic analysis	-	-		*	-	-	-	-
obtaining file list	-	-		*	-	-	-	-
obtaining start up list	-	-		*	-	-	-	-
obtaining task list	-	-		*	-	-	-	-
obtaining task scheduling list	-	-		*	-	-	-	-
obtaining registry	-	-		*	-	-	-	-
forensic	-	-		*	-	-	-	-
countermeasures to prevent recurrence	-	-						*
abstract	-	-					*	*

Table 4 Time for a manual initial incident response.

No.	SOC reporting time	Isolating time	Alert mail sent time	Total Time (sec.)	Host Locating method	Remarks
289	21:43:02	02:06:14	02:00:18	-	manual	malformed SOC reporting mail.
284	12:58:32	13:25:xx	-	1,620	manual	no alert mail sent.
257	19:48:48	20:05:xx	20:25:15	2,187	manual	
196	16:29:01	16:35:xx	xx:xx:xx	360	manual	
182	16:11:49	16:47:xx	xx:xx:xx	2,160	manual	
172	15:28:38	15:33:xx	15:46:xx	1,080	manual	

means the proposed system may be able to save several hours for CSIRT members.

In addition, we usually had 5 CSIRT members to handle an incident before implementing the proposed system. We now need only 1 CSIRT member to handle all incidents, and other members can focus on their own daily tasks. CSIRT members now also do not need to respond to an incident outside office hours because an initial incident response automatically finishes.

Table 5 Time for an orchestrated initial incident response.

No.	SOC reporting time	Isolating time	Alert mail sent time	Total Time (sec.)	Host Locating method	Remarks
303	23:04:05	23:04:22	23:04:22	17	mongo	
302	11:10:01	11:10:47	11:10:47	36	on-demand	false report, host isolation was automatically canceled.
301	11:38:00	11:38:31	11:38:32	32	mongo	host isolation failure due to bug.
300	10:47:22	10:47:xx	10:47:50	28	mongo	host isolation failure due to bug.

Table 6 The number of alerts from SOCs in Tottori University.

Fiscal year	Alerts from SOCs				Incident	Remarks
	Total	LAC	WideAngle	NII-SOCS		
2014	6	6	-	-	6	
2015	9	9	-	-	9	
2016	26	26	-	-	26	forcely removed all executable files attached to mails since October. (including Microsoft Office files contain macro)
2017	42	5	17 *3	20 *4	19	joined NII-SOCS. changed from LAC to WideAngle in September. installed the next generation firewall in September. implemented the proposed system in October.
2018	36	-	2	34 *4	2	as of 27th March 2019.

3.3 Effectiveness of ITS

This section presents how ITS can improve a manual response of CSIRT members after an automated initial response.

Before utilizing ITS workflow, all CSIRT members except for the author could not properly change status for 42 incidents in 2017. All CSIRT members always asked the author how to change status, and forgot to summarize all incidents that were requested by CISO. In 2018, there have been no questions regarding how to change status, and there have been no incident whose summary is not input even though we have 36 alerts.

ITS has 50 fields at maximum for an incident to be input as shown in Table 3. By hiding unnecessary fields, CSIRT members need to see only 34 fields when they create an incident. As shown in Table 6, there are many false positive alerts, and these invisible fields can be considered as very effective.

3.4 Wrong Host Isolation

There are two benign hosts (out of 73 alerts) that were wrongly isolated due to false positive detection since October 2017. On the other hand, there is no suspicious host that was not isolated except for a software bug.

4. Discussions

This section discusses operational issues regarding security incident handling.

4.1 Reliabilities of SOC Alerts

It may be thought that all alerts should be checked if alerts are false positive or not. We here discuss the reliabilities of SOC alerts, and we present that reliabilities of SOC alerts depend upon a SOC or an organization.

Let us take a look at Table 6 again. As shown in Table 6, the number of alerts from WideAngle in 2017 is 17, and three of those alerts were false positive. False positive detection of WideAngle might be seen right after we migrated from LAC SOC service to WideAngle. This is because our firewall log was not so accurate and included malformed log messages. Since these false positives, we have seen no false positive detection of WideAngle, and we now treat an alert from WideAngle as an incident. Similar to WideAngle, another commercial SOC service of LAC had also no false positive detection from fiscal years 2014 to 2018. We can then say that commercial SOC services that we, Tottori University, contracted may have enough accuracy.

On the other hand, we now treat all NII-SOCS alerts as false positive for us because we consider that they are not so critical so far. Almost all alerts from NII-SOCS were related to *adware* or *coinminer* that we do not regard as critical malware and do not need to immediately isolate an infected host. Other alerts from NII-SOCS were just downloading malware that had been already quarantined by anti-virus software as described in Section 2.8.

As described above, reliabilities of SOC alerts depend upon a SOC. We, Tottori University, then do not have enough human resources to check reliabilities of all alerts. This is one reason that we contract a commercial SOC service. It can be then said that we can believe alerts from commercial SOC services.

In addition, it may be allowed only in our environment to treat all NII-SOCS alerts as false positive because we, Tottori University, contract a commercial SOC service. Other university or organization may not contract such a commercial SOC service, and they may need to treat alerts from NII-SOCS as an incident in order to reduce security risks. It can be then said that the reliabilities of SOC alerts may depend upon an organization.

5. Related Work

Information Security Management System (ISMS) ISO/IEC-27001 [8] briefly defines the requirements of computer security incident responses. There are many security or network vendors such as TrendMicro, Paloalto, FireEye, Fortigate, Cisco, Alaxala and so on who try to produce the best security solutions.

Nagai et al. investigated and reported differences between ISMSs in national universities in Japan [9]. They also presented their own incident management system using trac [6]. 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.

*3 Three alerts were treated as false positive.
*4 All alerts were treated as false postive.

Hasegawa et al. proposes the supporting system against an incident caused by targeted attacks [10]. Their system automatically suggests 9 types of access filtering across VLANs to an administrator in accordance with the severity of an incident when a network configuration is pre-defined and given. They, however, assume only filtering across VLANs, and do not consider the case where there is a router run by a department, not an information infrastructure department that is in charge of a management of a campus wide network. In addition, they do not consider a mobile host that moves around while our proposal does.

Request Tracker for Incident Response (RTIR) [11] is a famous ITS written in Perl. There are also BTSs or ITSs such as trac [6] written in Python, mantis [7] written in PHP and so on. We will try to find the best system for our purpose.

6. Concluding Remarks

This paper has proposed automation and orchestration of an initial computer security incident response using centralized Incident Tracking System (ITS). The proposed system has reduced the time required for the initial incident response to automatically isolate a suspicious host to less than 40 seconds while a manual operation has required more than 30 minutes, several hours or even several days in some cases. ITS workflow has been simplified by the proposed *combined status*, and a CSIRT member has been able to intuitively change the status of an incident without referring any document on an incident response.

References

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

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

[3] 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.

[4] Ohmori, M., Higashino, M., Kawato, T., Fujio, S. and Nakashima, K.: On-Demand Suspicious Host Isolation Adopting Software Defined Network Approach on a Computer Security Incident Response, *Journal of Information Processing*, Vol.27 (2019).

[5] Lang, J.P.: Overview - Redmine (2006), available from ⟨http://www.redmine.org/⟩ (accessed 2017-05-19).

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

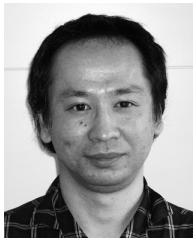
[7] MantisBT Team: Mantis Bug Tracker (2000), available from ⟨https://www.mantisbt.org/⟩ (accessed 2017-05-19).

[8] ISO/IEC 27001:2013: Information Security Management Systems Requirements (2013).

[9] 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).

[10] 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).

[11] Best Practical Solutions, L.: RT for Incident Response (2002), available from ⟨https://bestpractical.com/rtir/⟩ (accessed 2017-05-19).



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 the 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 interests include network architecture, multicasting, routing, mobile networking and energy efficient network operation. He is a member of the IPSJ, IEICE, JSSST, IEEE CS/ComSoc and ACM.