Development and Evaluation of Intelligent Network Forensic System LIFT Using Bayesian Network for Targeted Attack Detection and Prevention

Ryoichi Sasaki¹, Hiroshi Yamaki¹, Yoshio Kakizaki¹, Kazuki Shimazaki¹
Masato Terada² Tetsutaro Uehara³

1) Tokyo Denki University
5 Senju-Asahi-cho, Adachi-ku, Tokyo 120-8551, Japan
r.sasaki@mail.dendai.ac.jp

2) Hitachi Ltd., 292-0817 Yoshida-Cho, Totsuka-ku, Yokohama-shi, Kanagawaken 244-0817 Japan

3) Ritsumeikan University 1-1-1 Nojihigashi, kusatsu-shi, Shiga-ken 525-8577 Japan

ABSTRACT
Recently, the number of cyber attacks targeting companies or government departments has been increasing. Although such organizations are required to prepare countermeasures against targeted attacks, it is very difficult to implement these measures during an attack without the assistance of a support system. Therefore, the authors developed the Live and Intelligent Network Forensic Technologies (LIFT) system to guide the attack response using artificial intelligence techniques, such as a Bayesian network. This system analyzes collected logs, detects clues (signs) of attacks, then uses Bayesian networks to estimate the probability of an attack from the detected clues. If the certainty factor is large enough, an attack is assumed to be occurring, or else the LIFT system requires the collection of additional clues from the logs. Moreover, the LIFT system guides the implementation of countermeasures and/or conducts automatic operations with knowledge of the relation between the event and the proposed action, which would be a guide to the operator or an automatic operation. The authors developed a prototype of the LIFT system and applied this prototype to attack sequences that occurred in the past. As a result, it was confirmed that LIFT was able to detect the clue and event and recommended a countermeasure appropriately because the abnormal clue ended when the recommended countermeasure was conducted.

KEYWORDS
Keywords: Digital Forensics, Network Forensics, Targeted Attacks, Artificial Intelligence, Bayesian Network

1 INTRODUCTION
In recent years, the number of cyber attacks has been increasing, especially targeted attacks against organizations and individuals are a problem. Many companies and government departments have suffered damage from targeted attacks. Although such organizations are required to prepare countermeasures against targeted attacks, it is very difficult to implement these measures during an attack without the assistance of a support system.

The Security Information and Event Management (SIEM) system has been attracting attention as a support system for defending against targeted attacks [1]. The SIEM system provides real-time security threat detection capabilities to the log management system. Because it performs network forensics in real time, SIEM is called a live network forensics system. Network forensics are trained with past records to enable the analysis of logs and detection of attacks in real time.

However, it is difficult to protect against an attack or mitigate the effects of an attack by using only the SIEM system, because events that can be estimated by the SIEM system are limited, and current guides on countermeasures are insufficient.

Therefore, operators need enough knowledge and skill to use the system appropriately. Unfortunately, the number of security engineers with the necessary level of technical expertise is insufficient. Therefore, a guide function to grasp the situation of attack and to take necessary measures and a semiautomatic driving function are required to enable an emergency response.
and log conservation even in organizations that do not have highly skilled operators.

To solve such problems, the authors developed the Live and Intelligent Network Forensic Technologies (LIFT) system to guide operations and/or conduct operations automatically using Bayesian networks, which is an artificial intelligence technique. In more detail, the LIFT system collects the logs from servers, PCs and communication equipment such as routers, and detects abnormalities in the collected logs. Next, the LIFT system calculates the certainty factor of an event occurrence using Bayesian networks representing the relation between the abnormalities and the estimated event type. If the certainty factor is large enough, the event is assumed to have occurred, or else the LIFT system requires the collection of additional clues from the logs. Moreover, the LIFT system guides the operator in implementing the proper countermeasures and/or conducts the countermeasures automatically using knowledge of the relation between the event and the proposed action, which could be a guide or an automatic operation.

This paper describes the development of the LIFT system, and the presents the developed prototype for the LIFT system and the experimental results from applying the prototype to the data from an attack sequence that occurred in the past. From the experimental results, the authors could confirm that the LIFT system can be a useful tool to perform countermeasures during a targeted attack.

2 RELATED RESEARCH

Many studies on targeted attacks have been conducted. For example, some studies have considered detection methods for targeted attacks [2] [3] [4] and measures for defending against targeted attacks [5]. Also, other studies have used a combination of SIEM and artificial intelligence methods such as Support Vector Machines (SVM) [6] for anomaly detection [7], and the performance evaluation of a classifier [8], and so on.

Studies have also applied Bayesian networks to the vulnerability assessment of chemical facilities subjected to external attacks [9], and dynamic security risk management using Bayesian Attack Graphs [10]. However, there are no studies on intelligent systems dealing with targeted attacks which estimate events from clues using Bayesian networks. The authors also proposed a system for estimating events from clues about targeted attacks, but this estimation uses a rule-based system instead of Bayesian networks [11]. It is known that Bayesian networks have the advantage of increasing the accuracy of the posterior probability by continuing observations if the prior probability is not sufficiently accurate [12].

The authors engaged in the research on digital forensic as shown in [13][14]. This paper is based on the knowledge obtained from these researches.

3 LIFT PROJECT AND LIFT SYSTEM

3.1 LIFT Project

The LIFT project began at the Cyber Security Research Institute of Tokyo Denki University in 2013. In the project, the authors developed the LIFT system and related systems, such as the countermeasure planning assist system and the experiment support system, for acquiring knowledge for situational awareness, as shown in Figure 1. The planning assist system is used to determine the optimal action for a given situation. The experiment support system carries out penetration experiments and obtains the relation between an event and abnormal signs.

3.2 The Structure of an Attack and Terms Used in the LIFT System

Figure 2 shows the structure of an attack and the related terms used in the LIFT system.
Figure 2. Structure of an attack and terms used in the LIFT System.

- **Attack case**
  The “Attack Case” represents the flow of attacks that have occurred in the past.

- **Attack phase**
  The “Attack Phase” represents the current state of the attack. Here, the attack phase is determined based on the attack scenario described by the Information-technology Promotion Agency (IPA), Japan, as shown in Table I.

### Table 1. Attack Phase.

<table>
<thead>
<tr>
<th>Phase No.</th>
<th>Attack Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Trespass Phase</td>
</tr>
<tr>
<td>II</td>
<td>Attack Infrastructure Construction Phase</td>
</tr>
<tr>
<td>III</td>
<td>Invasion &amp; Survey Phase</td>
</tr>
<tr>
<td>IV</td>
<td>Purpose Execution Phase</td>
</tr>
</tbody>
</table>

- **Event (Attack Result)**
  The “Event” is the intended result of an attack. If the event was correctly identified, it is possible to estimate the attack phase, predict the attacker's actions and plan the emergency response.

- **Clue (Sign)**
  The “Clue”, which is also called a sign, is the detected result of an attack (event). An event generates several clues. The certainty factor for the relation of the event and the clues is given by observing abnormalities in the experiment. The event is estimated by using patterns in the clues and the certainty factor.

- **Source**
  The “Source” represents the event log that stores the clue origin.

### 3.3 Overview of the LIFT System

- An overview of the LIFT system is illustrated in Figure 3. The LIFT System consists of a core module and an extension.

The LIFT core module is used in emergencies during attacks. In the core module, a Bayesian network representing the relation between the clue and event, and an event and measure relation table are used. Here, the clues are collected from object networks using OSSEC [15], which is an open source program having functions such as log collection, log analysis, finding clues, etc. The Bayesian network representing the relation between the clue and event is used to estimate the current event, as described in more detail in Section 3.4. The event and measure relation table is used to guide the countermeasures that the operator should employ as described in Section 3.6.

- The extension of the LIFT system is used after an emergency to estimate the intrusion source of the targeted attack and the impacted area in the object network based on the log obtained by Onmitsu and the structure information of the network [16]. Here, Onmitsu is a process logging tool developed by Mimura et al. to record information about network volatility which is useful for identifying the sources of suspicious communications [17]. Onmitsu records the process state on the main memory and the executed process information in a sequential fashion. Onmitsu’s log can reduce the time taken to identify the malware and software used for attacks.

### 3.4 Bayesian Network

A Bayesian network is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph. In the field of artificial intelligence, Bayesian networks have been studied in probabilistic reasoning algorithms since around 1980, and there is a long history of research and practical use of these networks. For
example, a Bayesian network can be used to represent probabilistically the relationships between diseases and symptoms. Given some symptoms, the Bayesian network can be used to compute the probabilities of various diseases being present [18].

Figure 4 shows a simple example of a Bayesian network which is composed of three nodes, D1, D2 and T1 and two edges [19]. The nodes D1 and D2 are called the parents of node T1. Here, the D1 and D2 nodes represent the cause, and the T1 node represents the evidence. For example, D1 represents whether a person has disease i (i = 1, 2), and T1 indicates whether the inspection result is normal or abnormal. Specifically, when the person is suffering from disease i, D1 = 1, and when the inspection result is abnormal, T1 = 1.

Here, P(D1) is the probability of D1 and P(D2) is the probability of D2; both are given as shown in Figure 4. Here, P(D1 = 0) = 0.999, P(D1 = 1) = 0.001, P(D2 = 0) = 0.998 and P(D2 = 1) = 0.002, where, if D1 = 1 then D1 is true, and if D2 = 0, D2 is not true.

The conditional probability of P(T1|D1, D2) is also shown in Figure 4. For example, P(T1 = 1| D1 = 1, D2 = 1) represents the probability that T1 = 1 when D1 = 1 and D2 = 1. Here, P(T1 = 1| D1 = 1, D2 = 1) = 0.950. This equation represents that the probability of the inspection result being abnormal is 0.950 under the condition of the person having diseases 1 and 2.

In general, the following expression holds [19].

\[ P(X_1, X_2, \ldots, X_n) = P(X_1|\text{Pa}(X_1)) \cdot P(X_2|\text{Pa}(X_2)) \cdot \cdots \cdot P(X_n|\text{Pa}(X_n)) \]  

(1)

Here, Pa(Xi) represents the parents of Xi.

In Figure 4,

\[ \text{Pa}(D_1) = \{ \}, \text{Pa}(D_2) = \{ \}, \text{Pa}(T_1) = \{D_1, D_2\}. \]

(2)

Here, \{\} represents the empty set.

Therefore, from Bayes’ Theorem, the following equation can be obtained.

\[ P(D_1, D_2, T_1) = P(D_1)P(D_2)P(T_1|D_1, D_2) \]  

(3)

Here, the value of D1, when T1 = 1 is called the “belief” of D1 and can be expressed as

\[ \text{BEL}(D_1=1, T_1=1) = P(D_1=1|T_1=1) = \frac{P(T_1=1|D_1=1)P(D_1=1)}{P(T_1)}. \]  

(4)

Also, the probability that D1 and D2 occur simultaneously can be obtained by the following equation.

\[ \text{BEL}(D_1=1, D_2=1, T_1=1) = P(D_1=1, D_2=1|T_1=1) = P(T_1=1|D_2=1, D_2=1)P(D_1|P(D_2)/P(T_1)) \]  

(5)

In this case, the value of \text{BEL}(D_1=1, T_1=1) can be calculated as follows.

\[ \text{BEL}(D_1=1, T_1=1) = P(D_1=1|T_1=1) = \frac{P(D_1=1)P(T_1=1|D_1=1)}{P(T_1)} = \{P(D_1=1)P(D_2=1)P(T_1=1|D_1=1, D_2=1) + P(D_1=1)P(D_2=0)P(T_1=1|D_1=1, D_2=0)\}/P(T_1=1) \]  

(6)

Here,

\[ P(T_1=1) = P(T_1=1|D_1=1, D_2=1)P(D_1=1)P(D_2=1) + P(T_1=1|D_1=1, D_2=0)P(D_1=1)P(D_2=0) + P(T_1=1|D_1=0, D_2=1)P(D_1=0)P(D_2=1) + P(T_1=1|D_1=0, D_2=0)P(D_1=0)P(D_2=0). \]  

(7)

By using equations (6) and (7), and the values given in Figure 4, \text{BEL}(D_1=1, T_1=1) = 0.37.

(8)

This equation represents that if the inspection result is abnormal, then the probability of disease 1 is 0.37.

In the same way,

\[ P(D_2=1, T_1=1) = 0.23 \]  

(9)

\[ P(D_1=1, D_2=1, T_1=1) = 0.00076. \]  

(10)

In this way, by giving the evidence value when the cause is given as shown in Figure 4, it is possible to obtain the probability of occurrence of the cause when the evidence is given by the Bayesian network.

Figure 5 shows part of the Bayesian network used in the LIFT system. In the LIFT system, the cause is denoted “event”, and the evidence is denoted “clue”.

The occurrence probability of \text{P(Di)} for i = 1, 2, 3, 4 and the conditional probabilities of \text{P(Tj | Di )} for i = 1, 2, 3, 4 and j = 1, 2, 3, 4, 5 as well as
the graph structure are mainly determined from engineering expertise and the past data. Here, $D_i$ represents the $i$-th event candidate, When $T_i = 1$, the $i$-th clue was observed, and the value of the belief, $P(D_j | T_i)$, for $i = 1, 2, 3, 4$ and $j = 1, 2, 3, 4, 5$ can be calculated using the Bayesian network as described above.

At this time, the maximum value among these beliefs is called the certainty factor, and the $j$ which gives the maximum value is called the estimated event.

The LIFT system starts to estimate an event when clues are detected on some terminals due to an attack. The detected clues are input to the Bayesian network and the beliefs calculated for each event and the largest belief is used as the certainty factor as described in Section 3.4. If the certainty factor exceeds the reference value, it is assumed that the event is currently occurring. If there is no event above the reference value, an additional survey of the clues will be conducted. If additional surveying can be performed automatically, it will be done by itself. If it cannot be performed automatically, a guide will be displayed to the operator, and the survey result will be input by the operator. If a new clue is detected within a certain time, the certainty factor is again compared with the reference value. When no clues are detected within a certain period, or the certainty does not change or decreases after leaving the record, the estimation

### 3.5 Event Estimation

The flow of event estimation using a Bayesian network in the LIFT system is shown in Figure 6.
process is terminated as the first clue may have been a false alarm.

3.6 Selection of Measure

After the event estimation, the LIFT system selects countermeasures as illustrated in Figure 7. The selection of measures is carried out using the event and measure relation table. Table 2 shows an example of an event and measure relation table. The possible measures are set for each event in advance based on engineering expertise. The degree of recommendation is calculated based on the implementation cost of measures such as the time required and complexity, the scope of impact, and whether automatic execution is possible.

After selecting a measure, the LIFT system checks whether the selected measure can be executed at that time. This is because there is a possibility that the organization's work will be affected by the execution of measures. If execution is permitted, those that can be done automatically are executed, and those that cannot be done automatically are displayed in the guide to the operator and the input results. If the execution permission of the measure is not issued or if the execution of the measure is not successful, a recommendation for the next measure to take is presented. When a measure is successfully executed, a report showing the detected clue, the estimated event, the attack type and phase and so on are displayed.

4 LIFT PROGRAM

4.1 Overview of LIFT program

We developed a program to implement the LIFT core module and named it the LIFT program. This program is written in C# and consists of about 1000 steps.

The LIFT program is composed of three modules "liftbatch", "liftmain", and "liftweb". Figure 8 shows the configuration of the system.

1) Liftbatch

The Liftbatch module estimates whether an attack is occurring using a Bayesian network and the clue/log stored in a database obtained from a server or a PC using OSSEC. It also has functions for selecting responses and to save the results in a database. Furthermore, by acquiring clues and logs periodically from the database at short intervals and carrying out these processes, close to real-time estimation becomes possible.
2) Liftweb
The Liftweb module displays information received from the Liftmain module through an API and to write data entered by the operator to the database via Liftmain.

3) Liftmain
In order to interact with Liftweb, the Liftmain module converts data to the specified API format given by Liftweb.

4.2 LIFT program screen
4.2.1 Normal operation

The user interface of the developed LIFT program is shown in Figures. 9–12. The upper bar represents the state of the object system. In the normal state, it is green, and changes to yellow when clues are detected, and then to red when events are detected. In the TODO list, urgent messages from the LIFT program to the operator are displayed. Examples of messages include "estimate events" and "additional survey of clues" and so on. On the left part of the screen, various types of information such as clues and logs can be displayed in chronological order. On the screen shown in Figure 9, the display can be changed with a button on the TODO list, the clue list, the event list, the measure list, and the network diagram. For clues that cannot be detected from automatically collected logs, it is also possible to input clues manually by with the discovery button in the clue list.

![Figure 9. User interface during normal operation.](image)

4.2.2 Behavior during clue detection

This section describes the operating procedure for when a clue is detected. Figure 10 is a screenshot of when the LIFT program detected the clue "Access to domains registered in the blacklist". The message "access to domains registered in the blacklist" is also displayed in the clue list. In addition, as a clue is detected, and the certainty factor of an event has increased, the color of the upper bar changes to yellow. The clues are detected from the automatically collected logs, but operators can also detect clues that cannot be automatically collected, and then manually enter them. Moreover, by selecting “details” from the options, you can see the log of the detected object.

![Figure 10. User interface after a clue is detected.](image)

4.2.3 Behavior when estimating events

Figure 11 shows that the LIFT program detected multiple clues and an event was estimated. This Figure shows that the clue "Communication not via proxy" and the clue "Access to a domain registered in the blacklist" is detected, and the event "Malware communicates with the C&C server" is estimated. Because the event has been estimated, the color of the upper bar changes to red.

![Figure 11. User interface during event estimation.](image)
clues disappear before a certain period of time elapses. If the specified time elapses and the clues do not disappear, it is assumed that the measure was not effective and the program instructs the administrator to execute another measure.

In the experiment, during the communication with the C&C server, the clue "communication by User-Agent different from usual" was detected. In addition, when downloading the tool for the attack, the clue "Access to a URL included in the blacklist" was also detected. Then, the event "Download of necessary functions" was detected by the LIFT program. Moreover, "Network isolation of corresponding terminal" was proposed as the countermeasure. Communication with the C&C server was stopped by disabling the ethernet adapter of the corresponding PC or server.

Experimental results confirmed that there were no problems with the basic operation of the LIFT system. We would like to further test our system with different attacks in the future. Because the authors used a small network in this experiment, there was no problem with the log analysis. However, there is a possibility that it will become a problem for large networks. The authors think that it is possible to solve this issue by increasing the power of server used for analysis.

6 CONCLUSIONS AND FUTURE WORK

This paper provides an overview of the LIFT system, describes the development of a prototype system, and presents the experimental results from applying the prototype to an attack sequence that occurred in the past. From the experimental results, the authors could confirm that the LIFT system can be a useful tool for countering a targeted attack.

Although the LIFT core module has been developed, it is desirable to also develop the extensions of the LIFT program shown in Figure 3 in the future.

In addition, although the LIFT system can handle known attacks, it is not clear whether it can handle new types of attack. Therefore, the research team plans to develop new attacks and check whether the LIFT system can successfully counter them.

ACKNOWLEDGMENT

The authors would like to express the gratitude to the members of the LIFT project.
who gave us guidance and support for this research.

REFERENCES

### TABLE 2. Example of event and measure relation table.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Events</th>
<th>Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrusion</td>
<td>Mail with malware attached is received</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Employees start malware attached to e-mail</td>
<td>4  4  2  1  1</td>
</tr>
<tr>
<td>Construction of</td>
<td>The malware communicates with the C&amp;C server</td>
<td>3  4  3  4  2  1  1</td>
</tr>
<tr>
<td>Fundamentals</td>
<td>Download necessary functions</td>
<td>3  2  2  4  2  1  1</td>
</tr>
<tr>
<td>Internal penetration</td>
<td>Information in the terminal that became the attack base was obtained</td>
<td>3  2</td>
</tr>
<tr>
<td></td>
<td>An attacker searched the internal network from the attack infrastructure</td>
<td>3  3  4  2  3  1</td>
</tr>
<tr>
<td></td>
<td>An attacker infiltrates from attack infrastructure to another terminal</td>
<td>2</td>
</tr>
<tr>
<td>Objective</td>
<td>Destruction of data in the terminal</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Sending confidential information</td>
<td>2  3  3  2  2  4  3</td>
</tr>
</tbody>
</table>

Numbers represent the priority of measures. 4: high, 3: medium high, 2: medium low, 1: low.