Cyber Attack Stage Tracing System based on Attack Scenario Comparison

Masahito Kumazaki[1], Hirokazu Hasegawa[2], Yukiko Yamaguchi[3], Hajime Shimada[3] and Hiroki Takakura[4]

[1] *Graduate School of Informatics, Nagoya University, Nagoya, Japan*

[2] *Information Security Office, Nagoya University, Nagoya, Japan*

[2] *Information Technology Center, Nagoya University, Nagoya, Japan*

[3] *Center for Cybersecurity Research and Development, National Institute of Informatics, Tokyo, Japan*

Keywords: Targeted Cyber Attack, Cyber Attack Scenario, Attack Stage.

Abstract: In the current organizational network consisting of multiple branch sites, there is a difference in security between sites, making it difficult to protect against targeted attacks. Therefore, it is important to detect and respond to attacks early, but it is also difficult to achieve this with the current network management. In order to solve this problem, we previously proposed a response support system for multiple sites. This system has two functions. First, it provides recommendations for an incident response by using information of incidents similar to the one. Second function estimates correlations among incidents and targets of cyber attack. To enable recommendations, we also proposed a method for evaluating the similarity of incidents and conducted experiments to investigate its effectiveness. We were able to correctly estimate the similarity of attacks when their attack stages were the same, but not when they were different. The result indicates the necessity to conduct similarity estimation for the same stage of attacks even if their current stages differ. By investigating stage transitions of attacks, we have to make alignment among their stages. In this paper, we propose a method to expect the attack methods and a system to generate information divided by attack stages. We also confirmed the effectiveness of proposed method by conducting experiments using a simulated cyber attack.

1 INTRODUCTION

The rapid increase in targeted attacks, which cause serious damage, has caused social issues in today's society. A targeted attack is a cyber attack targeting a specific individual or organization. It is difficult to protect an organization from this attack because it is specialized for the target using methods such as preliminary investigation.

Due to the internationalization of companies and the development of communication technology, an organizational network often becomes large, consisting of multiple branch sites around the globe. While such a network has advantages such as work efficiency, one of the crucial disadvantages is the difficulty in maintaining the security of all sites at a satisfactory level.

For the reason above, some targeted attacks have intruded from overseas sites with weak security, penetrated into the organization network with lateral movement and finally stole confidential information from the head office. Although quick response plays a key role in mitigating the damage caused by such attacks, the conventional network management in organizations cannot afford to take such response due to the following two reasons.

First, an incident response heavily relies on the administrator's skills at each site. As a general security manager, a security engineers who is assigned to a head office manages the security level of the entire organization network. Due to the budget limitation, however, only network engineers are usually assigned as site administrators in order to manage site's network including security. At the initial stage of an incident, not a general security manager but a site administrator usually recognizes its occurrence. In this case, the administrator needs to report the occurrence to the general manager and handle the incident under the instruction of the manager. Because of the different skill levels among site administrators, it is difficult to maintain quality on the correctness of the

Kumazaki, M., Hasegawa, H., Yamaguchi, Y., Shimada, H. and Takakura, H.
Cyber Attack Stage Tracing System based on Attack Scenario Comparison.
DOI: 10.5220/0010918200003120
In *Proceedings of the 8th International Conference on Information Systems Security and Privacy (ICISSP 2022)*, pages 587-594
ISBN: 978-989-758-553-1; ISSN: 2184-4356

report and the effectiveness of the incident response.

Second, it is difficult to make a correlation among incidents and use the knowledge of these incidents. At the stage of the lateral movement, several incidents occur simultaneously at many sites. The general security manager collects reports from the sites, investigates the correlation among the incidents and makes a decision on whether a critical cyber attack has occurred or not. However, the lack of uniform quality among reports, large distance, and time-zone difference among sites make the manager's work difficult.

To solve these problems, we previously proposed an incident response support system for multi-site networks(M. Kumazaki, H. Hasegawa, Y. Yamaguchi, H. Shimada, and H. Takakura, 2021a). In this system, both the security manager and the site administrator can obtain countermeasure recommendations against the incident, an estimation of the correlation of incidents across sites, and an identification of the attacker's objective, e.g., target device. To enable recommendation of incident countermeasures, we also proposed a method of evaluating the similarity of incidents that occur at multiple sites(M. Kumazaki, H. Hasegawa, Y. Yamaguchi, H. Shimada, and H. Takakura, 2021b). In this previous study, we compared similar cyber attacks regardless of their progress and evaluated their similarity. The result indicated the limitation of the conventional system. When different stages of the incidents are observed at the affected sites, this system did not correctly evaluate the correlation of incidents among the sites.

We found that we could evaluate the similarity among incidents with more precision if we divide the information about a series of attacks into each attack stage and compare the part of the same stages of the incidents. In this paper, we propose an attack method expectation method for estimating the attack methods and their execution times on the basis of the logs caused by a cyber attack. This method uses a table that summarizes attack methods, the logs caused by those methods, and the importance of those logs. It also compares this table with logs caused by a cyber attack and calculates the probability of each attack method.

We also propose a cyber attack stage tracing system to extract logs caused by a cyber attack and divide these logs into attack stages using the proposed method. The system collects logs for a certain period and extracts those caused by the cyber attack. The system expects used attack methods from the logs by the proposed method, and identifies the attack stage of the attack. We examined the effectiveness of the proposed method by conducting experiments using a simulated cyber attack.

2 RELATED WORK

Our conventional system uses communication behavior and various logs as evaluation indicators of incident similarity. These have also been used for detection methods of cyber attacks, and many studies have been conducted on them. As for attack detection using communication behavior, a method for detecting attacks using HTTP requests has been proposed(Y. Kanemoto, K. Aoki, M. Iwamura, J. Miyoshi, D. Kotani, H. Takakura, and Y. Okabe, 2019). This method extracts the attack code from the HTTP request, and executes it in the sandbox. Finally, it compares the execution result and actual HTTP response to determine the success or failure of the attack. As a system to detect cyber attacks using various logs, there is Security Information and Event Management (SIEM). This system not only centralizes the management of various logs, but also enables early detection of incidents by correlating and analyzing them. To enable detection of a wider range of cyber attacks, there are methods of extending the capabilities of SIEM(I. Kotenko and A. Chechulin, 2012; B. D. Bryant and H. Saiedian, 2017). These methods can detect cyber attacks, but depend on the skills of the administrators in terms of response. As mentioned in Section 1, attack detection alone is insufficient because of the variation in skills among site administrators, and it is desirable to provide response support as well.

There have been several systems proposed focused on information sharing within an organization, such as our conventional system (M. Colajanni, D. Gozzi, and M.Marchetti, 2008; C. Wagner, A. Dulaunoy, G. Wagener, and A. Iklody, 2016). The purpose of these systems is to share threat information within an organization. However, they share the malware information collected by honey pots or threat information collected by the administrators themselves. If administrators lack skills, they may not be able to effectively use the shared information well or collect information to register in the system. Our conventional system solves these problems by collecting threat information itself and sharing countermeasure recommendations instead of threat information.

Our proposed system focuses on the attack stages of a cyber attack to evaluate the similarity of incidents more accurately. Our conventional system uses this similarity to make recommendations to site administrators and enables early resolution of incidents. Many studies have been conducted to mitigate damage and resolve incidents early by focusing on the attack stages. Pivarníková et al. propose a method for detecting cyber attacks in their early stage and predicting how the attacks proceed by using Bayesian network algorithms(M. Pivarníková, P. Sokol, and T. Bajtoš, 2020). This method collects alerts of an intrusion detection system and predicts the correlation among these alerts. It builds a Bayesian network based on the results of the prediction and identifies the alerts caused by a certain cyber attack. In this way, the method enables detection of the early stage of an attack and prediction of subsequent attack activities. Saulius et al. proposed a method of detecting cyber attacks in the early stage by using a set of 31 logical filters(S. Japertas, and T. Baksys, 2018). This method is based on the Cyber Kill Chain(E. M. Hutchins, M. J. Cloppert, R. M. Amin, and others, 2011), and inserts an avoidance of the attack between the 4th stage (Exploitation) and the 5th stage (Installation) to mitigate the damage of the attack. This method filters the communications that occurred before the 4th stage of the kill chain by using multiple logical filters and detects attacks from the filtering results. These methods detect and predict attacks, but do not proposed countermeasures. As we described in Section 1, the different skill levels among site administrators makes it difficult to maintain the effectiveness of the incident response. For early response to attacks, it is necessary to provide extensive support that includes collecting incident information and proposing countermeasures.

3 OUR CONVENTIONAL SYSTEM

We have proposed an incident response support system for multi-located networks. To manage an organization's network, a site administrator is assigned to each branch site, and a general security manager is assigned to the head office. We also assume that the log server at the head office centrally manages the logs of all devices in the organization.

The system provides recommendations for an incident response by using information of similar incidents. When an incident occurs and the site administrator registers the incident data (shown in Table 1) to the system, the system searches for similar incidents among the stored information. If the most similar incident is found, the system will share its recommendation (e.g., resolution methods and response status) to the site administrator who registered the information of the incident.

The incident data consist of the items shown in Table1. "Incident ID", "Sub system ID", "Progress status" are given by the system. The site administrator registers "Device IP", "Occurrence time", and "Suspicious information". The system searches for similar incidents in the organization's incident database by using this information. If the system finds similar incidents, it shares its "Countermeasures and results".

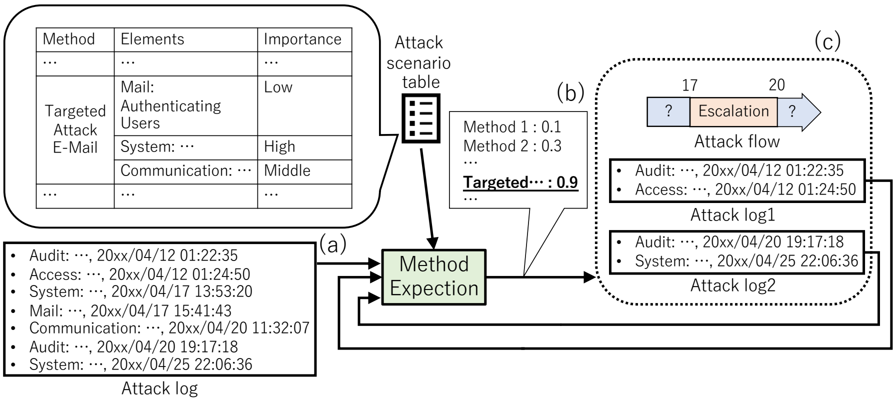
Table 1: Contents of incident data.

Item (type)	Example
Incident ID (int)	153
Sub system ID (int)	5
Device IP (str)	www.xxx.yyy.zzz
Occurrence time (date)	2020/05/10 18:10:55
Suspicious Information (list)	[Communication with abc.def.fed.cba, Vulnerable application A]
Similar Incident IDs (int)	[110,86]
Countermeasures and results (list)	[[Stop Application A, Unexecuted]]
Progress status (int)	1

To verify the effectiveness of the Current Incident Similarity Estimator, one of modules in our conventional system, in evaluating the similarity among ongoing incidents, we conducted an experiment using simulated cyber attacks(M. Kumazaki, H. Hasegawa, Y. Yamaguchi, H. Shimada, and H. Takakura, 2021b). We executed multiple simulated attacks with different attack stages and captured their communications in the experimental environment. We estimated the similarity of each attack using these communications as suspicious information. As a result, we were able to correctly estimate the similarity of attacks when their attack stages were the same, but not when they were different.

4 PROPOSED SYSTEM

The previous results indicate the necessity to execute similarity estimation for the same stage of attacks even if their current stages differ. By investigating the stage transitions of attacks, we have to make align their stages. In this paper, we propose a method to expect the attack methods and their execution times. We assume that expected attack methods are used for identifying attack stages, and their execution times are used to more accurately evaluate the similarity among incidents. In addition, we propose a cyber attack stage tracing system. The system collects information related to an incident and divides it into each attack stage. We assume that this system will be used as a sub-system of the Current Incident Similarity Estimator of our conventional system.



Method	Elements	Importance
···	···	···
Targeted Attack E-Mail	Mail: Authenticating Users	Low
	System: ···	High
	Communication: ···	Middle
···	···	···

- Audit: ···, 20xx/04/12 01:22:35
- Access: ···, 20xx/04/12 01:24:50
- System: ···, 20xx/04/17 13:53:20
- Mail: ···, 20xx/04/17 15:41:43
- Communication: ···, 20xx/04/20 11:32:07
- Audit: ···, 20xx/04/20 19:17:18
- System: ···, 20xx/04/25 22:06:36

Method 1 : 0.1
Method 2 : 0.3
···
Targeted··· : 0.9
···

- Audit: ···, 20xx/04/12 01:22:35
- Access: ···, 20xx/04/12 01:24:50

- Audit: ···, 20xx/04/20 19:17:18
- System: ···, 20xx/04/25 22:06:36

Figure 1: Flow of attack-method expectation.

4.1 Attack-method Expectation

To describe the proposed expectation method, we define two terms: Attack log and Attack scenario table. When a cyber attack occurs, various types of logs are caused depending on the attack method (e.g., terminal system logs indicating execution of a suspicious application and file server logs indicating access to unauthorized files). We define an Attack log as a set of this information recorded in multiple logs and arranged in a chronological order.

The Attack scenario table is a table of methods used by cyber attacks and information about them. The attack methods used for expectation are defined by general cyber attack frameworks such as MITRE ATT&CK(B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, 2018). For each method, the table summarizes elements caused by that method (e.g., system log) and the importance of each element. We define importance having the following three levels, "low" for elements that are recorded even in normal business and those that are often seen in other attack methods, "high" for those that are specific to that method, and "middle" for those that are neither. It is assumed that a general security manager will prepare this table before the system goes live, and that he/she can change it as needed.

The proposed method uses the Attack log and the Attack scenario table to expect the attack methods used for a cyber attack and the execution time of these methods. To do this, this method repeats the following cycle (Figure 1).

1. The method checks whether the element of each attack method exists in an Attack log (Figure 1 - a), and calculates the possibility of occurrence using the importance of each element in the Attack scenario table. For each method, there are elements of that method in the Attack log. If there are them, it adds the probability of the method in accordance with the importance of the element.

   For the attack method with the highest probability, if its probability exceeds a certain threshold, the proposed method expects that the attack method was used when the element was confirmed (Figure 1 - b).
2. It creates an Attack flow and Remaining logs (Figure 1 - c). The Attack flow is a summary of the attack methods, their related logs, and their execution times. It checks whether the probability of the attack method with the highest probability exceeds a certain threshold. If the probability exceeds that threshold, it adds the information of the attack method to the Attack flow. The Remaining logs are the logs removed from the Attack log that are related to this attack method. It repeats this cycle using these Remaining logs. If the probability is below the threshold, it deletes the received logs.

   If there is more than one attack method with the highest probability that cannot occur in parallel, then the proposed method treats them disjointly and outputs a distinct Attack flow for each method.

If there are no remaining logs left in the cycle, it stops this cycle and outputs the Attack flow at this point. Therefore, we can obtain the information about attack methods and their execution time.

4.2 Cyber Attack Stage Tracing System

4.2.1 Outline

The proposed system collects information related to an incident and divides it into each attack stage. The system can be used as a sub-system of our conventional system's Current Incident Similarity Estimator.

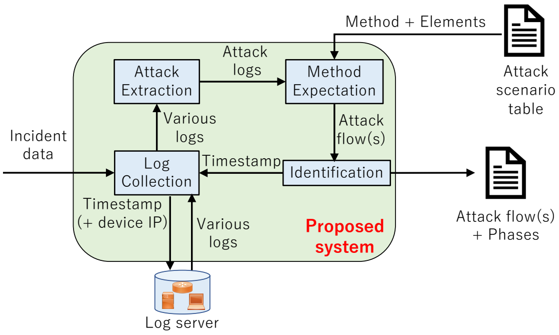


Figure 2: Proposed system's configuration.

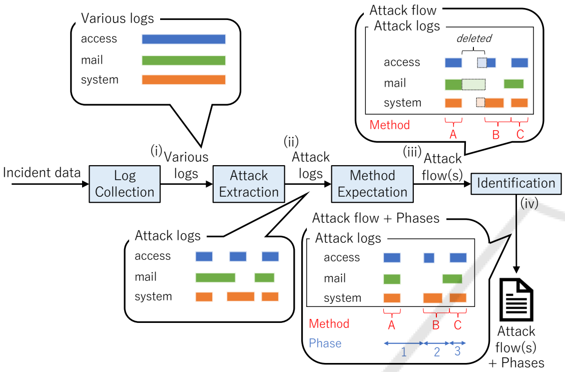


Figure 3: Proposed system's process flow.

The proposed system first receives the Incident data from the conventional system's Current Incident Manager. Next, it collects and narrows down the information necessary to identify the attack method. Finally, it estimates the attack method, execution time, and the attack stage. The system consists of four modules shown in the Figure 2 to achieve this. Figure 3 shows the process flow when information is input.

4.2.2 Log Collection Module

This module receives the Incident data from the Current Incident Manager. When the module receives such information, it accesses the log server and collects various logs including each terminal logs, e.g., communication logs, mail logs, terminal's system log, etc. As we described, the log server collects all logs recorded in the organization. This module searches and collects all server's logs and communication logs. If the Incident data includes the occurrence time, the module collects logs from that time to the current time. If not, it collects logs for a certain period of time predefined by the general security manager. If the Incident data includes the Device IP, which is the IP address of the device where the incident occurred, the module additionally collects logs about the device from system logs, audit logs, etc.

This module also receives a timestamp from the Identification module (explained in Sec.4.2.5). The module uses the timestamp instead of the occurrence time received from the Current Incident Manager, and collect logs from the timestamp to the current time.

Finally, this module sends the collected logs to the Attack Extraction module (Figure 3 - (i)).

4.2.3 Attack Extraction Module

This module extracts Attack log from various logs received from the Log Collection module. It removes logs of legitimate activities from those the Log Collection Module sends. We give examples of removing logs as follows.

- Mail log and communication log

  The module removes them based on the whitelist, which is created by using the results of threat intelligence on domain information obtained from DNS (Domain Name System) logs and email addresses.

- Terminal's system log and server's access log

  We assume that an ongoing investigation by the organization can create a whitelist of logs that occur routinely in these logs. This module removes logs on the basis of this white list.

It sends the remainder as the Attack log to the Method Expectation module (Figure 3 - (ii)).

4.2.4 Method Expectation Module

The Method Expectation module expects the methods used for an attack and their execution times from the Attack log sent from the Attack Extraction module. The details of the proposed expectation method were described in the Section 4.1. This module sends the flow created by this iteration to the Identification module (Figure 3 - (iii)).

4.2.5 Identification Module

The Identification module identifies the attack stage from the Attack flow received from the Method Expectation module. This module determines the attack stage and its period from the attack method recorded in the Attack flow and checks the first attack stage of the Attack flow. This module focuses on the following four stages defined by the Information-technology Promotion Agency, Japan (IPA) (Information-technology Promotion Agency, Japan, 2013): initial compromise stage, attacking infrastructure building stage, penetration/exploration stage, and mission execution stage, which may leave

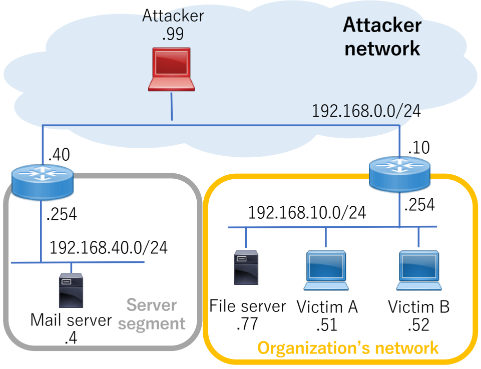


Figure 4: Experimental network.

records in the attack target organization. If the first attack stage is other than the initial compromise stage, it is possible that information about the attack exists in logs prior to this time. For additional investigation, this module sends the timestamp a certain time before the beginning of the Attack flow to the Log Collection module only once. This certain time is decided by the general security manager. The module then outputs the Attack flow and the attack stage together (Figure 3 - (iv)).

5 EXPERIMENT

To verify the effectiveness of the proposed expectation method for the attack method (Section 4.1), we executed a simulated cyber attack in an experimental environment, and created an Attack flow using the proposed method with the logs caused by this attack as the Attack log.

5.1 Environment

Figure 4 shows the experimental network. In this network, there were two LANs and one attacker terminal. In reality, an attacker attacks via the Internet, but in this experiment, we simulated the attack over the Internet with the configuration shown in Figure 6. The network with the IP address of 192.168.0.0/24 was assumed to be the Internet. One of LANs was assumed to be an organization's network and consisting of two employee terminals and a file server. These employee terminals were attacked by the Attacker in this experiment, so we call the employee terminal with the IP address of 192.168.10.51 as Victim A and the other terminal as Victim B. The file server has an employee directory that can be accessed by both employees. The other LAN is a server segment for the organization, and there is one mail server in this network that has employee mail accounts, and employees sign in to the mail server with their own terminals.

In this network, each terminal or server recorded the system log and authentication log of the employee terminal, the access log of the file server, and authentication log and sending/receiving log of the mail server. The routers also recorded the communication between the LAN and outside.

5.2 Experimental Operation

Table 2 shows the series of attack activities executed in the experiment. In the initial compromise stage, a targeted attack email with a Remote Access Tool (RAT) was sent to the mail server, and the Victim A ran the RAT. The Attacker escalated the system privilege of the Victim A and placed the RAT on the file server. The Victim B downloaded the RAT from the file server and ran it. The Attacker escalated the system privilege of the Victim B and stole the confidential information.

To acquire the logs recorded by these attack activities, we collected system and audit logs of victims, server's logs, and communication logs of the router. We manually created an Attack log from the collected logs, executed the proposed method for the Attack log, and confirmed the results. Regarding the method of calculating the probability of occurrence, the percentage of the highest elements is 60%, that of the next highest elements is 30%, and that of the lowest elements is 10% of each attack method. If the element of attack method was in the Attack log, the score of that method was added.

5.3 Result and Consideration

The Attack log obtained from attack activities described in Section 5.2 is listed shown in Table 3. We expected the attack method and its execution time by using the proposed method (Figure 5). First, we calculated the possibility of occurrence of each attack method using the logs with log numbers 1 to 9 as the Attack log. As a result, the possibility of distribution of attack tools by RAT and privilege escalation by tools was 100%, which was the highest. The log numbers related to these attack methods were 3,4,8, and 9, and when they were removed, we obtain two Remaining logs, one with log numbers 1 and 2 and the other with log numbers 5, 6, and 7. From these Remaining logs, we again calculated the possibility of occurrence of each attack method. When the log numbers 1 and 2 were used as the Attack log, the possibility of internal infection/initial infiltration by sending an email with a link to a malicious site was the highest. When the

Table 2: Attack scenario in the experiment.

Attack stage	Method	Execution content
Initial compromise	Targeted attack mail	Distributing and run RAT in the Victim A
Attacking infrastructure building	Escalation of privilege	Escalating the system privilege of the Victim A
	Distribution tools	Distributing and execute a tool to obtain credentials
Penetration/exploration	Network search	Searching the file server
	Malware placement to file server	Placing RAT in the file server and run it on the Victim B
Attacking infrastructure building	Escalation of privilege	Escalating the system privilege of the Victim B
	Distribution tools	Distributing and execute a tool to obtain credentials
Mission execution	Upload information	Uploading the confidential information file in the Victim B

Table 3: Attack log in the experiment.

Log-ID	Content
1	Receiving mail
2	Sustainable communication between Attacker and Victim A
3	Download file with Victim A
4	Escalation of privilege in Victim A
5	Searching for virtual volumes on Victim A
6	Access to file server
7	Regular communication between Attacker and Victim B
8	Escalation of privilege in Victim B
9	Download file with Victim B

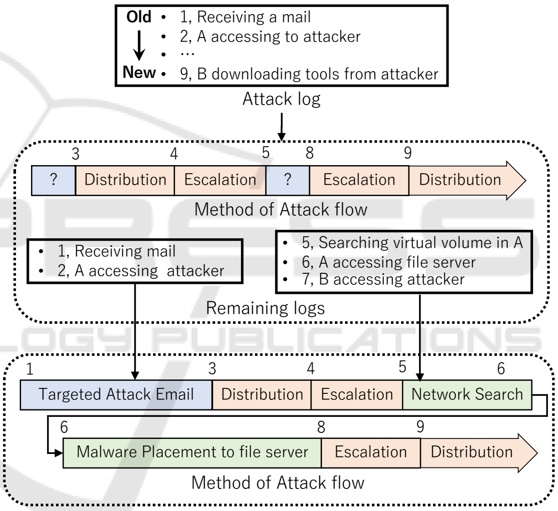


Figure 5: Expectation flow in the experiment.

log numbers 5, 6, and 7 were used as the Attack log, the possibility of the occurrence of the search of the shared folder and placement of malware in the shared folder became the highest. As a result, the Attack flow as shown in Figure 5 was obtained.

From the obtained Attack flow, the proposed method could expect execution time correctly when the Attack log did not contain any noisy data. However, there were parts where the attack methods were judged to be different from the actual methods. The experimental results determined that the first attack method in the Attack flow was a “Targeted attack email with a link to a malicious site”, but in fact it was a “Targeted attack email with a RAT”. This was due to the fact that there were fewer elements recorded than those in the Attack scenario table. For this reason, the attack methods with fewer elements in the Attack scenario table tended to have a higher probability of occurrence.

From the result, the proposed method needs to be improved in terms of expecting the attack method. However, it can expect the execution time of the attack method and can be used to identify the stage of the attack. To improve the method, we will conduct the same experiment with another attack scenario using a different attack method and check the recorded logs to create a more accurate Attack scenario table. We will also implement this method on logs with noisy data to determine the effect of such data.

6 CONCLUSION

We proposed a cyber attack stage tracing system and an attack method expectation method. The proposed system receives information from the site administrator about the terminal where the incident is occurring then collects information from the log server in the organization and extracts the logs related to the attack. After that, the system expects the attack method and execution time from the extracted logs by using the proposed method, and determines the stage of the attack.

We conducted an experiment to examine the effectiveness of the proposed method. We simulated an attack in an experimental environment and executed the proposed method using the obtained log as the Attack log. We found that the expectation of the attack method was wrong in some points, which needs to be improved. However, it was able to expect the execution time sufficiently. We will conduct similar experiments with attack scenarios using other attack methods. This will enable us to create a more accurate Attack scenario table. Since we did not confirm the effect of noisy data in the Attack log in this experiment, we will confirm this effect in future work.

We also need to study the details of how to extract the Attack log from various logs. We will study how to determine the stage of an attack from the Attack flow without relying too much on the expected attack method.

ACKNOWLEDGEMENTS

Part of this work is supported by the Cabinet Office (CAO), Cross-ministerial Strategic Innovation Promotion Program (SIP), “Cyber Physical Security for IoT Society” (funding agency: NEDO).

REFERENCES

B. D. Bryant and H. Saiedian (2017). A novel kill-chain framework for remote security log analysis with SIEM software, Journal of computers & security, Vol. 8, pp. 198–210.

B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas (2018). Mitre att&ck: Design and Philosophy, Technical report.

C. Wagner, A. Dulaunoy, G. Wagener, and A. Iklody (2016). Misp: The design and implementation of a collaborative threat intelligence sharing platform, Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, pp. 49–56.

E. M. Hutchins, M. J. Cloppert, R. M. Amin, and others (2011). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains, Leading Issues in Information Warfare & Security Research, Vol. 1, p. 80.

I. Kotenko and A. Chechulin (2012). Attack modeling and security evaluation in SIEM systems, International Transactions on Systems Science and Applications, Vol. 8, pp. 129–147.

Information-technology Promotion Agency, Japan (2013). System Design Guide for Thwarting Target Email Attacks. https://www.ipa.go.jp/files/000035723.pdf.

M. Colajanni, D. Gozzi, and M.Marchetti (2008). Collaborative architecture for malware detection and analysis, IFIP International Information Security Conference, Vol. 278, pp. 79–93.

M. Kumazaki, H. Hasegawa, Y. Yamaguchi, H. Shimada, and H. Takakura (2021a). Incident Response Support System for Multi-Located Network by Correlation Analysis of Individual Events.

M. Kumazaki, H. Hasegawa, Y. Yamaguchi, H. Shimada, and H. Takakura (2021b). Method of Similarity Evaluation among Incidents for Multi-Located Network (In Japanese), IEICE Technical Report, Vol. 120, No. 384, pp. 31–36.

M. Pivarníková, P. Sokol, and T. Bajtoš (2020). Early-Stage Detection of Cyber Attacks, Journal of Information, Vol. 11, pp. 560–581.

S. Japertas, and T. Baksys (2018). Method of early staged cyber attacks detection in IT and telecommunication networks, Journal of Elektronika ir Elektrotechnika, Vol. 24, pp. 68–77.

Y. Kanemoto, K. Aoki, M. Iwamura, J. Miyoshi, D. Kotani, H. Takakura, and Y. Okabe (2019). Detecting Successful Attacks from IDS Alerts Based On Emulation of Remote Shellcodes, Proceedings of the 43rd IEEE Computer Society Signature Conference on Computers, Software and Application, Vol. 2, pp. 471–476.