4-3 A Suspicious Process Analysis in Cooperation with End Hosts

Junji NAKAZATO, Yu TSUDA, and Yaichiro TAKAGI

The targeted attacks cannot be prevented easily, because a malware that is used in the attack is difficult to detect by antivirus software. Consequently, the malware has been active for a long term in order to access important and serious information in targeted organization. In this paper we propose a new suspicious process detection scheme. The proposed scheme decides suspicious degree of a process by calculating feature value constructed with process frequency and number of users executing the same process. Moreover, we use the network conditions, such as communication of a process in order to reduce a false positive.

1 Introduction

Targeted attacks first skillfully utilize tools such as social networks to gather information on targeted opponents and organizations. After that, they attempt to contact directly using common communication tools such as email. At this time, for the purpose of installing a backdoor for intrusion into the organization, they attach malware to the email or include a link for downloading malware. The attached malware or link needs to be executed by the recipient (targeted party) by himself, but the attacker skillfully uses the information obtained in advance to have the opponent execute the attached file without distrust. In addition, the malware etc. used makes it difficult to discover that there was an attack, such as avoiding detection by antivirus software. Therefore, the targeted opponent will be late in noticing the fact that he was attacked, which allows illicit activities in the organization for a long time. In fact, an incident occurred where Japan Aerospace Exploration Agency (JAXA) was infected by malware during a targeted attack (intrusion by email) on March 17, 2011. There is a high probability that much information was leaked by malware without noticing infection for over a year and a half, until it was discovered the following year on November 21, 2012 by antivirus software [1]. Thus, in endpoint measures that use antivirus software to prevent intrusion of malware against target-type attacks, there is a risk of inviting serious incidents such as information leaks due to missed detections. It is no longer possible to say one is thoroughly prepared. In particular, when a targeted attack was not detected and the target was infected by malware for a long period of time, this gives the attacker many opportunities to gather information in the organization, and make new attacks. For example, it gives opportunities to explore various services, their administrator accounts and specific important systems, and gives opportunities for intrusion, increasing the possibilities of reaching greater quantities of confidential information. Therefore, as well as countermeasures to prevent intrusion itself (entry countermeasures and endpoint countermeasures), internal countermeasures that assume infection are becoming important [2].

A known feature of targeted attacks is combined use of various tools, including Remote Administration Tools (RAT) for controlling operation of a host in the target organization, and administrator commands for obtaining information about the OS, etc. That is, processes that differ from applications that general users use regularly (such as the Office Suite) will be executed in the target host during the attack process. Therefore, when infected with malware containing these tools due to execution of a file attached to an email as mentioned above, etc., processes may occur from this user and other users in the organization, who never executed them in the past. Also, it is conceivable that communication with the outside occurs, such as when using the RAT tool or downloading new malware, so it is important that process monitoring is performed that considers the communication status (standby port, communication destination address, etc.) of each process. In fact, it is known that the Japan Pension Service was infected by malware when a targeted attack email was opened, and illicit communication occurred [3]. Therefore, in countermeasures assuming malware infections, it may be effective to detect the occurrence that are normally used rarely and the occurrence of communications that differ from normal, as soon as possible. In particular, it can be said that these are very effective means in well managed environments where applications used in the organization, etc. are controlled.

Therefore, this research proposes a method for periodically obtaining information on processes operating in the terminal of each user (process list), and judging whether the processes are normally used by the user. For the process judgment, the process name and place where the process was executed (execution path) are compared at the same time. Including the execution path in the process comparison makes it possible to identify illicit processes masquerading as normal processes. An attack tool is very likely to operate periodically, so it is also very likely that a process operated only by specific users over a long period is a very suspicious process. Therefore, for the suspicion level of a process, we define the occurrence feature using frequency of occurrence of processes, number of users who use the same process, period of execution of the process, etc. The communication state of each process is monitored, and processes that conduct suspicious communications are detected based on the occurrence frequency of the communication destinations seen for each process, number of hosts communicating with the same communication destination, etc. When a process occurs with greater than a certain suspicion level, that process is monitored and more detailed information is obtained (for example, history of APIs used), which can be expected to find malicious processes such as malware at an early stage, and lead to prompt countermeasures.

This paper uses process information obtained from actual users, to show relationships between frequency of occurrence of a process, number of users of the same process, frequency in case of communication, etc. This shows that processes with higher occurrence frequency tend to have many users, so very suspicious processes are those with high occurrence frequency and few users, and those with low occurrence frequency but few users and long execution periods. Therefore, considering the communication state of each process, processes involving suspicious communication are detected. This actually shows that many processes with high occurrence features involve communication with low occurrence features (communication frequently used in others). Finally, the usefulness of this method is simulated from information clarified by the incident at the Japan Pension Service.

2 Related research

Countermeasures against targeted attacks are roughly classified into three categories: entrance countermeasures to prevent intrusion of attacks, internal countermeasures to prevent further damage even if they were able to intrude, and exit countermeasures to prevent important information from leaking outside.

Entrance countermeasures include a firewall and IPS/IDS that prevent malware from reaching the user in the first place, and even if malware reaches the user, to prevent infections, boundary defense is performed by antivirus software etc. that prevents intrusion of malicious software. However, especially in targeted attacks, it is difficult to completely prevent intrusions, partly because they may avoid countermeasures to prevent these intrusions of malware from outside. Therefore, internal and exit countermeasures are very important.

Since the beginning of the year 2000, targeted attack surveys are being conducted by JPCERT Coordination Center (JPCERT/CC) [4], Information-technology Promotion Agency (IPA) [5], etc. [6]–[8]. In attacks aimed at information theft, it is shown that in order to reduce the risk of attackers, they increase stealth and make highly accurate attacks. Increased stealth is said to benefit the attacker, by reducing risk of attacks being found and enabling longer-term attacks.

Reference [9] proposes countermeasures against attacks that attempt intrusion by attachments to email, etc. For when targeted attack emails are sent from a remotely operated computer, this proposes a method to compare the behavior characteristics of that person obtained in advance vs. the characteristics when operated by an attacker, and judge whether the transmitted emails are attacks.

In internal countermeasures, much research is being done on countermeasures focused on network activities performed by infected sacrificial hosts in response to targeted attacks [10]–[12]. In reference [10], changes in data trends from time series characteristics obtained according to source IP address, destination IP address, and destination port number are observed, and suspicious communications are detected. It is possible to discover suspicious communications earlier than ChangeFinder [13] which is the conventional method. References [11][12] focus on attack methods (choke points) that all attackers must use internally, such as remote control of target nodes in the process of expanding the attack infrastructure.

By using chokepoints, one judges whether there is inconsistency/abnormality in the behavior of the system, which has successfully detected attacks by malware that avoids antivirus software, such as attacks pretending to be a legitimate program, attacks disguised as legitimate communications, or malware sub-varieties, unknown malware, obfuscation, etc. These observe external activities such as network communications performed by infected hosts, but reference [14] focuses on the parent-child relationship of processes operating in the terminal, and proposes a method of identifying new processes. In particular, the identification method was improved in reference [14] by normalizing execution path information, which caused the erroneous judgments in reference [15]. This paper proposes a method to judge suspicious processes focusing on the network state, in addition to execution frequency and execution period of the process.

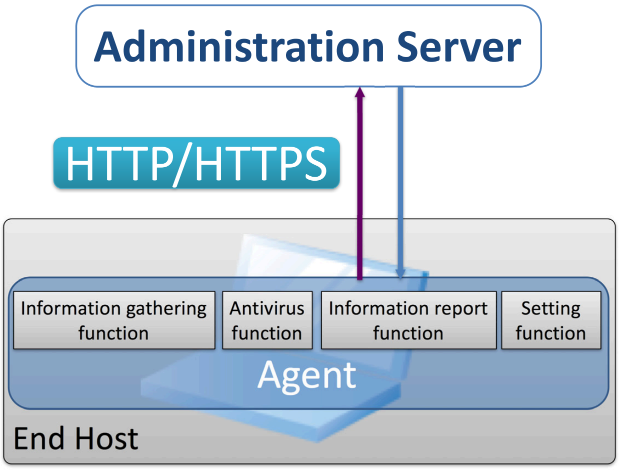

Fig. 1 Agent diagram

3 Process features

Here, we show how to calculate the occurrence features of a process from the occurrence frequency, number of users, and the execution period, of processes operating on each terminal.

3.1 Process list acquisition

Processes operating on each terminal are acquired in the same manner as in reference [15]. An outline of a system for acquiring a process list from each terminal will be described. An agent tool for information acquisition is installed on each terminal, and it periodically sends the process list to the management server. Figure 1 shows a diagram of an agent that collects information.

The agent tool periodically records operations of processes being monitored in each terminal. Specifically, it obtains information on the process ID, process name, CPU usage ratio, memory usage amount, process state, parent process ID, process execution path (storage location of the executed process), process creation time, and network status. The parent process ID represents the parent process

```
<?xml version="1.0" encoding="utf-8”?>
<nirvana_request message_type="1" version="2" request_datetime="2015-08-10 18:00:00">
  <host_information id="a799ebba9388...cb825ae9d">
    <!-- Information of network interface -->
    <network_interface macaddr="**:**:**:**:**:**">
      <ipaddress addr="***.***.***.***"/>
    </network_interface>
    <!-- logon user name -->
    <logon_user user="******"/>
    <!-- System information -->
    <os_information os="Windows 7" service_pack="Service Pack 1" architecture="x86"/>
  </host_information>
  <!-- Detected a malware or not already -->
  <detected_state detected="true"/>
  <!-- Process information -->
  <process_information_list>
    <process_information id="2016" name="CcmExec" cpu="0.0" mem="34148352” status="Execute"¥
          parent_id="568” path="C:¥Windows¥System32¥CCM¥CcmExec.exe" creation_time="2015-08-10 09:00:38.656">
      <tcp_state ip_src="***.***.***.***" port_src="49296" ip_dst="***.***.***.***” port_dst="80" state="ESTABLISHED"/>
      <udp_state ip_src="127.0.0.1" port_src="58798"/>
    </process_information>
                  :
  </process_information_list>
                  :
</nirvana_request>
```

Fig. 2 Process status report XML example

that executed that process, and this information makes it possible to reproduce the process tree (parent-child relationship of the processes).

In this system, information obtained by the agent is periodically sent to the management server via HTTP. In networks managed by companies etc., connectivity from the network segment of the client to the network segment of the server is ensured, but in the opposite case, it is often not possible to connect directly. Therefore, by using a periodic polling method from the agent side, effects of connectivity due to the network environment is minimized.

The information sent by each agent to the management server by polling is sent in the XML format shown in Fig. 2.

Figure 2 shows the result of polling performed at 18:00 on August 10, 2015. The process information operating on each terminal is recorded in the <process_information_list> tag. In this example, the parent process whose process ID is 568 (parent_id="568") executes the CcmExec process (name="CcmExec") whose process ID is 2016 (id="2016") and the execution time is 9:00:38 (creation_time="2015-08-10 09:00:38.656"). In addition, it shows that the actual state of the executed program exists in "C:\Windows\System32\CCM\CcmExec.exe". In addition, this process is network-connected, and we see that TCP protocol is used to communicate from port 49296 to port 80. Furthermore, it can be confirmed that it is waiting for UDP connection at port 58798. Information sent from the agent is stored and managed in a database on the management server.

3.2 Obtain frequencies of processes

individual processes are counted as the same process if they ran separately during a certain period and had two matching elements: same process name and same execution path. Even if there are processes having the same name but different paths, they can be identified as different processes by including the execution path in their comparison. For example, if an illicit process executes using the name of a normal process, it can be identified as a non-normal process because the execution path of the process differs. In this system, process information is collected by periodic polling from the agent, so processes are reported as the same process if these five factors are the same at different times: process ID, parent process ID, process name, execution path, and process creation time. These processes are continuously running processes, so a process with different process ID and process creation time is counted as a newly executed process.

3.3 Process occurrence features

References [16] and [17] show that usual processes (that are not malicious) satisfy four conditions:

- Frequent occurrence of process, and many users
- Many users, even if the process occurs infrequently
- Frequent occurrence of process, and many usage days
- Many usage days, even if the process occurs infrequently

Therefore, occurrence feature $F_{i,j}$ of process P_j used by user $\mathcal{U}_i$ is defined as follows.

$$pf_{i,j} = \frac{p_{i,j}}{\sum_k p_{i,k}} \tag{1}$$

$$uf_j = \frac{|\{u:u \ni P_j\}|}{|\mathcal{U}|} \tag{2}$$

$$df_{i,j} = \frac{|\{d:d \ni P_j\}|}{|D_i|} \tag{3}$$

$$F_{i,j} = pf_{i,j} \times \left(\log_2 \frac{1}{uf_j}\right)^{df_{i,j}} \tag{4}$$

Here, Equation (1) indicates the occurrence frequency of process P_j. $p_{i,j}$ indicates the number of occurrences of process P_j used by user $\mathcal{U}_i$, and $\sum_k p_{i,k}$ indicates the total number of processes used by $\mathcal{U}_i$. Equation (2) indicates the user frequency of the process P_j (proportion of users who use process P_j). $|\{u: u \ni P_j\}|$ indicates the number of users who executed process P_j, and $|\mathcal{U}|$ indicates the total number of active users. Equation (3) indicates the usage frequency of process P_j (ratio of the number of days on which process P_j was executed). $|\{d: d \ni P_j\}|$ indicates the number of days on which process P_j was executed, and $|D_i|$ indicates the number of days when user $\mathcal{U}_i$ became active.

For example, if user $\mathcal{U}_i$ became active for 14 days ($|D_i|$= 14) of the past 14 days and used the "chrome" process seven times ($p_{i,j} = 7$) in 7 days ($|\{u: u \ni P_j\}| = 7$), and used other processes three times during the period (all processes $\sum_k p_{i,k} = 10$), and 3 ($|\{u: u \ni P_j\}| = 3$) of 10 users ($|\mathcal{U}| = 10$) used the "chrome" process, then the occurrence frequency $pf_{i,j}$ is 7/10 = 0.7, the user frequency uf_j is 3/10 = 0.3, and the usage frequency $df_{i,j}$ is 7/14 = 0.5. Therefore, the occurrence feature $F_{i,j}$ of process $p_{i,j}$ executed by the user $\mathcal{U}_i$ is

$$\begin{aligned} F_{i,j} &= 0.7 \times \left(\log_2 \frac{1}{0.3}\right)^{0.5} \\ &= 0.92 \end{aligned}$$

Figure 3 shows the occurrence features of processes ($\sum_k p_{i,k} = 146$ processes) that a user used during 1 hour from 9:00 to 10:00, for the 14 day period before December

1, 2015 (within which, $|D_i| = 9$).

From Figure 3, it can be seen that there are 6 processes with occurrence features greater than 0.05, which are regarded as suspicious processes in reference [16]. In reference [16], it seems caused by the execution path differs depending on the user environment, such as the fact that the actual state of the process exists in the user directory as the cause of the appearance feature becoming higher. Actually, in four out of six processes whose occurrence characteristics were 0.05 or greater, the actual state of the process existed in the user directory (c:¥%USERPROFILE%).

4 Network features

Here we show how to calculate the occurrence characteristics of the network, from the frequency of communications performed by processes running on each terminal and the number of access hosts that are the same.

4.1 Communication frequency

When the process involves communication, the communication information is acquired from the process list shown in Section **3.1**. In the communication information, when the network connection is established (state = “ESTABLISHED” in Fig. 2), and the communication destination IP address or port number are on standby (state = “LISTEN” or UDP protocol communication in Fig. 2), then information on the standby port can be obtained

The occurrence frequency of communication N_l of the processes used by the user $\mathcal{U}_i$ is defined as

$$nf_{i,l} = \frac{n_{i,l}}{\sum_k n_{i,k}}$$

Here, $n_{i,l}$ indicates the number of occurrences of processes having the same communication information as N_l, among the processes executed by $\mathcal{U}_i$. $\sum_k n_{i,k}$ is the total number of processes accompanied by communication used by $\mathcal{U}_i$. Therefore, this shows the proportion of processes that have the same communication information, among processes involved in communication. For example, if there are two processes ($n_{i,l} = 2$) where the process of HTTP (80/TCP) connection to the destination IP address 203.0.113.13 exists, and 10 processes involved communication out of all processes executed during a certain period ($\sum_k n_{i,k} = 10$), then the occurrence frequency $nf_{i,l}$ is 2/10 = 0.2.

Figure 4 shows the distribution of occurrence features and communication frequency of processes that a user used during 1 hour from 9:00 to 10:00 for the 14 day period before on December 1, 2015.

From Figure 4, it can be seen that there are some things in which network communication occurs in a process, in which the process has a large appearance feature. In particular, it is highly likely that the process occurrence feature is suspicious when $F \leq 0.05$, for which one must be careful.

4.2 Communication user frequency

$$hf_l = \frac{|\{h: h \ni N_l\}|}{|\mathcal{U}|}$$

Here, $|\{h: h \ni N_l\}|$ indicates the number of users who perform the same communication as communication information N_l, and $|\mathcal{U}|$ indicates the number of active users. Therefore, for example, if three users ($|\{h: h \ni N_l\}| = 3$) out of 10 users ($|\mathcal{U}| = 10$) send HTTP (80/100) to the destination IP address 203.0.113.13 TCP), the communication user frequency becomes $hf_l = \frac{3}{10} = 0.3$.

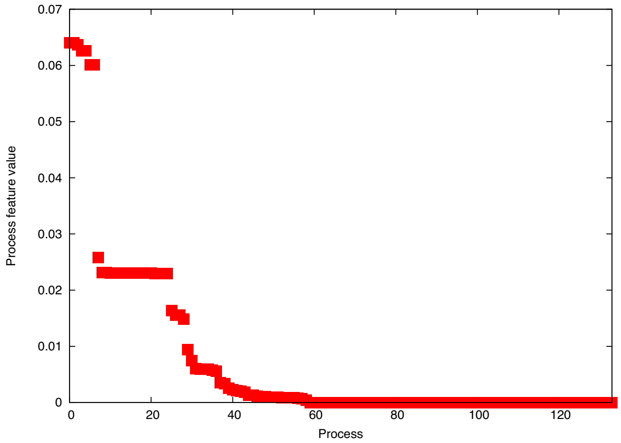

Fig. 3 Occurrence features distribution

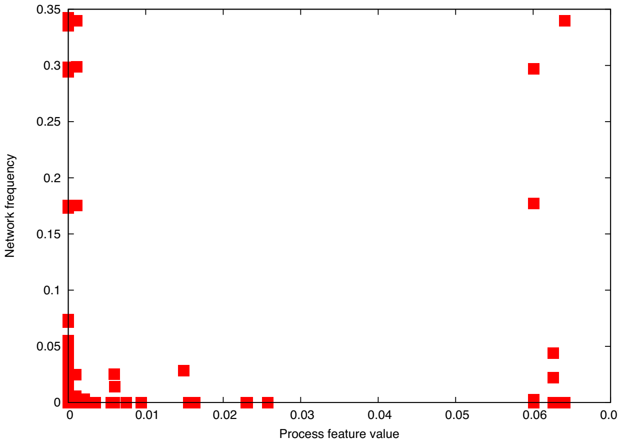

Fig. 4 Occurrence features and communication frequency distributions of processes

Figure 5 shows the distribution of occurrence features and communication frequency of processes that a user used during 1 hour from 9:00 to 10:00 for the 14 day period before December 1, 2015.

As shown in Fig. 5, even when the process has a large occurrence feature, there are some cases where the same communication as the process is generated by many users. In other words, even in a process used by a specific user, we see that communications generated are similar to communications of processes used by other users. In particular, when many users are communicating, risks may be low because the communication destination (or communication information such as the standby port) has a high possibility of being a general access destination. On the other hand, when the process has a large occurrence feature but few users are performing the same communication, there is a high possibility of illicit communication such as C&C communication. For this reason, it is important to find communication processes which many users are not utilizing.

4.3 Network occurrence feature

First, as in Section **3.3**, we define the occurrence feature of the network. When a malicious process is running, the occurrence of network access means the occurrence of a serious incident such as expansion of attack, further malware download, and information outflow to outside. Therefore, it is necessary to detect suspicious communication as soon as possible and take countermeasures. Thus the occurrence feature $NF_{i,l}$ of communication $N_{i,l}$ performed by the user $\mathcal{U}_i$ is defined as

$$inf_{i,l} = \log_2 \frac{1}{nf_{i,l}} \tag{5}$$

$$ihf_l = \log_2 \frac{1}{hf_l} \tag{6}$$

$$NF_{i,l} = inf_{i,l} \times ihf_l \tag{7}$$

Equation (5) becomes larger as the number of communications with the same communication information become fewer. As in Equation (2), Equation (6) becomes larger as the number of users performing the same communication becomes smaller. Therefore, $NF_{i,l}$ is large when a specific user generates a communication for the first time.

5 Suspicious degree of process

Here, suspicious degree of the process is defined from the process occurrence feature and network occurrence feature. Based on information of the incident that occurred at Japan Pension Service, we assess validity of this suspicious degree.

5.1 Suspicious degree of process

Process suspicious degree $S_{i,j}$ is defined from the process occurrence feature and network occurrence feature $(F_{i,j}, NF_{i,l})$.

$$S_{i,j} = \mathrm{MAX}\big(F_{i.j} \times NF_{i,l}\big)$$

There are cases where each process may be generating multiple communications, so the suspicious degree is calculated for each communication destination, and the maximum value is set as the suspicious degree of the process. Also, in the case of a process not involving communication, $S_{i,j} = F_{i.j}$ sets the suspicious degree equal to the occurrence feature of the process.

Figure 6 shows the distribution of suspicious degree of

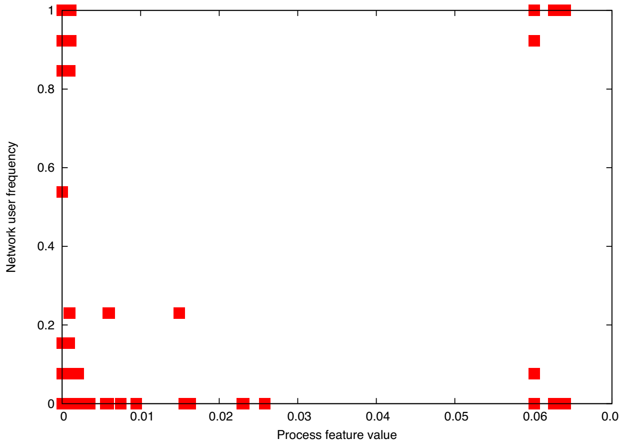

Fig. 5 Occurrence features and communication usage frequency distributions of processes

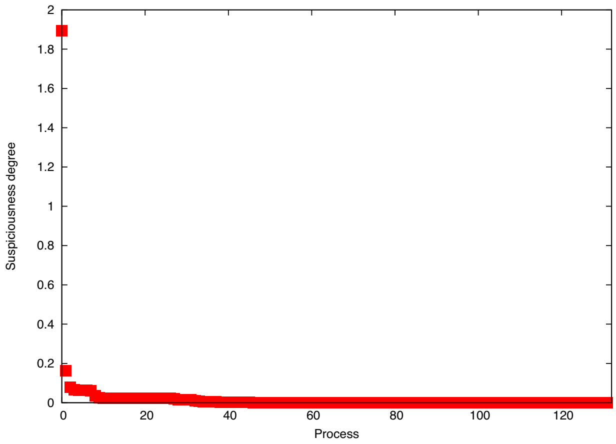

Fig. 6 Process suspicious degree

processes that a user used for 1 hour from 9:00 to 10:00 for the 14 day period before December 1, 2015.

From Figure 6, we see that by considering the network state, the suspicious degree is low even in a process having a high occurrence feature. That is, we see there are processes in which many users that communicate for access even when the occurrence feature of the process is high, and processes which many users run even with a high occurrence feature of communication.

On the other hand, we found there is a process with approximately 1.9 suspicious degree. This process has very few users (only 4 of the 13 users), and the occurrence feature of the process was also high at approximately 0.06. Also, among the 367 processes involving communications, only one process had the same communication information, and there was only one user, so the occurrence feature of communication is very high at 31.5. In fact, it is a process that generates a UDP socket, which was caused by changing the assigned port number (sender port number) for each socket generation. As a result, it was found that the occurrence feature of communication is very high, but it is not an especially illicit process.

5.2 Evaluation

Here, we select and evaluate suspicious processes based on information of the targeted attack that occurred at Japan Pension Service. Targeted attacks were made four times against the Japan Pension Service, and it was reported that three attacks were successful [3]. One terminal in the first attack on May 8, 2015, three terminals in the second attack on May 18, and one terminal in the fourth attack on May 20 and the infection spread to more than 20 terminals in the 2 days thereafter.

Therefore, we simulate the suspicious degree of the process when each attack occurs. As of April 1, 2015 at the Japan Pension Service, there were 12,000 regular employees and other staff, so we assume 12,000 terminals were being used [18]. In addition, three simulations were performed: 1) Attack 1: When first attack succeeded, 2) Attack 2: When second attack succeeded, 3) Attack 3: When the infection spread to 20 terminals after the successful 4th attack (2 days after infection). For Attack 1 and Attack 2, we set the usage frequency at one day to assume the time in case of infection, and the frequency of users is one person for attack 1 and three people for attack 2. Attack 3 took 2 days for the infection to spread, so the frequency of use is 2 days and the frequency of users is 20. Figure 7 shows the suspicious degree of the process in each attack.

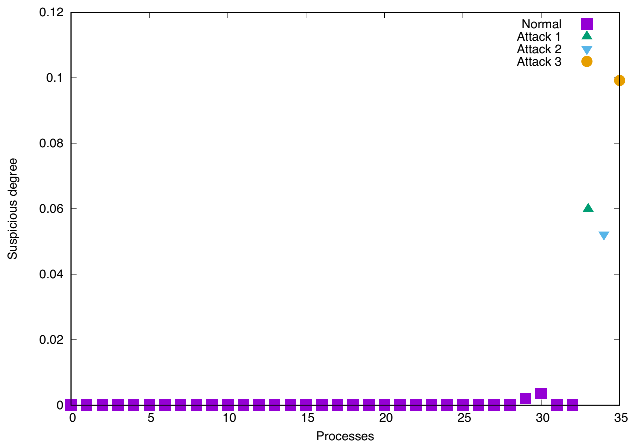

Fig. 7 Process suspicious degree by simulation

Figure 7 shows a comparison of the suspicious degree described in Section **5.1**, of processes run at a specific type by users at a specific time (Normal in Fig. 7). From Fig. 7, we see that the suspicious degrees of the processes used for the attacks are much greater than the suspicious degrees of other processes. Especially, when attacks succeeded like Attacks 1 and 2 (when usage frequency was low), we also found large suspicious degrees. Accordingly, we see that it is very useful for detecting suspicious processes. Also, Attack 3 had a process suspicious degree that was higher than the other two attacks. This may be because it has larger usage frequency, user frequency, etc., so that the suspicious degree of the process also increased. Furthermore, the suspicious degree of a process changes to higher values as infection spreads, so we can expect that it can be detected quickly as a suspicious process.

6 Conclusion

This paper proposed a method of periodically obtaining information on processes (list of processes) running in the terminal of each user, and determining whether these are processes that users normally use, by using the suspicious degree of the process. It was possible to identify an illicit process that pretends to be a normal process by simultaneously comparing the process name and the place where the process was executed (execution path).

Frequently used processes are used by many users, and the number of days the processes run (usage frequency) are also multiple days, so we defined the occurrence feature $F_{i,j}$ of processes we can find with few users and many usage days. In addition, as a result of comparing the occurrence features vs. communication frequency of processes, we found there are processes with large occurrence features and large communication frequency. Furthermore, as a result of comparing the usage ratios of processes involving the same communication, we found that some of the communications were only generated by a limited number of users.

Based on the above results, we defined the communication occurrence feature $NF_{i.l}$ of processes that can be found with small communication frequency and few users, and the suspicious degree of the process was determined from each occurrence feature. We showed that many processes that operated in a one hour period are actually processes that are not suspicious. On the other hand, after detecting one suspicious process and analyzing it in detail, we concluded that it was not an illicit process. Moreover, as a result of simulation based on the report of the personal information leak case in the Japan Pension Service, we demonstrated that suspicious degrees differed greatly between processes normally used and suspicious processes, and the proposed method is sufficiently useful.

In the future, when a suspicious process is discovered, it is necessary to place it under further monitoring such as obtaining detailed information (API history etc.) about the process, enabling one to catch illicit operations (suspicious network connections, etc.) early, and take action quickly. In judging suspicious processes, by obtaining detailed information such as API history and communications information also, it becomes possible to identify illicit processes with higher accuracy, and it is expected that countermeasures can be taken more quickly and surely.

Acknowledgment

We are deeply grateful for the great cooperation of everyone in the Information System Group for installing and managing agent tools on support terminals, for performing this research.

References

1 Press Release of The Japan Aerospace Exploration Agency (JAXA), “Investigation Result of JAXA Computer Virus Infection Incident,” http://global.jaxa.jp/press/2013/02/20130219_security_e.html

2 Information-Technology Promotion Agency, JAPAN(IPA), “System design guide against measures of ‘Advanced Targeted Attacks,’” https://www.ipa.go.jp/security/vuln/newattack.html (in Japanese)

3 National center of Incident readiness and Strategy for Cybersecurity, “The findings on cause investigation of incidents that is personal information leaks in the Japan pension service,” http://www.nisc.go.jp/active/kihon/pdf/incident_report.pdf (in Japanese)

4 Japan Computer Emergency Response Team Coordination Center (JPCERT/CC), http://www.jpcert.or.jp/english/

5 Information-Technology Promotion Agency, JAPAN (IPA), http://www.ipa.go.jp/index-e.html

6 JPCERT/CC, “About Targeted Attacks,” https://www.jpcert.or.jp/research/2007/targeted_attack.pdf (in Japanese)

7 JPCERT/CC, “Investigative report of countermeasure against targeted attacks,” http://www.jpcert.or.jp/research/2008/inoculation_200808.pdf (in Japanese)

8 IPA, “Case analysis and measure report of targeted cyber-attacks,” https://www.ipa.go.jp/security/fy23/reports/measures/index.html (in Japanese)

9 Yoshinori Katayama, Takeaki Terada, and Hiroshi Tsuda, “Behavior-based anti-spoofing technology for targeted cyber-attack,” The 28th Annual Conference of the Japanese Society for Artificial Intelligence, 4G1-4, 2014. (in Japanese)

10 Shigeki Kitazawa, Tomonori Negi, Kiyoto Kawauchi, Hiroyuki Sakakibara, and Seiji Fujii, “Evaluations of A Targeted Attack Detection System,” anti Malware engineering WorkShop (MWS 2009), A6-3, 2009. (in Japanese)

11 Yuki Unno, Masanobu Morinaga, Masahiro Yamada, and Satoru Torii, “Proposal for a method for detecting the intelligence activity of targeted cyber-attack in the internal system,” Computer Security Symposium (CSS 2012), pp.360–367, 2012. (in Japanese)

12 Satoru Torii, Masanobu Morinaga, Takashi Yoshioka, Takeaki Terada, and Yuki Unno, “Multi-layered Defense against Advanced Persistent Threats (APT),” FUJITSU Sci. Tech. J., vol.50, no.1, pp.52–59, 2014.

13 Jun-ichi Takeuchi and Kenji Yamanishi, “A Unifying Framework for Detecting Outliers and Change Points from Time Series,” IEEE Transactions on Knowledge and Data Engineering, vol.18, Issue 4, pp.482–492, 206.

14 Junji Nakazato, Yu Tsuda, Yaichiro Takagi, Masashi Eto, Daisuke Inoue, and Koji Nakao, “A Countermeasure for Targeted Attacks Using Host Based IDS,” Computer Security Symposium (CSS 2014), 2B2-3, 2014. (in Japanese)

15 Junji Nakazato, Yu Tsuda, Yaichiro Takagi, Masashi Eto, Daisuke Inoue, and Koji Nakao, “A Suspicious Processes Detection Scheme using Host Based IDS,” The 32nd Symposium on Cryptography and Information Security (SCIS 2015), 2A1-5, 2015. (in Japanese)

16 Junji Nakazato, Yu Tsuda, Masashi Eto, Daisuke Inoue, and Koji Nakao, “A Suspicious Processes Detection Scheme using Process Frequency,” IEICE technical report, vol.115, no.334, pp.61–66, 2015. (in Japanese)

17 Junji Nakazato, Yu Tsuda, Masashi Eto, Daisuke Inoue, and Koji Nakao, “A Suspicious Processes Detection Scheme using Process Frequency and Network State,” IEICE technical report, vol.115, no.488, pp.77–82, 2015. (in Japanese)

18 Japan Pension Service, “Japan Pension Service and its Operation,” http://www.nenkin.go.jp/files/about_jps_operation.pdf

Junji NAKAZATO, Ph.D.

Researcher, Cybersecurity Laboratory,
Cybersecurity Research Institute
Cybersecurity

Yu TSUDA, Ph.D.

Researcher, Cybersecurity Laboratory,
Cybersecurity Research Institute
Cybersecurity, Countermeasure against APT

Yaichiro TAKAGI

Technical Researcher, Cybersecurity Laboratory, Cybersecurity Research Institute

Cybersecurity