Insider Threat Detection Using Decoy

Authors: *Takumi Yamamoto*, Kohei Nozawa**

**Information Technology R&D Center*

Abstract

Information leakage due to cyberattacks is often cited as a major issue, but in reality numerous information leakage incidents caused by insiders also occur[1]. An insider is a malicious user with legitimate access privileges, and it is difficult for existing security measures to properly detect such signs. We focused on decoys as a means of uncovering the malicious intent of insiders. We developed a method that dynamically places decoy files for users whose activities differ from usual and narrows down insiders based on access patterns to those decoy files.

1. Introduction

"Damage from information and other leakages due to malicious insiders" ranks fourth in IPA (Information-technology Promotion Agency, Japan)'s "Top 10 Information Security Threats 2025" and has been treated as a top ten threat for ten consecutive years[2]. In OT (Operational Technology) environments such as critical infrastructure and factories, malicious activity by insiders is also recognized as a serious threat[3]. It follows that measures against insiders need to be considered. This paper describes Mitsubishi Electric's efforts on technical measures against insiders.

2. Insider Threats

In this paper, an insider is defined as a malicious user who has a legitimate account. Given that traditional monitoring and defense models generally trust users once they have been authenticated, it is difficult to counter insiders using those models. Therefore, applying behavior analysis technologies such as anomaly detection and User and Entity Behavior Analytics (UEBA), which evaluate deviations from users' normal behavior—analyzed in advance—as a risk value and exposes high-risk users, is considered effective[4]. However, users' job duties vary widely, and even when the same user performs the same task, there are variations in users' behavior; therefore, it is difficult to calculate the risk value accurately. Insiders may also commit malicious activities within the range that behavior analysis technologies evaluate as low risk in order to avoid detection (for example, uploading confidential files in small quanities), and strict monitoring with behavior analysis technologies can lead to frequent false alarms.

For these reasons, accurately identifying insiders is considered to be a very difficult challenge. Given that they carry out malicious acts within the scope of legitimate access privileges, the difference between insiders and users without malicious intent (referred to as "legitimate users") is only the presence or absence of malicious intent, and merely passively observing behavior makes it difficult to distinguish them.

3. Proposed Insider Countermeasure Technology

In light of this, rather than separating insiders from legitimate users solely based on passively observable information, we adopted an approach in which the defending side actively induces actions that an insider would likely take, indirectly uncovering the insider's malicious intent and observing actions related to that intent. To indirectly uncover an insider's malicious intent, the concept of the proposed method is to dynamically place decoy files for users whose activities differ from usual—files likely to attract interest if the user is an insider—and to narrow down insiders based on access patterns to the decoy files.

An overview of the proposed method is as follows (Fig. 1).

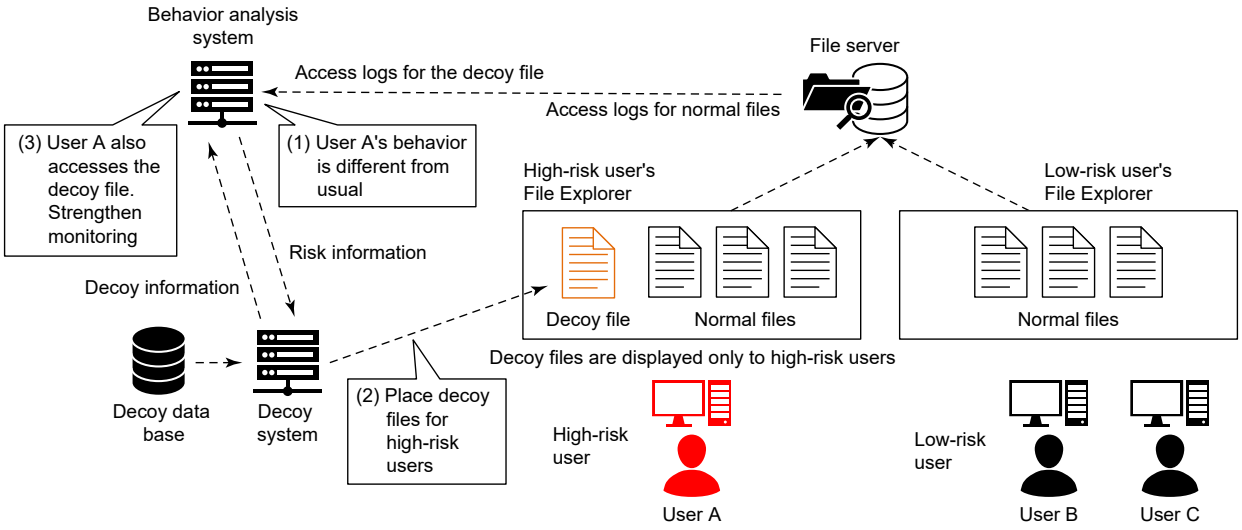


Fig. 1 Overview of the proposed method

(1) Using a behavior analysis system such as anomaly detection, analyze file access logs, etc., and calculate anomalies in behavior as a risk value. Furthermore, create risk information that includes the target user name, risk value, and file access information.

(2) The decoy system receives risk information from the behavior analysis system and, for high-risk users, infers from file access information the topics likely to attract the interest of the user, then selects decoy files containing content related to those topics from a decoy data base in which decoy files on various topics are stored. After that, control the display of the user's file explorer so that the selected decoy file appears as if it is placed on the file server. By placing decoy files that are appealing to insiders, the probability of selecting the decoy files increases, making insiders easier to identify. Information about the placed decoy files (file names, target user names, etc.) is passed to the behavior analysis system as decoy information.

(3) The behavior analysis system also monitors access to decoy files, just like normal files. The behavior analysis system determines whether there has been access to decoy files based on decoy information from the decoy system. If access to a decoy file occurs, this is fed back to the behavior analysis system's risk value (or threshold) to strengthen monitoring of higher-risk users who access decoy files in a proactive manner and narrow them down as potential insiders.

By limiting the placement of decoy files to users determined to be high risk, we can reduce disruption to the work of low-risk legitimate users performing their usual tasks. In the unlikely event that a legitimate user is judged high risk and decoy files are placed, it is expected that they will not actively access files other than those they usually use, so the likelihood of business disruption is low.

4. Prototype and Operational Image of the Proposed Approach

We created a prototype to verify the feasibility of the proposed approach. The configuration of the prototyped system is shown in Fig. 2.

Details are available in our existing research paper[5], however, Fig. 2 's decoy selection server, decoy distribution server, decoy displaying plugin, and decoy data base work together, and the functions corresponding to the decoy system in Fig. 1 are achieved. Given the recent increase in cloud-based file sharing, in this prototype Fig. 1 's file server was not used; instead, on SharePoint[6] we placed content that simulates files used for business. The behavior analysis system was built on Azure[7] using Elastic Stack[8]. The decoy distribution server was also built on Azure. The decoy selection server, which obtains reference information for decoy files from the decoy data base, was built in the business environment. The decoy data base was placed on the same SharePoint as the simulated business content. Assuming access to files on SharePoint via a browser, the control to make only target users able to view decoy files was implemented by installing a decoy displaying plugin in the browser.

4.1 Flow of operation of the prototype system

The following outlines the flow of operations in the system configuration (Fig. 2).

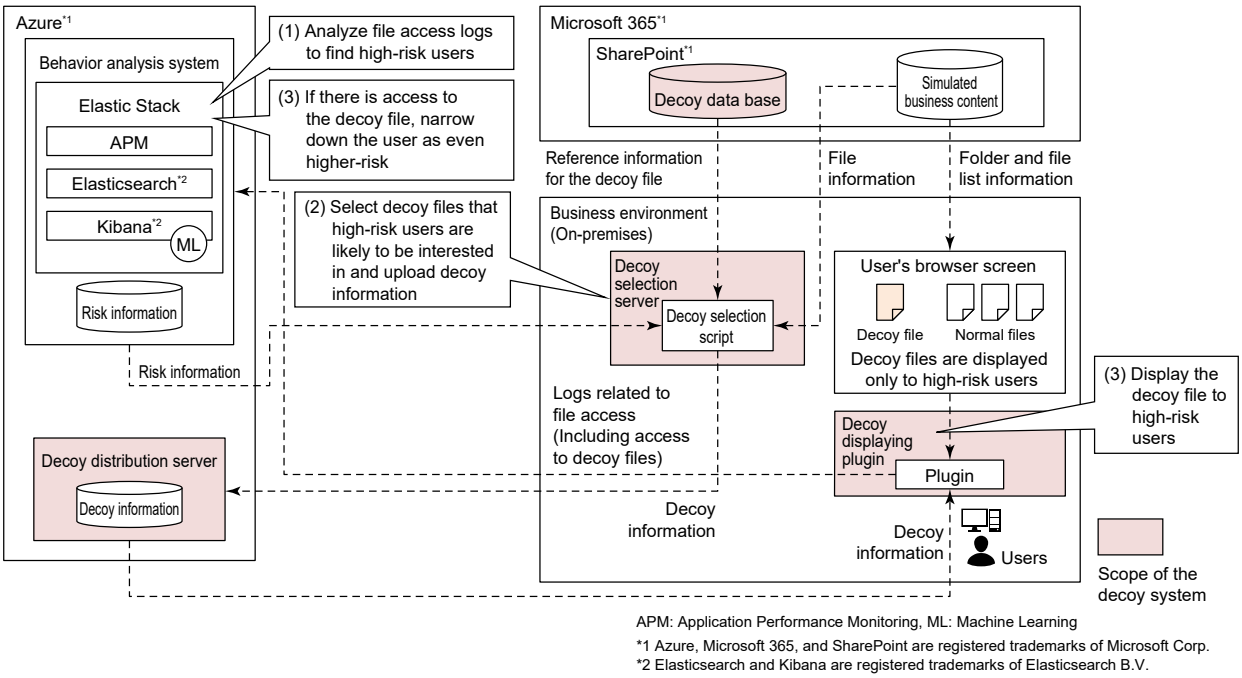


Fig. 2 Configuration of the prototype system

(1) The behavior analysis system analyzes logs related to file access on SharePoint and calculates a risk value. Furthermore, it creates risk information that includes the names of high-risk users, the risk value, and information on file accesses that were determined to be high risk.

(2) The decoy selection server periodically obtains the risk information from the behavior analysis system and estimates file topics from information about the files that high-risk users accessed most recently (such as file names and file contents). After obtaining reference information for decoy files related to the topic from the decoy data base, it creates decoy information for the relevant user and uploads it to the decoy distribution server.

(3) The decoy displaying plugin periodically obtains decoy information from the decoy distribution server and displays decoy files if the user is high risk. In this system, instead of the decoy information (Fig. 1) that is sent from the decoy system to the behavior analysis system, the decoy displaying plugin sends to the behavior analysis system, together with the logs of accesses to files on SharePoint, information (labels) indicating whether a decoy file was accessed. When the behavior analysis system confirms access to a decoy file, it further narrows down the user as higher risk.

4.2 Operational image of the prototype system

Using the dashboard of risk values from the behavior analysis and the screen where decoy files are displayed as examples, the following outlines the operational image of the prototype (Fig. 3).

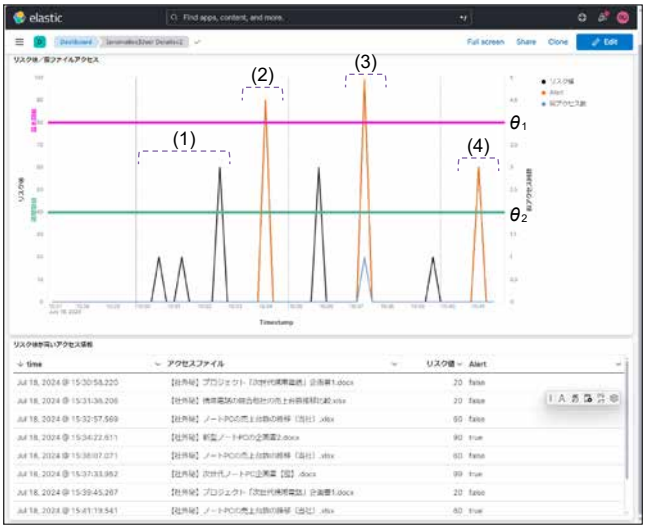
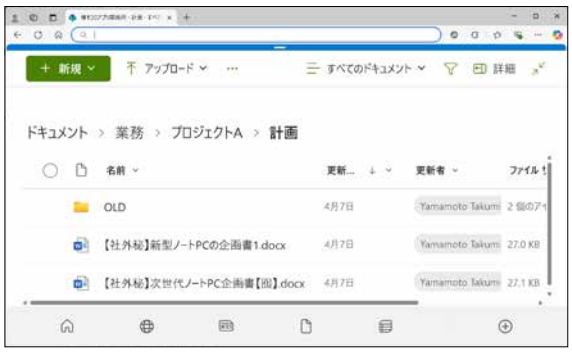


Fig. 3 Example of a dashboard for risk values in behavior analysis

(1) The behavior analysis system analyzes logs related to the user's file accesses on SharePoint and calculates a risk value. An example of visualizing risk values on a dashboard is shown in Fig. 3. In the graph, the horizontal axis is time and the vertical axis is the risk value, and the line shows the calculated risk values for a given user's behavior at each point in time.

(2) The behavior analysis system generates alerts when the risk value exceeds the specified threshold, and creates risk information. The decoy selection server verifies the presence of an alert by retrieving the risk information. Under normal conditions, the threshold is set to θ_1 (pink line in Fig. 3). In the example in Fig. 3, at point (2), the risk value exceeded the threshold θ_1, so risk information is created immediately afterward and the decoy selection server confirms the alert. In addition to the risk value, alerts include items such as the threshold and information about the accessed file (bottom of dashboard in Fig. 3).

(3) When an alert is generated, a decoy file is displayed to the relevant user. Based on information obtained from the decoy data base, it is also possible to reproduce the decoy file's meta information (timestamp, file size, last updater, etc.). An example of displaying a decoy file in a browser is shown in Fig. 4. In the example in Fig. 4 (a), the string "囮" (meaning "decoy") is added to the file name for explanatory purposes, so it is clear whether it is a decoy file. The example in Fig. 4 (b) shows the screen when a decoy file is opened in a browser. Note that, in the figures in this paper, the file names and file contents shown are entirely fictitious and have no relation to actual business operations. Also, while the decoy files were prepared manually this time, we believe that by leveraging generative AI technologies that are evolving rapidly in recent years, it would be possible to automatically generate large quantities of high-quality decoy file content; however, careful attention is required regarding the copyrights of the generated content.



(a) Decoy file displayed on the SharePoint screen

(b) Screen when opening a decoy file in a browser

Fig. 4 Example of a displayed decoy file

(4) When a user accesses a decoy file, the alert generation threshold is changed ($\theta_1 \rightarrow \theta_2$) to a more stringent value for that user only and monitoring is increased. In the example in Fig. 3, the threshold used changes from the pink line (θ_1) to the green line (θ_2), so that alerts will be generated even for lower risk values (in Fig. 3, (4)). If there is no access to a decoy file for a certain period, the threshold will be returned to its original value (θ_1). To make it clear whether a decoy file was accessed, when there has been access to a decoy file, a blue line graph is drawn in addition to the risk value line graph (Fig. 3, (3)). Whether a file is a decoy file is determined based on the label attached to the file access log.

Note that, for explanatory purposes, the risk values in Fig. 3 are calculated based on predetermined rules. In actual operation, it is also possible to use machine-learning-based anomaly detection technologies that do not require manually defining rules in advance.

4.3 Addressing the possibility of the countermeasure being disabled

Finally, it is conceivable that an insider who learns that this countermeasure has been introduced may try to disable it. For attacks that attempt to disable the plugin, they can be mitigated by enforcing the plugin via the organization's group policy and configuring the environment so that SharePoint hosting confidential files cannot be accessed without the plugin. For insiders who try to distinguish decoy files by checking file meta information or decoy information, it is possible to counter them by scoring such actions as dangerous behavior and applying additional monitoring.

5. Conclusion

To protect against insider threats that have been difficult to detect using conventional methods, we proposed a method that indirectly uncovers an insider's malicious intent by dynamically placing decoy files likely to attract the interest of users who behave differently than usual, and narrowing down insiders based on tendencies of accessing the decoy files. We also presented a simple example of implementation and demonstrated its feasibility.

Going forward, we will conduct user studies based on past insider cases and examine the effectiveness of the proposed approach.

References

(1) Proofpoint, Inc.: 2022 COST OF INSIDER THREATS GLOBAL REPORT (2022)
https://protectera.com.au/wp-content/uploads/2022/03/The-Cost-of-Insider-Threats-2022-Global-Report.pdf
(2) Independent Administrative Agency Information-technology Promotion Agency: "Top 10 Information Security Threats 2025" (2025)
https://www.ipa.go.jp/security/10threats/10threats2025.html
(3) National Counterintelligence and Security Center: INSIDER THREAT MITIGATION FOR U.S. CRITICAL INFRASTRUCTURE ENTITIES, GUIDELINES FROM AN INTELLIGENCE PERSPECTIVE (2024)
https://www.dni.gov/files/NCSC/documents/nittf/20240926_Insider-Threat-Mitigation-for-US-Critical-Infrastructure.pdf
(4) IBM Japan, Ltd.: What is User and Entity Behavior Analytics (UEBA)
https://www.ibm.com/jp-ja/topics/ueba
(5) Yamamoto, T., et al.: Proposal of an Insider Detection System Using a Decoy, Computer Security Symposium 2024 Papers, 494–500 (2024)
(6) Microsoft Corp.: What is SharePoint?
https://support.microsoft.com/en-us/office/what-is-SharePoint-97b915e6-651b-43b2-827d-fb25777f446f
(7) Microsoft Corp.: What is Azure?
https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-azure
(8) Elasticsearch B.V.: Elastic Stack
https://www.elastic.co/jp/elastic-stack