Computer Security Symposium 2018
22 - 25 October 2018

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

Motoyuki OHMORI[1,a] Masayuki HIGASHINO[1,b] Toshiya KAWATO[1,c] Naoki MIYATA[1,d]
Kenichi TAKAHASHI[1,e] Takao KAWAMURA[1,f]

Abstract: A critical computer security incident may cause great damage on an organization such as confidential data breach or malware pandemic. In order to avoid or mitigate such damage, a quick and accurate response against a computer security incident has been then getting more important. In order to realize these quickness and accuracy, this paper presents the Incident Tracking System (ITS) that orchestrates several information systems and automate an initial incident response. The ITS automatically locates and isolates a suspicious host, and sends a mail notification to a 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 getting more attentions 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. To this end, it is now becoming common that an organization forms Computer Security Incident Response Team (CSIRT).

In many cases, a malicious communication is detected by an external organization such as Japan Security Operation Center (JSOC) [1] 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) [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 on an organization caused by an incident, a quick and proper initial response against an incident is important. A quicker initial response can reduce a possibility of data breach itself, also may reduce an operation to investigate data breach. A more proper initial response may be able to avoid misoperation and keep more availability. It may be, however, difficult to make an initial response quicker and more proper.

To this end, we propose to automate and orchestrate an initial incident response using centralized Incident Tracking System (ITS). An initial incident response here indicates to isolate a suspicious host from a network. All processes of an initial incident response are basically automated, and automated processed 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 proper. 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 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 FSM,
- *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

[1] Tottori University, Koyama-minami, Tottori Japan, 680–8550 Japan
[a] ohmori@tottori-u.ac.jp
[b] higashino@tottori-u.ac.jp
[c] t.kawato@tottori-u.ac.jp
[d] miyata@tottori-u.ac.jp
[e] takahashi@eecs.tottori-u.ac.jp
[f] kawamura@tottori-u.ac.jp

and orchestrated incident response 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.

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

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

2.1 Overview

Fig. 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 following sections.

2.2 Logging System

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 13GB per day when the log is stored as a text file. A log stored in a text file is, however, not useful for searching purpose because keywords for a search are not indexed. Database is proper for a search. Database, however, requires more storage space. It can be said that a text file is suitable for long term while database is suitable for 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

Alert parsing system polls a mail box and parse an alert mail sent from a SOC. We have implemented to support an alert mail from 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 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.

Alert parsing system parses above information for other systems. Generally speaking, in case of a commercial SOC service, traffic is monitored in an internal network. Ones can then identify a suspicious host by given IP addresses and port numbers even 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.

Alert parsing system parses 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, alert parsing system accesses to a portal site of an organization of NII-SOCS. Alert parsing system then obtains necessary information for the portal site.

2.4 Host Locating System

Host locating system dynamically locates a suspicious host; the suspicious host is connected to which port on which switch. 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 do not requires 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 locates even a host that is not registered to such host database. Host locating system has two operational modes: *on-demand* and *proactive*.

2.4.1 On-demand Host Locating

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* rout 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 a edge switch accommodating the MAC address, and
(12)produces *location information* of the suspicious host.

2.4.2 Proactive Host Locating

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 logging system. Ones may consider that it is difficult to deploy network authentications to all network equipment. In this case, MAC address authentication can be configured to authenticate all MAC address where it is difficult to deploy network authentications. We, Tottori University, actually enables IEEE802.1x, Web authentication

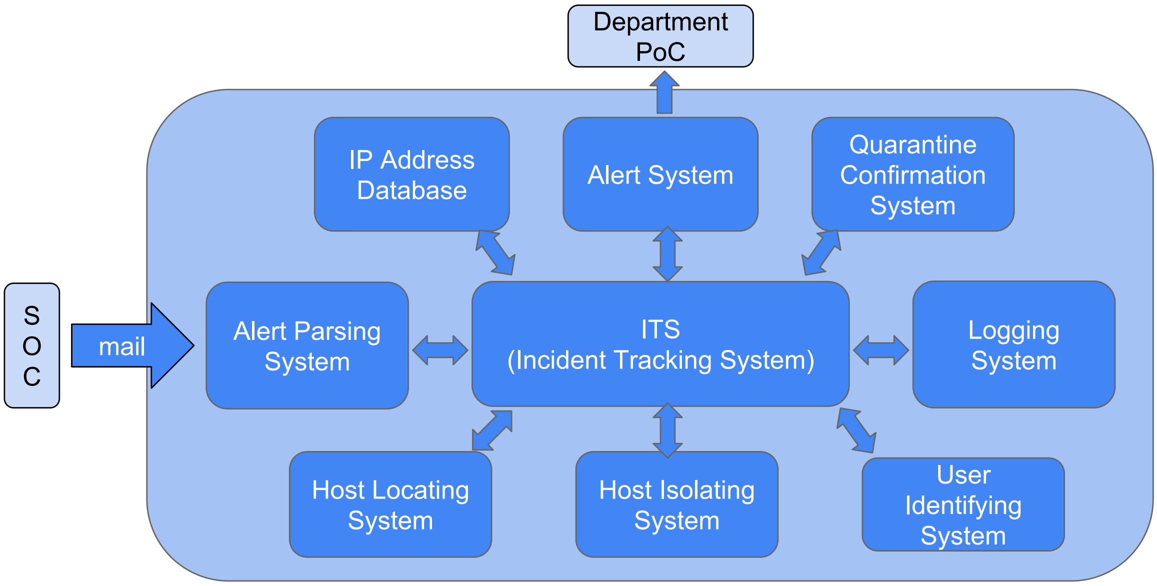

Fig. 1 Overview of components of automation and orchestration of an initial computer security incident response.

and MAC address authentication in all network switches in our university. Host locating system then locates a suspicious host from network authentication log.

2.5 Host Isolating System

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 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.

Host isolating system then operates as follows:

(1) connect to a router or switch that host locating systems 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, and
(4) register its content to ITS.

2.6 IP Address Database

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 ot Contact (PoC),
- department or division,
- section, and
- remark.

2.7 Alert System

Alert system automatically sends an alert mail to departmental PoC in accordance with information given by 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

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. Quarantine confirmation system can then avoid unnecessary host isolation, and mitigate reduction in availability. We have implemented 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 other server using syslog protocol. We have then 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 seach 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. 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 following other systems:

- Shibboleth IdP,
- dovecot, and
- groupware.

These logs are held in logging server as described above. When a suspicious host is not identified by a network authentication, user identifying system searches for a login record from logging system. User identifying system then suggests possible users at the time of 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 make 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 the 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 [4], [5], [6]. ITS, however, needs to assign an incident to a group of CSIRT members while BTS usually assigns to a one person. ITS is very different from BST or issue tracking system in this point. In this paper, we use Redmine [4] as ITS.

2.10.1 Status of a Ticket

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

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, had not experienced to develop a information system from a scratch in real environment or in commercial use. They were, hence, not accustomed to close a ticket. They could not then close a ticket even 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 notify 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 address because we adopts NAT or NAPT for all hosts in our campus network. In our organization, CSIRT is responsible to compute a private IP address from the global IP address. It was, however, difficult for CSIRT to notice at a glance whether this computation was required or not. We had then introduce 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 a person in charge could not notice that he or she should have updated the field. Even the field was properly updated, almost all CSIRT members did not check to see a *ball* field, and did not join an incident handling.

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)

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 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 solve these problems using *workflow* in Redmine. In order to adopt *workflow*, we firstly have modified and defined *status* of Redmine like below.

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

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

2.10.2 FSM for ITS

Using *combined status* as defined in 2.10.1, we define FSM of our ITS as shown in **Fig. 2**. In Fig. 2, each box and arrow represent 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 changes status toward to *closed* status, and there is no event that goes back toward initial status. In addition, all blue boxes have two or less arrows, that is, there are only two choices at maximum when status is changed except for *closed* status. As ones can also see in Fig. 2, lesser critical incident requires lesser status changes. While a really critical incident rarely happens, false positive detection often occurs in our environment. This nature decreases operations that CSIRT member must do on an incident handling. We have then implemented this FSM in Redmine using *workflow*.

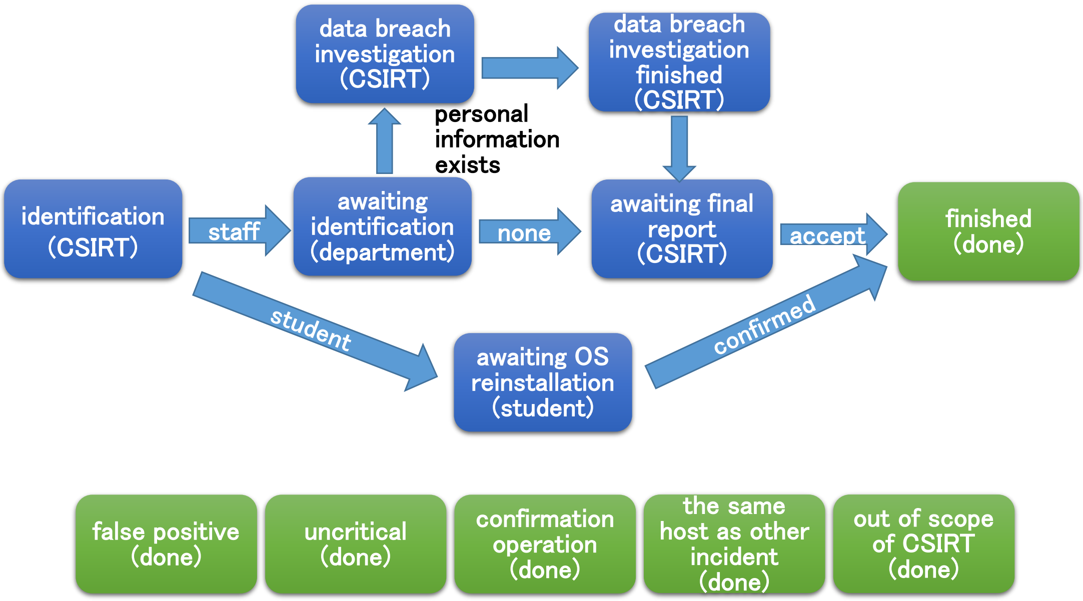
data breach investigation (CSIRT)
data breach investigation finished (CSIRT)
personal information exists
identification (CSIRT)
staff
awaiting identification (department)
none
awaiting final report (CSIRT)
accept
finished (done)
student
awaiting OS reinstallation (student)
confirmed
false positive (done)
uncritical (done)
confirmation operation (done)
the same host as other incident (done)
out of scope of CSIRT (done)

Fig. 2 FSM on ITS.

2.10.3 Input Fields of a Ticket

When ones handle an incident, 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 filed. We faced the problem that it was difficult for 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, ones do not need to preserve all data stored in the PC for digital forensic since there is no possibility of data breach. Ones 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 CSIRT member, we have utilized privilege control of Remine. **Fig. 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.

3. Evaluation

This section presents if automation and orchestration of an initial incident response can shorten 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 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 respones could never be calculated.

As shown in Table 4, a manual incident response required at least 6 minutes. 6 miniutes were the minimum and such fast handling was only the incident no. 196, and the other 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 case of the 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 case of the incident no. 257, it took more than 20 minutes. These longer time 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** show 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 the incident no. 289 in Table 4.

Interestingly, in case of the 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 10GbE 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

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

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.

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.

4. Discussions

This section discusses operational issues regarding an security incident handling.

4.1 Confidentiality of Security Event versus Automation

It would be better, especially for a small organization, to automatically isolate a suspicious host when an external organization alerts an event via an e-mail. We are then planning to implement this automatic host isolation based upon an alert only from an reliable external organization such as JSOC, NII-SOCS and government organizations. We are, however, facing difficulties on implementation. We here discuss the difficulties on an automatic host isolation.

JSOC never includes detailed information such as an IP address of a suspicious host and content of a suspicious communication in

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	-	1620	manual	no alert mail sent.
257	19:48:48	20:05:xx	20:25:15	2187	manual	
196	16:29:01	16:35:xx	xx:xx:xx	360	manual	
182	16:11:49	16:47:xx	xx:xx:xx	2160	manual	
172	15:28:38	15:33:xx	15:46:xx	1080	manual	

an alert e-mail. NII-SOCS never includes an IP address of a corresponding host and port numbers of a transport protocol. They may consider that detailed information is confidential, and should not be sent via a plain e-mail. An operator then needs to manually access to their portal site in order to obtain detailed information. This manual operation ironically takes longer time, and avoids a quick response.

In case of JSOC, JSOC portal site requires a two-factor au-

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.

thentication, and authenticates an operator by a random number in addition to a user name and password. A software protection dongle that JSOC provides generates the random number, and displays the random number on a screen. An operator then needs to manually confirm and input the random number when the operator logs in JSOC portal site. This strict user authentication may protect confidentiality but delays an incident response. To make matters worse, this makes it impossible to remotely isolate a suspicious host because an operator cannot carry a software protection dongle outside of an organization for security. We requested JSOC to disclose detailed information more in an e-mail of an event, and JSOC rejected due to implementation policy. We then consider this authentication is too strict, and prevents us from automating a suspicious host isolation. In addition, an operator needs to manually access JSOC portal site, and this also avoids a quick response against an incident. It may be true that detailed information of an event may be confidential. It may, however, be common, especially for small organizations, that the most of organizations do not have a department dedicated for security. We then think that too much consideration as described above is unnecessary for a small organization, and is rather compromising security. We can then conclude that a two-factor authentication requiring a software protection dongle or client certificate should be avoided and limited to less large organizations because the most of organizations are small organizations. We would propose simpler two-factor authentication such as IP address or others and disclosing more detailed information in an alert e-mail.

5. Related Work

Information Security Management System (ISMS) ISO/IEC-27001[7] briefly defines requirements of computer security incident responses. There are many security or network vendors such as TrendMicro, Paloalto, FireEye, Fortigate, Cisco, Alaxala and so on try to produce the best security solutions.

NAGAI, Y. et al. investigated and reported differences between ISMSs in national universities in Japan[8]. They also presented their own incident management system using trac[5]. 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, H. et al. proposes the supporting system against an incident caused by targeted attacks [9]. Their system automatically suggests 9 types of access filtering across VLANs to an administrator in accordance with a severity of an incident when a network configuration is pre-defined and given. They, however, assumes only filtering across VLANs, and do not consider the case where there is a router run by a department, not a 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 do.

Request Tracker for Incident Response (RTIR) [10] is a famous ITS written in Perl. There are also BTSs or ITSs such as trac[5] written in Python, mantis[6] 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 can have been simplified by the proposed *combined status*, and a CSIRT member has been able to intuitively change a 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 — LAC Co., Ltd., https://www.lac.co.jp/english/service/operation/jsoc.html (1995). Accessed: 2017/05/26.

[2] National Institute of Informatics: National Institute of Informatics, http://www.nii.ac.jp/ (2007). 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] Lang, J. P.: Overview - Redmine, http://www.redmine.org/ (2006). Accessed: 2017/05/19.

[5] Software, E.: The Trac Project, https://trac.edgewall.org/ (2003). Accessed: 2017/05/19.

[6] MantisBT Team: Mantis Bug Tracker, https://www.mantisbt.org/ (2000). Accessed: 2017/05/19.

[7] ISO/IEC: Information Security Management Systems Requirements (2013). ISO/IEC27001:2013.

[8] 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).

[9] 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).

[10] Best Practical Solutions, L.: RT for Incident Response, https://bestpractical.com/rtir/ (2002). Accessed: 2017/05/19.