A Framework for Detecting Insider Threats using Psychological Triggers*

Takayuki Sasaki
Service Platforms Res. Labs.
NEC Corporation
1753 Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa, Japan
Tel:+81-44-431-7686
t-sasaki@fb.jp.nec.com

Abstract

Malicious insiders are difficult to detect and prevent, because insiders such as employees have legitimate rights to access organization's resources in order to carry out their responsibilities. To overcome this problem, we have developed a framework that detects suspicious insiders using a psychological trigger that impels malicious insiders to behave suspiciously. Also, we have proposed an architecture comprising an announcer, a monitor, and an analyzer. First, the announcer creates an event (called a "trigger") that impels malicious insiders to behave suspiciously. Then the monitors record suspicious actions such as file/e-mail deletions. Finally, the analyzer identifies the suspicious insiders by comparing the number of deletions before/after the trigger. In this paper, we extend monitoring reaction from only "data deletion" to "stop further malicious activities". This extension allows a wider variety of use cases such as "finding private web browsing" and "finding use of unnecessary applications". Also, we extend the architecture so as to monitor servers as well as clients. The server monitoring architecture is required in the case of server side data deletions, i.e., e-mail or file deletions at the server side. Moreover, we describe the effectiveness of our approach in such cases.

Keywords: Insider threats detection, sealing of evidences

1 Introduction

In recent years, insider threats have become a big issue. Examples of information leakages in Japan are the leakages of a terrorist-suspect list from a police department and a confidential video from the Japan Coast Guard, which were major news stories in 2010. Furthermore, the biggest business news story in 2009 was 1.5 million people's records being leaked from a finance/insurance company by an insider [2]. As for financial fraud, according to the U.S. CERT report [3], the cost of damage caused by one insider acting illegally may reach $1 million. According to the ACFE report [4], "the typical organization loses 5% of its annual revenue to fraud."

For mitigating these threats, access control and anomaly detection are the well-known existing countermeasures. However, insider threats have two particular features that make it difficult detect or prevent them. One is that malicious insiders have legitimate access rights, and the other is that the malicious access sequences can be similar to those for their responsibilities. The access control approach enforces pre-defined policies on the basis of the access rights for their legitimate jobs, thus it cannot prevent malicious insiders exploiting their legitimate rights. The anomaly detection approach learns normal states using a machine-learning technique and detects anomalies as outliers. However, the anomaly detection approach cannot detect malicious activities similar to the insider's responsibilities.

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 3, number: 1/2, pp. 99-119

*This paper is an extended version of the work originally presented at the 3rd International Workshop on Managing Insider Security Threats (MIST'11), Fukuoka, Japan, December 2011 [1].

To solve this problem, we have proposed an active framework [1] that creates an event (called a "trigger" in this paper) that is expected to impel malicious insiders to behave suspiciously, monitors the reactions of the insiders, and detects suspicious insiders by comparing their actions before/after the trigger. The key idea of the framework is triggering anomalous actions of malicious insiders. The malicious insiders are generally anxious about their malicious activities being revealed. Therefore, the psychological trigger heightens their anxiety and helps us to find suspicious insiders. On the basis of this framework, we have designed an architecture comprising an announcer, a monitor, and an analyzer. First, the announcer announces an investigation as a trigger. Then, the monitor monitors deletions on each client, and the analyzer calculates per-user scores on the basis of the number of deletions.

In this paper, we extend the target reaction patterns that are impelled by the trigger and monitored by the monitor. The malicious insider is expected to react in two patterns after the trigger. One is "delete evidence" to hide their malicious activities; for example, deleting documents and e-mails that show the malicious activities. The other is "stop further malicious activities" after the trigger in order to avoid detection. To deal with these two patterns, the architecture monitors deletion events and access rate.

Also, we extend the architecture for monitoring servers such as file servers and mail (IMAP) servers as well as clients, because the evidence is stored on both clients and servers. The extended architecture enables the solving of many use cases such as "information leakage via e-mail" and "illegitimate purchases".

2 Detecting Suspicious Insiders

2.1 Use case analysis

The problem this paper tackles is detecting suspicious insiders who use legitimate rights to access computer systems for malicious purposes and undertake malicious activities that look similar to their responsibilities.

Next, we clarify the above problem and identify the requirements of insider threat detection using the following four typical use cases. First, we suppose two critical threats: one is information leakage via an e-mail (situation 1) and the other is accounting fraud such as kickbacks (situation 2). Then, we also suppose two situations that are not critical but problematic: private web browsing (situation 3) and use of applications unnecessary for their jobs such game software (situation 4).

Situation 1

Engineer A of organization X and engineer B of organization Y are working on a collaborative project, and A often sends reports regarding the project to B. However, one day, A deliberately sends a confidential report of organization X to B.

In this situation, A often sends reports to B, so the malicious activity (sending a confidential report) is not an anomaly. Thus, the anomaly detection cannot detect this leak, causing a false negative.

Situation 2

Manager A is responsible for purchasing and always purchases from shop B. However, once, he purchased from shop C at a price higher than that of shop B.

The reason for this activity may have been corruption; for example, the manager received a kickback from shop C. However, it may have been for a legitimate purpose; for example, shop B did not have anything in stock at that time.

Obviously changing shops is an anomalous action, so the anomaly detection system can detect it.

Nevertheless, we have many exceptions to general rules in the real world; for example, if the usual shop is out of stock in Situation 2, the anomalous action is probably not dishonest. Thus, anomaly detection systems cause false positives.

Thus, the anomaly detection system is not effective in either situation: it obtains a false negative in Situation 1 and a false positive in Situation 2.

Next, we assume the following two cases.

Situation 3

The insiders such as employees access a lot of web pages. Which pages are necessary for their jobs, and which are unnecessary?

Situation 4

The insiders install and use many applications. Which applications are necessary for their jobs, and which are unnecessary?

In neither case can the anomaly detection system answer the above questions, because “normal and anomaly” are not always related to “necessary and unnecessary”. For example, a frequently accessed web page such as a popular SNS may not be necessary for their jobs.

2.2 Requirements of Insider Threat Detection

According to the above use case analysis, existing anomaly detection obviously has many false positives and false negatives. Additionally, we can identify many types of insider threats. Therefore, we define the following three requirements.

- Fewer false negatives
  The detection system must detect all malicious insiders.
- Fewer false positives
  After detecting events using a detection system, the investigator must check the detected events manually. Therefore, fewer false positives are required.
- Wider coverage of use cases
  As described above, there are many types of insider threat. The detection framework must be able to deal with most cases.

Also, we define an additional requirement.

- Minimal storage
  For a lawsuit or an incident report, the investigator must check the evidence preserved by the system. The simplest way to preserve evidence is for the system to permit adding but not deleting. For example, we can use write-once media such as CD-R and write-once file systems [5]. However, this is not efficient in terms of storage volume, because these media or file systems preserve all information including that of responsibilities. Thus, only information related to the malicious activities needs to be preserved.

3 Detection Framework

The existing frameworks passively monitor insider actions and directly detect the malicious activities. However, as described in Section 2, access sequences for malicious activities are hard to distinguish from those for actual jobs.

We propose an active detection framework that gives a trigger, leading to the detection of suspicious behavior of malicious insiders. The framework distinguishes the malicious insiders from normal insiders on the basis of their actions (Figure 1 lower part). Our framework detects the malicious insiders indirectly by finding suspicious actions prompted by the trigger. Thus, our framework achieves fewer false negatives and fewer false positives. For example, in Situation 2 described in Section 2, the corrupt manager is expected to delete evidence of the purchase, and the detection framework can find the manager on the basis of these deletions.

The proposed detection framework flows as follows.

- Step 1 Trigger: Giving all organization members a trigger that impels malicious insiders to behave suspiciously. For example, the investigator announces that an investigation will start today. This announcement can be issued orally or by e-mail.
- Step 2 Monitoring: Monitoring the actions of the insiders and making logs.
- Step 3 Analysis: Analyzing the logs and detecting the suspicious insiders by identifying the suspicious actions prompted by the trigger.

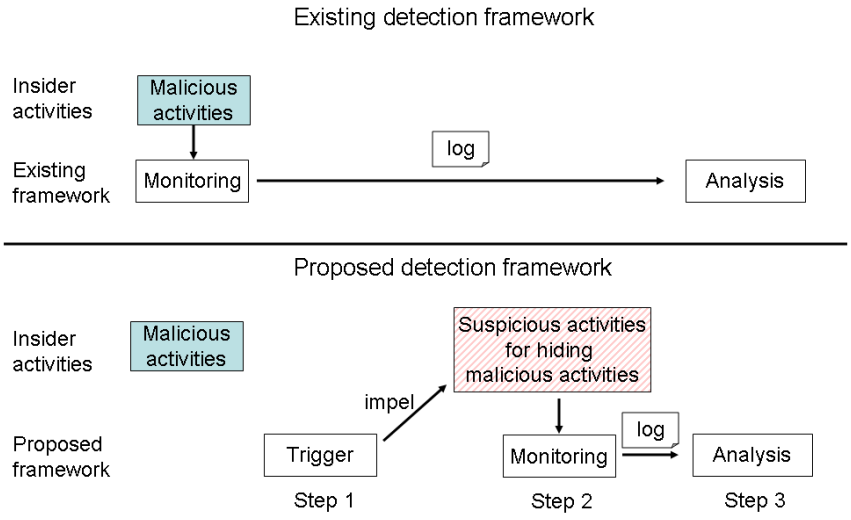

Figure 1: Proposed Framework

3.1 The Effectiveness of the Trigger

In this section, we describe how the proposed framework meets the requirements as mentioned in Section 2.2.

- Fewer false negatives
  The malicious insiders are expected to delete evidence of their malicious activities in order to hide them. These deletions are easy-to-find anomalies, thus the framework achieves fewer false negatives.

  As described in Section 2.1, the existing anomaly detection frameworks directly distinguish malicious activities from legitimate ones. For example, the anomaly detection classifies normal events

and anomalous events in the feature space. In the upper part of Figure2, the event distribution of malicious activities partially overlaps with that of legitimate ones, and the anomaly detection sets a boundary line on the overlapping area. Thus, the overlapping area causes false negatives and false positives.

Our framework gives a trigger and causes suspicious reactions such as deletions of evidence, which have different patterns from legitimate activities. These reactions have different features compared with the legitimate activities, thus we can easily distinguish between these two types of activities (Figure 2 lower part).

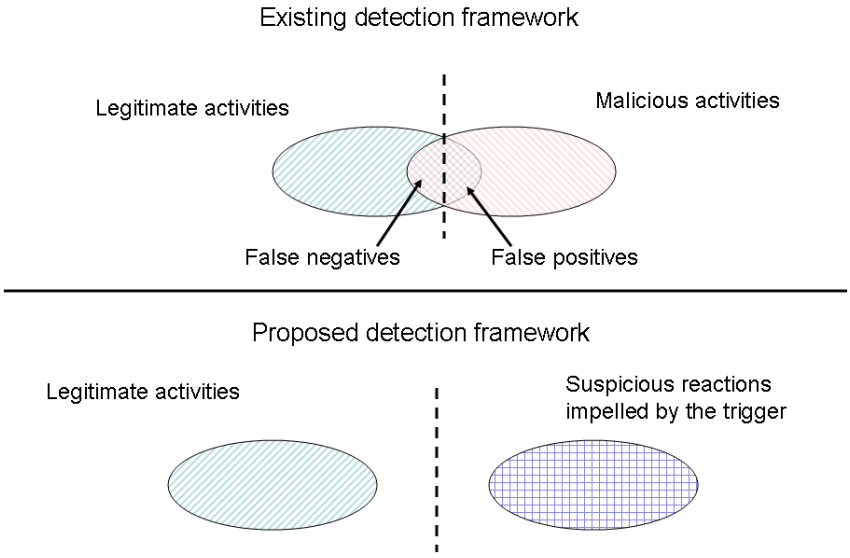

Figure 2: Effectiveness of Proposed Framework

- Fewer false positives
  For this requirement, the proposed framework takes effect on two points. The framework can decrease false positives by triggering suspicious reactions. Moreover, the framework allows shorter time for monitoring and gathering information.

  The number of false positives is calculated as follows.

  number of false positive = false positive rate * number of events

  To reduce the number of false positives, we can change the false positive rate or the number of events to be checked. The false positive rate is hard to change, so this framework tries to reduce the number of events. To do this, the trigger announces the start time of an investigation because the malicious insiders will delete the evidence in the period between the announcement and start of the investigation (Figure 3). Before the trigger, the malicious insiders do not delete the evidence, because they do not know about the investigation yet. After the announced start time of an investigation, the malicious insiders do not delete evidence, because they need to delete the evidence before the investigation.

- Wider coverage of use cases
  As described in Section 2.1, we can identify many cases, and the purposes of the crimes are diverse,

for example stealing money, escaping work, etc. Thus, the action patterns of crimes depend on each case, and existing frameworks have difficult covering many cases.

However, the purposes of psychological reactions caused by trigger are simple: the insiders are made to hide their malicious activities. These psychological reactions do not depend on the purposes of the crimes, thus the proposed framework can cover many cases by monitoring triggered reactions for hiding malicious activities.

- Minimal storage
  As mentioned above, the detection system only needs to preserve the evidence that is deleted within the period after an announcement and before the start.

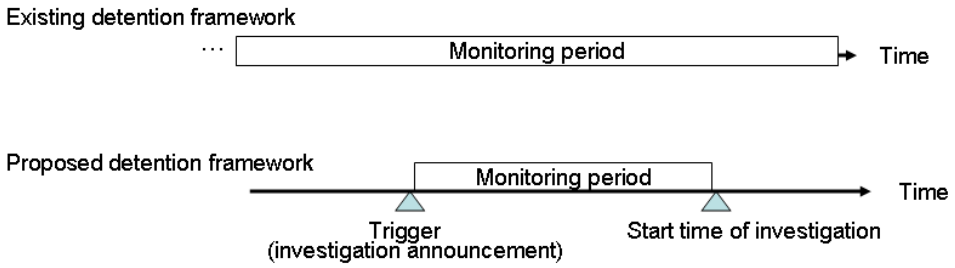

Figure 3: Monitoring Period

4 Architecture

In this section, we describe an architecture based on the framework mentioned in Section 3.

The architecture is comprised of four components.

- An announcer announces an investigation as the trigger.
- A monitor makes a log for recording deletions of evidence or access rate to relevant objects, such as files and e-mails.
- A log store is responsible for storing the log created by the monitor.
- An analyzer compares the numbers of file/e-mail deletions before and after the trigger.

In the following section, we extend the architecture in terms of “type of reactions” and “deployment patterns”. First, we discuss types of malicious insider reactions that are monitored by the monitor. Also, we propose two deployment patterns of the monitor: one for the client monitoring and the other for server monitoring.

4.1 Types of reactions impelled by the trigger

In this paper, we assume two types of reactions: “delete evidence” and “stop further malicious activities” (Figure 4). “Delete evidence” is a proactive reaction for hiding malicious activities that have already been done. On the other hand, “stop further malicious activities” is a more passive reaction with the aim to make oneself less of a target for investigation. Thus, we need to monitor both types of reactions for

dealing with many use cases. For example, we can use the former for finding malicious purchases, and the latter for finding private web browsing (we discuss these cases in Section5).

"Delete evidence" can be broken down into two patterns in terms of monitored objects. One is file deletion and the other is deletions of network messages such as e-mail deletions. In most modern business situations, files and e-mails are used, thus the architecture covers most use cases (we discuss coverage of the architecture in Section 5). Also, "stop further malicious activities" can be broken down into two patterns: stop file accesses and stop network accesses, such as web access and sending/receiving e-mails.

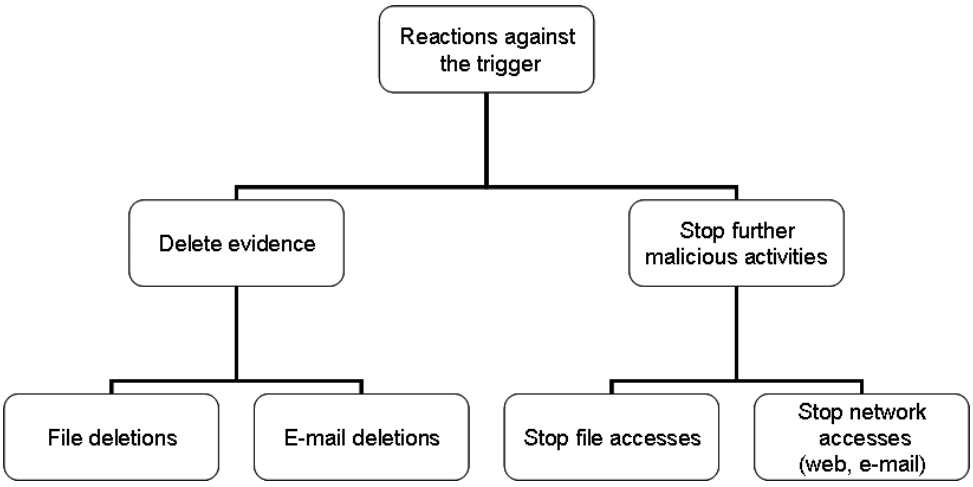

Figure 4: Reaction Types

4.2 Deployment patterns

Next, we discuss deployment of the monitor. We designed two deployment patterns of the monitor: client monitoring and business server monitoring. For client monitoring, the monitors are deployed in each client, and for the business server monitoring, the monitors are deployed in each server such as a file server and an e-mail server.

4.2.1 Architecture for Client Monitoring

The architecture for client monitoring comprises clients and an analysis server (Figure5). The clients are used by the insiders such as organization members, and each client has an announcer to give a trigger and a monitor to record deletions. The analysis server has a log store and a log analyzer to detect suspicious insiders.

In the client, the announcer displays an announcement of when an investigation will start. When the announcement is made, the monitor hooks deletion APIs such, as file deletion APIs provided by an OS and e-mail deletion APIs provided by a mailer. Then the monitor generates logs that record deletions. Furthermore, the monitor preserves backups of the deleted files/e-mails. Also, the monitor records other file operations such as execute and read to find cases of "stop further malicious activities".

In the analysis server, the log store stores logs and backups that are received from the monitors. Using the logs, the analyzer calculates per-user scores that show the possibility of a malicious insider. The backups are used for the investigator to confirm if the client user is suspicious or not.

4.2.2 Architecture for Business Server Monitoring

Figure 6 shows an architecture for business server monitoring. Evidence such as files/e-mails is often stored in the business servers. For example, we often use a file server for storing and sharing the files and an IMAP server for storing the e-mails. For these cases, we need the monitor in the business servers for monitoring operations to evidence..

As for detecting cases of “stop further malicious activities”, the architecture also monitors access operations to relevant objects. We consider three objects to be monitored: files, e-mails, and web pages. Thus, the architecture monitors file accesses on the file server, mail send/receive on the mail server, and web accesses on the web proxy or firewall.

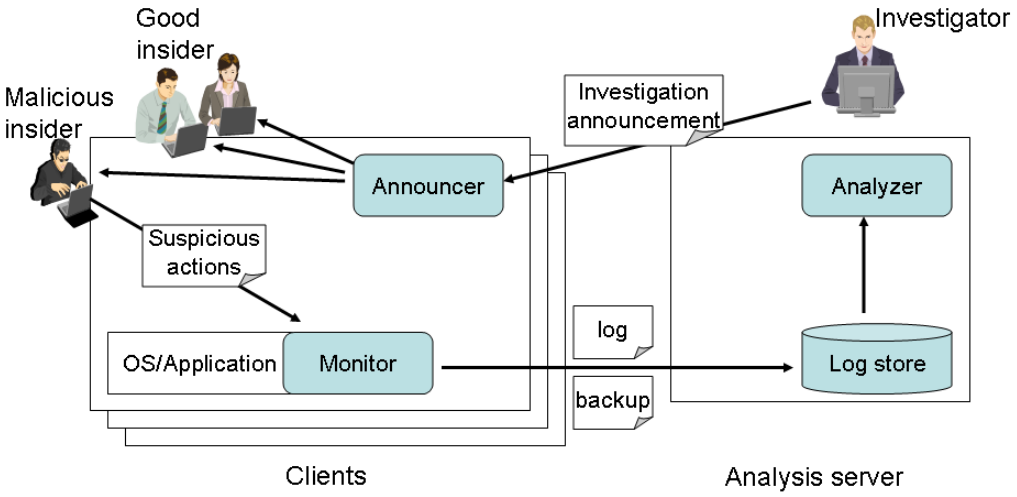

Figure 5: Architecture for Client Monitoring

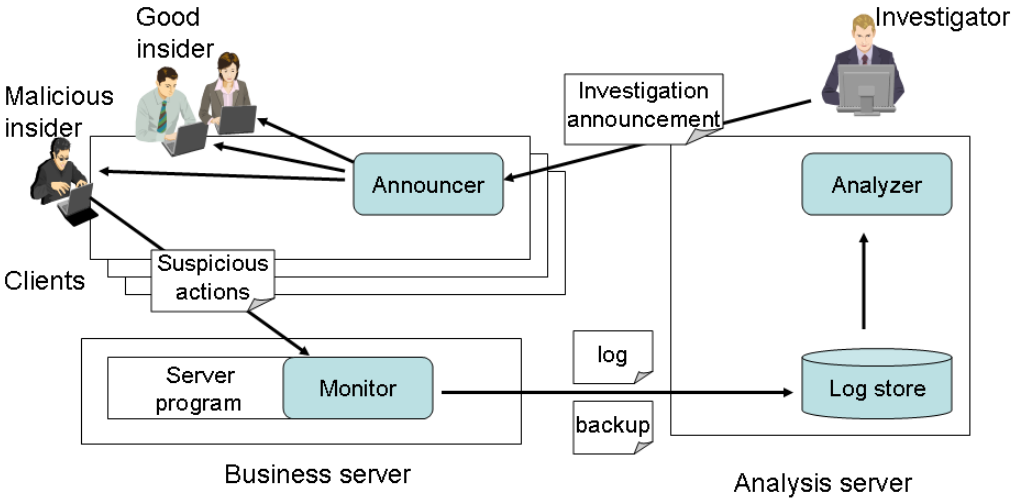

Figure 6: Architecture for Business Server Monitoring

4.3 Announcer

The announcer announces an investigation as a trigger. As the announcer, we can use e-mail or an alert application that opens a popup window to display the announcement. A small organization may not necessarily implement an electronic announcer, but an oral announcement should be enough.

4.4 Monitor

As described above, the monitor records cases of “delete evidence” or “stop further malicious activities”. Deletions of evidence are typical suspicious action caused by the trigger. For recording these actions, the monitor hooks deletion APIs of the OS and mailer. Otherwise, an insider stops further malicious actions if he/she aims to not be focused on. For example, the malicious insider stops private web browsing after the announcement of the web access investigation (we discuss this use case in Section5.1.3). To leverage this reaction for the analysis, the monitor records the access rates to the monitored objects such as files, e-mail address, and web sites

As described in Figure 5 and 6, we need to deploy monitors on two sides (on the clients or on the business servers) because the suspicious actions are executed at both side. Also, we can implement monitors in two ways: a file monitor or a network (e-mail/web) monitor. Thus, we can identify the following four types of monitors (Figure 7).

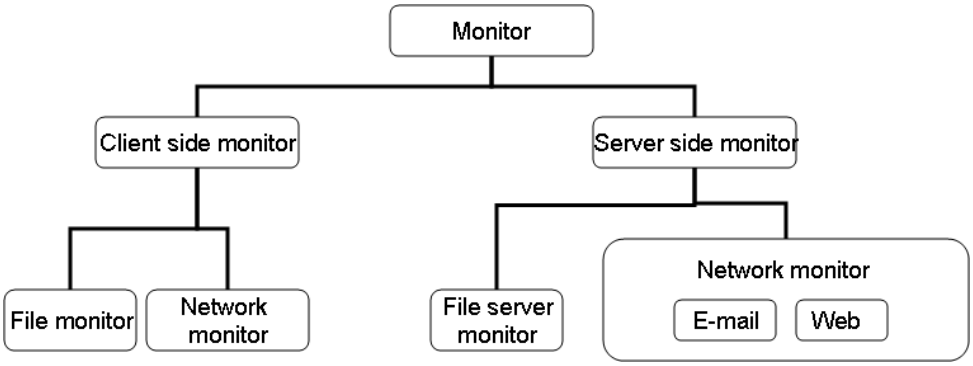

Figure 7: Monitor Types

4.4.1 Client Side File Monitor

The client side file monitor records file deletions on the client. However, files are deleted by users manually or by programs automatically. For example, many programs make temporary files and delete these files automatically. Automatic deletions should be ignored, because these events are not related to the trigger. For this purpose, the monitor records user interface (UI) events in relation to the file deletions. For example on Windows, the monitor records the UI deletion operations such as “right click on a file and select delete” or “push delete key” on Explorer, and the file monitor records “file name” and “deletion time”. Moreover, the file monitor makes backups of the deleted files to preserve the evidence. Finally, the file monitor sends logs and backup files to the server periodically.

Also, the file monitor monitors access rates to the specific target files. The target files are specified by the investigator in consideration of the use case. For example, when the investigator wants to find unnecessary applications, he/she specifies executable files to be monitored.

As for the backup function, the integrity of backups is an important issue as well. For this purpose, some methods have been developed such as a log preserving system using TPM [6], Virtual Large-Scale Disk [7], and so on. However, this is out of the scope of this paper.

4.4.2 Client Side Network Monitor

We designed two network monitors: e-mail monitor and web monitor.

E-mail monitor
The e-mail monitor records e-mail deletions. It can be implemented as a plug-in of the mailer. Some mailers store an e-mail as a file in the MH format. In this case, the mail and file relationship is 1:1, so we can use the file monitor as the mail monitor. Nevertheless, other mailers store some e-mails as a file in the mbox format. In this case, the file monitor observes e-mail deletions as file modifications. Therefore, the e-mail monitor is essential.

The monitor can also monitor e-mail send/receive, but these actions should be monitored by the server side, because the server can record these events at a single point.

Web monitor
The other use case of the network is web browsing. General web browsers record recently accessed web pages, thus the malicious insider may delete the record as incriminating evidence, and the web monitor records the deletions.

The client side web monitor can also monitor stop actions (stop web access). However, the web access log can generally be recorded at the server side.

4.4.3 Server Side File Monitor

The server side file monitor records file delete operations executed on the servers, especially on the file servers. The general file servers such as windows servers, SMB servers, and WebDAV servers create a log to record file operations, and we can use this log for the analysis mentioned in Section 3. From this log, we can extract access rates to the files for detecting cases of “stop further malicious activities”. Thus, for the file servers, we do not need to implement the file monitor. As for backing up deleted files, there are many backup solutions such as shadow copy of the windows server.

4.4.4 Server Side Network Monitor

We designed the following two network monitors.

E-mail monitor
For the e-mail server monitoring, we can also use a log function of the e-mail servers for recording e-mail operations. For example, a DOVECOT IMAP server can record the mail deletion log if the log option is enabled. Also, the monitor monitors e-mail sending for detecting the reaction of insiders who stop their malicious activities. For this propose, we can use the log recorded by the simple mail transfer protocol (SMTP) server.

E-mails are easily backed up, because all e-mails are stored in the server. We only need to backup e-mail folders, e-mail files, or databases of the server.

Web Monitor
The web monitor records web accesses for finding cases of “stop malicious web access”. For this pur-

pose, we can use web proxy, firewall, and web servers as web monitors, because these servers generally record web access logs.

4.5 Log store

The log store receives and stores the logs and backup files from the monitors. The communication paths between the log store and the monitors must be encrypted, because backup files will contain confidential information of the organization.

4.6 Analyzer

The analyzer compares the numbers of files/e-mails deleted/accessed in normal periods and an investigation period. Here, the investigation period is from the announcement to the start of an investigation. A normal period is a time slot the same length as the investigation period. Figure 8 shows an example where an announcement at 9 a.m. says that an investigation will start at 4 p.m. In this example, the investigation period is from 9 a.m. to 4 p.m. today, and normal periods are from 9 a.m. to 4 p.m. on previous days.

The analyzer calculates a score as follows. First, the analyzer counts the number of deletions/accesses in each normal period and calculates the distribution of the numbers. Next, the analyzer approximates the distribution as a Gauss distribution. Finally, the analyzer calculates the score using the following equation.

$$Score = (I - N)/\sigma \tag{1}$$

In this equation, I is the number of deletions/accesses in the investigation period and N is the average number of deletions/accesses in the normal periods. σ is the standard deviation of the distribution of the normal period. Using this equation, the analyzer calculates scores of all users and outputs a list of users who have scores higher than the pre-defined threshold.

In this paper, we use a very simple algorithm, but we plan to make it more sophisticated in order to make the analysis more accurate. For example, we can use anomaly detection algorithms such as support vector machine (SVM) and the neural network. Also, we plan to use probabilistic algorithms such as Bayes inference.

Furthermore, we can weight the file deletion events in accordance with the following rules.

- More weight for deletions caused by emptying a recycle bin
  Windows and Linux desktops such as Ubuntu have a recycle bin. When a user deletes files, the files are not actually deleted from the storage devices but are moved to the recycle bin. This is a common function, so the malicious users know about it. Therefore, malicious users generally empty the recycle bin after deleting files in order to delete the evidence from the storage devices. Innocent users do not empty the recycle bin because most computers have enough disk space. These deletions are given more weight than usual deletions.

- More weight for old file deletions
  Users rarely delete old files that are forgotten about and not usually accessed. We can see the deletion of old files as hiding evidence.

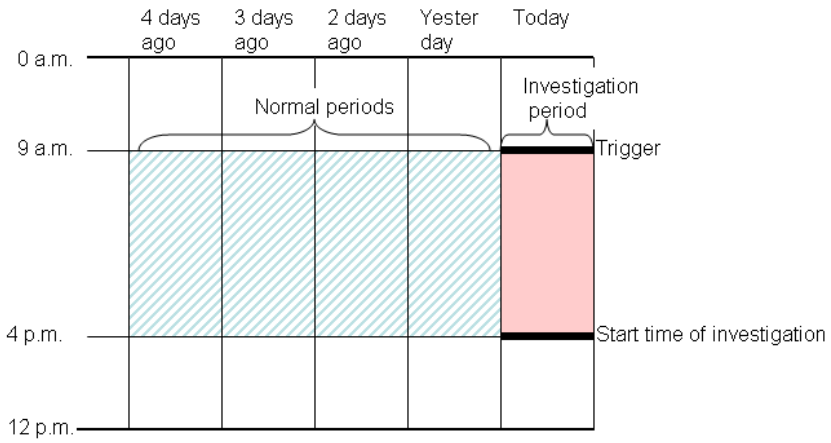

Figure 8: Time Period

Table 1: Use Cases and Reactions to be Monitored

Monitored reaction			Use case			
			Information leakage via e-mail	Accounting fraud	Private web browsing	Use of unnecessary applications
File		Deletions		✓		✓
		Stop actions				✓
Network	E-mail	Deletions	✓	✓		
		Stop actions	✓			
	Web	Deletions			✓	
		Stop actions			✓	

5 Use Cases

5.1 Basic Use Cases

Here, we suppose four basic use cases described in Section 2.1: “information leakage via e-mail”, “accounting fraud”, “private web browsing”, and “use of unnecessary applications”. Table1 shows effective monitors for these use cases. For the information leakage via e-mail, we can use an e-mail monitor that monitors deletions and sending of e-mails. For detecting accounting fraud, we can use the file monitor and the e-mail monitor, because the evidence may be stored in a document, such as an order sheet, or an e-mail. Also, we can find private web browsing by comparing web access rates before/after the trigger, or by detecting history/bookmark deletions after the trigger. Moreover, the use of unnecessary applications can be identified by monitoring applications that are uninstalled or stopped after the trigger.

As described above, the proposed architecture can deal with many use cases (Table 2). Next, we describe the details of each use case and compare the proposed and existing frameworks.

Table 2: Effectiveness of Proposed Framework and Existing Framework

Framework / Use case	Proposed framework	Access control	Anomaly detection	Psychological approach
Information leakage via e-mail	Effective (finding e-mail deletions or "stop sending")	Basic access control is not effective. Workflow approach and virtual domain approach are effective.	Content based anomaly detection is effective.	Effective for identifying risks.
Accounting fraud	Effective (finding e-mail/document deletions)	Workflow approach mitigates the threat.	Many false positives and false negatives	Giving a hint such as disgruntlement.
Private web browsing	Effective (finding stop web accesses or history/bookmark deletions)	Blacklist approach is effective but the list is hard to maintain	Many false positives and false negatives	Hard to make psychological model.
Use of unnecessary applications	Effective (finding uninstall or stop executions)	Whitelist/blacklist approach is effective but the lists are hard to maintain.	Hard to make signature.	Hard to make psychological model.

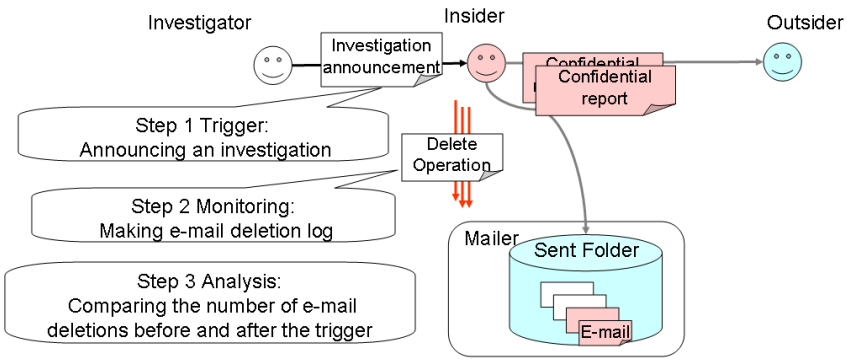

Figure 9: Detecting Information Leakage via E-mail

5.1.1 Detecting Information Leakage via E-mail

The first situation is detecting information leakage via e-mail.

Generally, we do not delete many e-mails before the investigation, but the malicious insider may delete e-mails to hide their malicious activities. Thus, we need to find these deletions to find malicious insiders. (Figure 9)

For this use case, we can use the client monitoring architecture for monitoring mailers on each client. Also, we can use the server monitoring architecture for monitoring the IMAP server.

For finding malicious activities involving e-mails, the system can be configured as follows.

- **Trigger:** Announcing an investigation.
  In this use case, we can use the e-mail system as the announcer and investigations are announced by e-mail. The announcements declare that the investigator will investigate the sent folder of the each client and the mail server to check for information leakage via e-mail.

- **Monitoring:** In the case in which the sent e-mails are stored in each client and mail server, we use e-mail monitors for clients described in Section 4.4.2. The monitor monitors "sent" folders and makes a deletion log on each client. When the organization uses an IMAP server and sent e-mails are stored in the server, we use the server monitoring architecture. For example, we can configure the IMAP server for making a deletion log and use the log for the analysis.

- **Analysis:** Comparing the number of deletions before and after the trigger, and calculating per-user scores.
  The investigator can identify criminal insiders from the suspicious (or high scoring) insiders by investigating deleted mails in the backup, because deleted e-mails have a high probability of containing evidence.

  For improving the accuracy of the analysis, we can weight the events on the basis of rules described in Section 4.6. General mailers have a trash folder and deleted e-mails are moved to that folder, so we can use "More weight for deletions caused by emptying a trash folder". Additionally, the mailers generally record send/receive date and time, thus we can adopt "More weight for old mail deletions rule" on the basis of send/receive time.

In the above configuration, the system monitors cases of "delete evidence", but we can also use cases of "stop further malicious activities" in order to detect information leakage. For this, at monitoring steps, the monitor records e-mail sending. Then at the analysis step, the analyzer compares per-user and per-address e-mail sending rates before/after the trigger. In the case in which the sending rate from an insider to specific e-mail address reduces, the insider may have sent confidential information to that e-mail address.

Next, we discuss advantages of our approach compared with existing approaches such as access control, anomaly detection, and physiological approach.

The basic access control approach cannot solve this use case, because these insiders have legitimate access rights for sending e-mails to outsiders for his/her legitimate jobs. For one of the access control approaches, we can use the workflow approach, which enforces pre-defined business processes. Using this approach, we can define a rule: manager's approval is required for sending e-mails to outside parties. This approach can find and prevent information leakage via e-mail, but the manager needs to check many e-mails including attached files. Thus, it may be impractical in terms of management cost.

Another access control approach is creating a virtual collaboration workplace [8, 9] for a project using virtualization technologies such as virtual machines and virtual network. This approach contains the project information within the virtual workspace using access control mechanisms. This approach is effective for preventing the information leakage from a project. However, this approach is heavyweight, because it uses computer resources (CPU, memory, storage space) for creating the virtual workspace. On the other hand, our framework is lightweight, because our framework only needs to monitor delete operations or access rates.

Next, we think about an anomaly detection approach. In this use case, the insider often sends e-mails to outside parties for legitimate reasons, thus sending e-mails to outside parties is not an anomalous event. Some anomaly detection systems investigate the mail content and determine the normal/anomaly of the e-mail on the basis of its contents [10]. This approach can identify information leakage of this use case, because the content is confidential information. However, an anomaly detection approach requires

signatures to define normal/anomaly, and it is hard to make a signature. Also, the e-mails may include many types of content, such as documents, pictures, videos, and it is hard to deal with all these types.

As for the psychological approach, Frank et al. have proposed risk analysis using a psychological indicator [11]. This approach can identify risk, but it has difficulty finding malicious events or malicious users.

5.1.2 Detecting Accounting Fraud

In the second situation, a manager purchases something at a higher price than usual. The question is "is this a malicious activity or a legitimate one?"

We assume that if the activity is malicious, the manager deletes evidence such as order sheets and e-mails that may show the malicious activity such as a kickback to the manager. Thus, the system can be configured as follows.

- **Trigger:** Announcing an investigation.
  In particular, the announcement declares that the investigator will check contents of files and e-mails stored in the clients and servers for finding illegitimate purchases. The announcement is sent only to purchasing managers in order to limit the scope of the investigation.

- **Monitoring:** Making deletion logs and backups of the documents and e-mails on each client or servers used by the purchasing managers.
  The malicious insiders are afraid of detection of their malicious activities, thus they are expected to delete evidence. Therefore, the system records these events and makes backups of files/e-mails deleted by the insiders. The evidence may be stored in the clients and servers, thus we need to use both client monitoring architecture and server monitoring architecture.

- **Analysis:** Counting and comparing the number of deletions before and after the trigger.
  The system calculates the per-user scores on the basis of the algorithm described in Section4.6 Then the system shows high-scoring users who delete the most files/e-mails.

  Finally, the investigator can identify criminal managers from suspicious managers. In this case, the investigator needs to investigate deleted documents or e-mails in the backup for checking legitimacy.

Next, we discuss effectiveness of existing solutions for this use case. It is hard for access control mechanisms to deal with this use case, because the manager has rights to purchase, and the access control mechanism allows the manager to buy something on the basis of these rights.

Using the workflow approach, we can mitigate illegitimate purchases. The workflow approach can make a rule that requires the boss's approval to carry forward the purchasing procedure. However, the boss needs to check all purchases, so the management cost becomes high. On the other hand, our approach shows suspicious insiders using a scoring algorithm and also shows backups of deleted files/e-mails that are likely to contain evidence of malicious activities, thus the workload of the manager is low in our approach.

Furthermore, the anomaly detection causes false positives, because the anomaly detection cannot distinguish between legitimate and illegitimate reasons.

Also, the psychological approach cannot solve this use case alone, but we can obtain a hint about which insiders are disgruntled, thus we can use our approach and the psychological one together. For example, Bishop et al. pointed out a combined approach [12] that uses a psychological indicator and a cyber indicator (system log) for accurate analysis.

5.1.3 Private Web Browsing in the Office

Here, we suppose a case of finding private web browsing that has nothing to do with work. The proposed system makes a blacklist of unnecessary web pages as a result. In this case, we can use "stop web access" reactions as examples of "stop further malicious activities". Thus, the system can be configured as follows.

- **Trigger:** Announcement of an investigation to check unnecessary web accesses.

  For example, the investigator announces "From today, we will strictly make web access logs and restrict private web browsing".

- **Monitoring:** Making access logs of web browsing at the server side. The general organizations have a web proxy or a firewall that monitor all traffic. Thus, we can use the server monitoring architecture, and monitoring can be executed without modifying the existing web system.

- **Analysis:** By detecting instances of "stop further malicious activities", the analyzer finds unnecessary web pages. The analyzer compares logs before and after the trigger and finds the web pages that are accessed less. On the basis of the access rates the analyzer decides which web pages are unnecessary for work. Then, the analyzer finds the employees who often access these web pages and decides if they browse the web for personal reasons in the office.

The above configuration is for finding cases of "stop further malicious activities", but we can also use cases of "delete evidence" for detecting unnecessary web pages. Generally, the web browser stores history of accessed web pages and bookmarks. After the trigger, the insider may delete histories and bookmarks of unnecessary web pages. For this, at the monitoring step, the monitor on each client monitors deletion of histories and bookmarks. Then at the analysis step, the analyzer counts deletions of histories and bookmarks for each web page and identifies more deleted pages that are unnecessary.

The advantage of our approach for this use case is that the manager can maintain the blacklist of webs page at low cost. Currently, many organizations maintain their blacklists manually, so the cost is high. In our approach, the insiders show unnecessarily web pages by themselves because of the trigger, and the manager only needs to investigate these web pages.

The anomaly detection approach can also mitigate the risk of this use case. It regards frequently accessed web pages as normal and identifies rarely accessed web pages as anomalies. However "frequently access" does not always correspond to "necessary", thus it may cause many false positives and false negatives. For example, a frequently accessed social network service may not be necessary for their jobs.

Psychological approaches such as MERIT [13] may give hints such as risk level or disgruntlement level. But, it is hard to make psychological model, because the reasons of the private web browsing are different for each insiders.

5.1.4 Use of Unnecessary Applications for the Jobs

Next, we suppose a situation where the insiders install unnecessary applications such as game applications onto the company computers and use them during office hours.

For identifying these activities, we can use file access information collected by the file monitor on the clients. We can identify these activities in two ways. One is detecting uninstallations of the applications as examples of "delete evidence". The other is detecting discontinuation of applications as examples of "stop further malicious activities". Thus, the system can be configured as follows.

- **Trigger:** The investigator announces that he/she will start monitoring for installations and executions of applications.

- **Monitoring:** The client side file monitor monitors uninstallations of the applications. It also monitors file executions.

- **Analysis:** The analyzer identifies the uninstalled applications as unnecessary. Also, the analyzer can find unnecessary applications by comparing the execution rates before/after the trigger.

For one of the access control approaches, we can use a blacklist/whitelist approach. Using a blacklist specifying denied applications or using a whitelist specifying allowed applications, we can control the application installations and executions. However, these lists are hard to maintain, because there are many applications. Our approach contributes to making these lists by showing candidates on the basis of the uninstallations and execution rates.

The anomaly detection may solve this problem. However, it requires the signature and is hard to prepare, because there are many types of applications.

Also the psychological approach such as MERIT [13] may solve the problem, but psychological models seem to be hard to make, because we cannot identify reasons for installing unnecessary applications.

5.2 Applied Use Cases (Detecting Accomplices)

Next, we show other cases in which the proposed framework can be used. The above case supposes a single malicious insider, but here we suppose two or more.

To solve this problem, we can introduce a correlation analyzer that finds similar deletion patterns among users. We assume that accomplices often have the same files or e-mails related to a crime. For example, an employee and his/her boss collude and make a fake report. The fake report is stored on both the employee and boss's computers along with e-mails regarding the crime. Thus, the correlation analyzer can detect accomplices by detecting the deletion of the similar files/e-mails among client computers (Figure 10).

To identify the accomplices, the system is configured as follows.

- **Trigger, monitor, and analysis:** This step is the same as that in previous use cases: announcing an investigation as a trigger, counting the number of file/e-mail deletions, and calculating scores on the basis of the numbers.

- **Selecting suspicious insiders:** The system picks up suspicious insiders using the calculated per-user score. The simplest way is picking up insiders over the threshold.

- **Analyzing correlation among selected insiders:** We can calculate correlation by calculating distance in the feature space (Figure 10). First, the system checks the contents of files/e-mails and maps the content to the feature space. For example, we can use the title of document/e-mail, word distribution, and creation date as feature quantity. Next, the system calculates the distance between users A and B. "Near" means that these two users delete similar files/e-mails and seem to be accomplices.

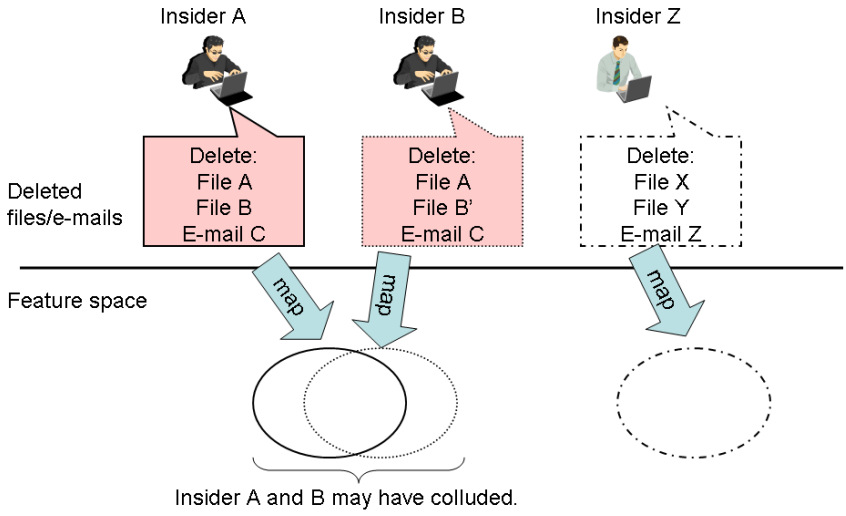

Figure 10: Finding Accomplices

6 Discussion

6.1 Effectiveness against Insiders Who Know the Framework

The proposed framework detects insider threats by giving a trigger and analyzing subsequent actions. However, this framework would not be effective against a person who understood it, because he/she could ignore the trigger. A solution for this problem is hiding the existence of this framework. From the viewpoint of the employees, the trigger would seem to be a true announcement of an investigation, because they would not be aware of this framework.

6.2 Effectiveness against Deletions Prior to the Trigger

In this paper, we suppose that evidence is stored on the client computer. However, when the evidence is deleted before the trigger is issued, these deletions are not prompted by the trigger. Thus, the framework is not effective in this case. To tackle this issue, we need to seek appropriate triggers or other suspicious actions.

6.3 Effectiveness against Physical Destruction of Evidence

The framework needs to monitor operations in which digital evidence is deleted, but the framework cannot detect physical destruction of evidence. For example, insiders may destroy hard disk drives physically.

6.4 Effectiveness against Other Reactions

The proposed architecture monitors deletions of files and e-mails. The malicious insiders may modify files and e-mails to hide their malicious activities instead of deleting them. For example, an insider modifies the amount of money stated in a fake report to make up a cover story. However, modifications are more complex than deletions. To detect malign modifications, we need to consider the context in which they are done. Thus, we need to monitor user actions closely to record how files/e-mails are modified and use more a sophisticated detection algorithm on the basis of the modification context.

6.5 Effectiveness against Deletion for Other Reasons

In the above cases, we suppose that files/e-mails are deleted to hide evidence. However, there may be other reasons, such as privacy, leading to false positives. For example, in small and medium-sized companies, the employees may use a computer for both work and personal business. Therefore, these employees do not want an investigator looking at their private information, so they may delete it.

7 Related Work

7.1 Access Control Approach

For the effective access control, RBAC [14] has been proposed and is used in many systems such as IBM Tivoli. RBAC defines some roles, each of which is a collection of rights (actions and objects) that is assigned to the users. Then RBAC performs access control on the basis of these roles. In this area, role mining methods have been proposed. Role mining is categorized into two types. One is a top down approach [15] that creates roles on the basis of an organization chart. The other is a bottom up approach [16] that creates roles on the basis of the existing access control rules. In addition, a hybrid approach [17] has been proposed.

A Malicious Activities Detection Engine (MADE) [18] has been proposed by Claycomb et al. This system focuses on a directory service and monitors changes in the directory. The system alerts administrators when the directory change violates the predefined policies.

The access control approaches define access control rules in advance and enforce them. Thus, these approaches are effective for routine work. On the other hand, our approach does not require the rules in advance and so can be applied to non-routine work.

7.2 Anomaly Detection Approach

Eberle et al. have proposed a graph-based approach for insider threat detection [19]. This approach models the normal workflows as a graph and detects insider threats as anomalies in the graph. Bertino et al. have proposed an anomaly detection architecture for a database management system [20]. This system parses and extracts some features of SQL commands and then analyzes the SQL commands on the basis of feature values. Chen et al. proposed a Community based Anomaly Detection System (CADS) [21]. CADS analyzes social graphs and finds users with low affinity by using an anomaly detection algorithm when such users behave maliciously. Brancik et al. proposed a data-intensive architecture [22] comprising “event and anomaly collection”, “data analysis and correlation”, and “e-discovery tools” for detection of and protection from insider threats.

These researchers aim to improve anomaly detection efficiency. On the other hand, we focus on how we can trigger anomalous actions of the malicious insiders. Thus, we may be able to improve the efficiency by using both our approach (triggering anomalous actions) and these approaches (detecting anomalous actions).

7.3 Psychological Approach

Greitzer et al. have proposed a predictive risk model for insider threat mitigation [11]. This model uses psychological indicators such as disgruntlement, accepting feedback, anger management, etc. Then the model calculates risks using Bayesian network of the indicators.

Cappelli and Moore et al. have proposed Management and Education of the Risk of Insider Threat (MERIT) [13, 23] for analyzing insider threats. MERIT analyzes insider activities using system dynamics framework [24] for detecting insider threats as early as possible.

8 Conclusion

Existing approaches have difficulty detecting insider threats, because the insiders have legitimate access rights for their job, and use these rights for malicious activities. For this problem, we have proposed a detection framework that creates a trigger that causes malicious insiders to perform actions that bring attention to themselves, such as deleting files and e-mails.

In this paper, we extend the reactions to be monitored: "stop further malicious activities" as well as "delete evidence". Also, we extend architecture for server monitoring as well as clients. These extensions allow a wider variety of use cases. Moreover, we show the effectiveness of the proposed architecture in four basic use cases and an applied use case.

References

[1] T. Sasaki, "Towards detecting suspicious insiders by triggering digital data sealing," in *Proc. of the 3th International Conference on Managing Insider Security Threats (MIST'11), Fukuoka, Japan*. IEEE, November-December 2011, pp. 637–642.

[2] Japan Network Security Association, "2009 incident survey (japanease)," http://www.jnsa.org/result/incident/data/2009incident_survey_v1.1.pdf, September 2009.

[3] E. Kowalski et al., "Insider threat study: Illicit cyber activities in the government sector," http://www.cert.org/archive/pdf/insiderthreat_gov2008.pdf, January 2008.

[4] Association of Certified Fraud Examiners, "2010 Report to the Nations - Introduction," http://www.cert.org/archive/pdf/insiderthreat_gov2008.pdf, 2010.

[5] "WFS: A Write-Once Read-Many File System," http://worm-filesystem.sourceforge.net/, 2010.

[6] B. Böck, D. Huemer, and A. Tjoa, "Towards more trustable log files for digital forensics by means of "trusted computing"," in *Proc. of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA'10), Perth, Australia*. IEEE, April 2010, pp. 1020–1027.

[7] A. Tomono, M. Uehara, and Y. Shimada, "Transferring trusted logs across vulnerable paths for digital forensics," in *Proc. of the 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM'09), Kuala Lumpur, Malaysia*. ACM, December 2009.

[8] A. Bussan et al., "Trusted Virtual Domains: Secure Foundations for Business and IT Services," IBM Research, Tech. Rep. rc23792, January 2005.

[9] T. Sasaki, M. Nakae, and R. Ogawa, "Content oriented virtual domains for secure information sharing across organizations," in *Proc. of the 2010 ACM workshop on Cloud computing security workshop (CCSW'10), Chicago, Illinois, USA*. ACM, October 2010, pp. 7–12.

[10] S. Kiyomoto and Y. Miyake, "On data importance analysis," in *Proc. of the 3th International Conference on Managing Insider Security Threats (MIST'11), Fukuoka, Japan*. IEEE, November-December 2011, pp. 628–633.

[11] F. L. Greitzer, P. R. Paulson, L. J. Kangas, L. R. Franklin, T. W. Edgar, and D. A. Frincke, "Predictive modeling for insider threat mitigation," Pacific Northwest National Laboratory, Tech. Rep. PNNL-SA-65204, April 2009.

[12] M. Bishop, S. Engle, D. A. Frincke, C. Gates, F. L. Greitzer, S. Peisert, and S. Whalen, "A risk management approach to the "insider threat"," in *Insider Threats in Cyber Security, Advances in Information Security*, C. W. Probst, J. Hunker, D. Gollmann, and M. Bishop, Eds., vol. 49. Springer-Verlag, 2010, pp. 115–137.

[13] D. M. Cappelli, A. G. Desai, A. P. Moore, T. J. Shimeall, E. A. Weaver, and B. J. Willke, "Management and Education of the Risk of Insider Threat (MERIT): System Dynamics Modeling of Computer System Sabotage," http://www.cert.org/insider_threat/, June 2008.

[14] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, February 1996.

[15] H. Roeckle, G. Schimpf, and R. Weidinger, "Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization," in *Proc. of the 5th ACM workshop on Role-based access control (RBAC'00), Berlin, Germany*. ACM, July 2000, pp. 103–110.

[16] M. Kuhlmann, D. Shohat, and G. Schimpf, "Role mining - revealing business roles for security administration using data mining technology," in *Proc. of the 8th ACM symposium on Access control models and technologies (SACMAT'03), Villa Gallia, Como, Italy*. ACM, June 2003, pp. 179–186.

[17] M. Frank, A. P. Streich, D. Basin, and J. M. Buhmann, "A probabilistic approach to hybrid role mining," in *Proc. of the 16th ACM conference on Computer and communications security (CCS'09), Chicago, Illinois, USA*. ACM, November 2009, pp. 101–111.

[18] W. R. Claycomb and D. Shin, "Detecting insider activities using enhanced directory virtualization," in *Proc. of the 2010 ACM workshop on Insider threats (WIT'10), Chicago, Illinois, USA*. ACM, October 2010, pp. 29–36.

[19] W. Eberle and L. Holder, "Graph-based approaches to insider threat detection," in *Proc. of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies (CSIIRW'09), Knoxville, TN, USA*. ACM, April 2009.

[20] E. Bertino and G. Ghinita, "Towards mechanisms for detection and prevention of data exfiltration by insiders," in *Proc. of the 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS'11), Hong Kong*. ACM, March 2011.

[21] Y. Chen and B. Malin, "Detection of anomalous insiders in collaborative environments via relational analysis of access logs," in *Proc. of the 2011 ACM conference on Data and application security and privacy (CODASPY'11), San Antonio, Texas, USA*. ACM, February 2011, pp. 63–74.

[22] K. Brancik and G. Ghinita, "The optimization of situational awareness for insider threat detection," in *Proc. of the 2011 ACM conference on Data and application security and privacy (CODASPY'11), San Antonio, Texas, USA*. ACM, February 2011, pp. 231–236.

[23] A. P. Moore, D. M. Cappelli, and R. F. Trzeciak, "The "Big Picture" of Insider IT Sabotage Across U.S. Critical Infrastructures," Carnegie Mellon University/Software Engineering Institute, Tech. Rep. CMU/SEI-2008-TR-009, May 2008.

[24] The System Dynamics Society, http://www.systemdynamics.org/index.html.

Takayuki Sasaki received B.S. degree in Applied Physics from Tohoku University in 2004 and M.S. degree in Physics from the University of Tokyo in 2006. He is currently a researcher of NEC Service Platforms Research Laboratories. He joined NEC in 2006, and has worked on access control mechanisms and security management architecture for cloud computing infrastructures.