Cyber Insider Mission Detection for Situation Awareness

Haitao Du, Changzhou Wang, Tao Zhang, Shanchieh Jay Yang, Jai Choi and Peng Liu

Abstract Cyber insider detection is challenging due to the difficulty in differentiating legitimate activities from malicious ones. This chapter will begin by providing a brief review of exiting works in the machine learning community that offer treatments to cyber insider detection. The review will lead to our recent research advance that focuses on early detection of ongoing insider mission instead of trying to determine whether individual events are malicious or not. Multiple automated software agents are assumed to possess different account privileges on different hosts, to perform different dimensions of a complex insider mission. This work develops an integrated approach that utilizes Hidden Markov Models to estimate the suspicious level of insider activities, and then fuses these suspiciousness values across insider activity dimensions to estimate the progression of an insider mission. The fusion across cyber insider dimensions is accomplished using a combination of Fuzzy rules and Ordered Weighted Average functions. Experimental results based on simulated data show that the integrated approach detects the insider mission with high accuracy and in a timely manner, even in the presence of obfuscation techniques.

Research supported by DARPA Cyber Insider (CINDER, FA8750-11-C-0038) program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. Distribution Statement A—Approved for Public Release, Distribution Unlimited.

H. Du · S. J. Yang (✉)
Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY, USA
e-mail: jay.yang@rit.edu

C. Wang · J. Choi
The Boeing Company, Seattle, WA, USA

T. Zhang · P. Liu
College of Information Sciences and Technology, Pennsylvania State University, University Park, PA, USA

R.R. Yager et al. (eds.), *Intelligent Methods for Cyber Warfare*,
Studies in Computational Intelligence 563, DOI 10.1007/978-3-319-08624-8_9

1 Introduction

Cyber insider threats have attracted much attention within the past decade [1, 3, 5, 8, 10, 11, 13], and raise concerns in various research communities including psychology, criminal justice, computer science and engineering. The key challenge to detect insider threats from the computing perspective lies in the difficulty to differentiate observables that are individually legitimate but together cause threats to critical information loss or operation degradation. This becomes even more challenging when multiple software agents are used in a collusive manner to execute insider activities in different dimensions of an insider mission.

The research undertaken in the past decade on cyber insider detection, for the most part, focuses on determining whether individual actions are malicious or not. This focus has shown to be not successful due to the inherent limitation that insider activities are mostly legitimate and can easily fits, or mimicked to fit, normal behavior profiles. Recognizing this limitation, this chapter discusses an approach that focuses on detection of the progress of an overall insider mission, instead of struggling with finding malicious event observables. Expanding from the multi-perspective notion discussed in Raissi-Dehkordi and Carr [11] and the colluding user roles in Kohli et al. [5], this work assumes that an insider mission is consisted of several dimensions of insider activities. These multi-dimensional insider activities require privileges likely to span across multiple account types and thus a number of software agents are needed to complete the mission. This is not an unreasonable assumption for complex insider missions that are critical and hard to analyze. Note that the objective is *not* to determine whether individual observables are caused by these insider activities; rather, it is to elevate a threat level as early as possible when an insider mission is likely being executed.

To accomplish the above research goal, one needs to go beyond the existing intrusion or misuse detection techniques that either assume malicious behaviors exhibits localized (e.g., per-process, per-user account) deviations from normal behavior or rely on pattern matching against known attack signatures. This chapter will describe an integrated approach that utilizes Hidden Markov Models (HMM) to estimate the suspicious level of insider activities, then fuses these *suspiciousness* values across insider activity dimensions using a combination of Fuzzy systems and Ordered Weighted Average functions to project the progression of an insider mission. The approach combines the benefits of data-driven learning and knowledge-based fusion techniques, to provide a robust system that exhibits early warning capabilities even in the presence of obfuscation techniques used by colluding software agents. The timely detection of cyber insider mission is essential to enhance situation awareness of the overall operation environment. Experimental results based on simulated data show that the integrated approach detects the insider mission with high accuracy and in a timely manner, even in the presence of obfuscation techniques.

2 Related Work

Salem et al. [12] provided a comprehensive survey on cyber insider attack detection in the computer security literature. They categorized the existing works into host-based user profiling and network-based sensing approaches. Host-based user profiling draws similarity to the techniques used for more general human insider behavior profiling works [3, 10, 13]. This set of work is limited, particularly in the cyber space, in that software agents can easily mimic legitimate usage. Relying on differentiating malicious insider cyber observables from legitimate ones is simply inconceivable and impractical.

Early work on cyber insider detection overlapped significantly with the general anomaly-based intrusion detection systems that built upon data mining and machine learning techniques. Singh and Silakari [14] reviewed 18 cyber attack detection systems and identified techniques such as associative rules, Hidden Markov Model (HMM), classification, clustering, Bayes network, Support Vector Machine (SVM), Principle Component Analysis (PCA), neural network, decision tree, and self organizing map. Unlike traditional knowledge-extensive signature based detection techniques, these data mining and machine learning techniques explored large data and machine intelligence to expedite the speed or expand the capability of attack detections.

Exsiting work often focuses on a single aspect of cyber attacks. For example, Liu et al. [6] proposed a multilevel framework as a high-speed transparent network bridge at the edge of the protected network to identify network applications, generate and detect content signatures and detect covert communication. It classified network traffics using statistical and signal processing techniques for signature generation and feature extraction.

Bertino and Ghinita [2] proposed a pattern matching based mechanism to create profiles of nominal user behavior and detect anomalous behavior with respect to database SQL queries. They identified a number of activities that are indicative of data exfiltration by insiders: data identification, retrieval, movement, and exfiltration. Mathew et al. [7] argued that query syntax alone is a poor discriminator of user intent, which is much better rendered by what is accessed. They proposed to model database access patterns profiling the data points that users access, in contrast to analyzing the query expressions. Statistical learning algorithms are trained and tested using a feature-extraction method to model users' access patterns.

Hu and Panda [4] presented a model for detecting insider malicious activities targeted at tampering the contents of files for various purposes. It employs two-dimensional traceability link rule mining to identify intrinsic file dependencies and model file access patterns. Activities that modify data without complying with various file traceability link rules will be identified as suspicious activities.

Raissi-Dehkordi and Carr [11] proposed to extend the notion of profiling by aggregating statistical analysis in multiple system perspectives and performing classification using SVM. Specifically, they analyzed metrics such as user usage behaviors, file server access statistics, and database server access statistics, and established tens

of SVMs to perform classification. One of their objectives was to use these multiple SVMs to tackle the colluding insider problem. Their experiments showed a slight improvement by missing around 25 % instead of 30 %. In terms of colluding cyber insider attacks, Kohli et al. [5] discussed a risk assessment framework that shown how multiple insider and even outsider roles can collude to perform attack and cause serious risks.

Cyber insider attackers, in comparison to outsiders, are stealthier to avoid being caught. Yang et al. [17] proposed an enhanced packet matching algorithm to detect stepping-stone insider attacks through comparing outgoing and incoming connections. In such attacks, the insiders use compromised outside computers as stepping-stones to launch their attacks against inside targets. This and similar techniques can be used to detect activities in covering the trace, a dimension often overlooked by existing works.

The notion of evaluating multiple dimensions of a cyber attacks is appealing, as it presents an opportunity to provide a robust solution that does not rely on detecting anomaly in a single aspect of cyber attacks, which can be error prone. Furthermore, modern cyber defense system often implements separation of user and system privileges, and, thus, an insider attack will require multi-dimensional penetrations into, e.g., file system, database, and web application. The approach to be described in the next section employs such a multi-dimensional approach, where HMM is used to generate the suspicious level, defined by a log-likelihood function, for each dimension. The suspicious level detection can be potentially further improved with other techniques. For example, Parveen et al. [9] proposed an ensemble-based data stream mining techniques to classify rare anomalies from dynamic data streams of unbound length. It demonstrated substantially increased classification accuracy over traditional supervised learning methods for real insider threat streams due to automatic adaptation of the models for evolving data. The suspicious levels across dimensions will then be fused by a combined used of Fuzzy rules and Ordered Weighted Average to produce an insider mission score over time. This combination of data-driven anomaly detection and knowledge-driven fusion will be shown to exhibit superior performance.

3 Approaches and Components

The insider mission scenario investigated in this work is described as follows: the ultimate goal of the intrusion is altering sensitive data stored in database. The targeted victim system has an web interface to allow user to query and potentially change the data with approval. In addition, the target system has strict security policy, every change on the data should have a report, which is a file saved in file system. To accomplish the intrusion task, the insider should take actions in different *dimensions*, from reconnaissance (Dim A), tamper data in database (Dim B), tamper data in file system (Dim C), tamper data in web UI (Dim E), watch for sensitive data updates (Dim E), cover the trace (Dim F).

The insider mission identification system has two major components, the event to activity (E2A) module and the cross dimension mission identification (CDMI) module. When an automated software agent, suspicious or not, performs various activities to achieve (both suspicious insider and normal business) mission objectives, it leaves traces, i.e., a sequence of traces staging an attack on the victim system or network. These traces can be tracked by host-based or network-based sensors and reported as *event instances*. The purpose of the E2A module is to map the events into activity space to estimate the degree of suspiciousness by calculating the deviation (log-likelihood) from the internal state machine that models normal behavior. The suspicious activities are categorized into different insider dimensions based on expert knowledge. The CDMI module fuses the output of the E2A module, i.e., possible activities in different insider dimensions and their suspiciousness values to determine a mission score indicating the likelihood of existence of an insider mission.

3.1 Event to Activity Module

The main purpose of the E2A module is to simplify and compress the problem space from the event domain to the activity domain. When insider activities are performed for achieving a mission, each activity will leave traces in the network traffic logs or file systems. These traces will be inspected by network- or host-based security sensors to generate events. In our target insider mission, there are hundreds or thousands of event types, since different sensors often generates different types of events and each sensor may generate multiple types of events. In addition, normal business operations also leave traces and lead to observable events, especially when sensors are tuned to capture events from insider activities that are very similar to normal business activities. Here, the events are observable and available for our mission identification task, but the exact underlying activities are hidden and unknown and need to be inferred from the events.

In general, the same activity may cause multiple observed events, and different activities may cause the same type of observed events. Hence a probabilistic model may be used to infer activities from events. Note that an individual event (instance) by itself usually does not provide sufficient indication of whether it is observed from an insider activity or the normal business operation. Instead, the preceding and succeeding events may provide additional context to help determine how likely a given event is observed when a given type of activity is performed. As a result, the temporal order of events is important in the event-to-activity inference. In the E2A module, Hidden Markov Model (HMM) is used to perform inference for corresponding activities from observed events and to calculate the suspiciousness of inferred activities. Each event type is considered an observable symbol, and each activity type is considered a hidden state in the HMM.

The HMM in the E2A module is initially specified by a group of three security experts with experience of enterprise penetration tests and cyber analytics, and then improved through training using historic data. Specifically, the types of events, the

types of activities, and the emission relationship from activities to events (i.e., whether an event can be caused by an activity), are specified based on the target insider mission and business operation environment. Security experts are often knowledgeable and skillful enough to provide such structure knowledge, but may have difficult to specify the exact probabilities in the HMM. The HMM training only requires historic sequences of observed events (without manual labeling of activities) to tune the probabilities. Once created and trained, the HMM can be used to calculate the forward probability for a particular event using only its preceding events. This enables us to support online mission identification as events are observed as a data stream. In HMM inference, one can also calculate the posterior probability for a particular event using the full sequence of events (including both its preceding events and its succeeding events). This enables us to find the optimal probabilities in offline or batch mode mission identification as a comparison baseline to measure the online mission identification method.

In addition, the E2A module estimates the *suspiciousness* of each inferred activity. This value is important for the CDMI module in determining whether the insider mission exists. We use the log-likelihood to estimate the suspiciousness of a given activity. In particular, Let e_i be the ith observed event and the probability associating e_i to each activity a_j is p_{ij}. The best activity match is the activity with the maximum probability $p_i^* = \max_j(p_{ij})$. The suspiciousness of e_i to a_j is defined as

$$L_i \triangleq -(\log(\prod_{k=1}^{i} p_k^*))/i = -(\sum_{k=1}^{i} \log(p_k^*))/i,$$

where p_1^* is set to 1. The suspiciousness value of an inferred activity indicates how bad the activity fits the normal activity model given the observed events in the context.

Figure 1 is an illustrative example of an HMM used in E2A module. The nodes labeled with $A_j,\ j \in \{1, \ldots, M\}$ represent the (types of) activities defined in the insider mission scenario; $E_i,\ i \in \{1, \ldots, N\}$ ($N \gg M$) represent the (types of) events reported by security sensors. An edge between two activity nodes represents the transition probability for the next activity after a given activity. On the other hand, an edge between an activity node and an event node represent the probability for that event being observed when the activity is performed. Note that the HMM is very sparse, because an activity usually only causes a few types of events being observed. Once the HMM is trained, for any given sequence of newly observed events, one can use the HMM to infer the underlying activity for each individual observation, as well as the suspiciousness of the inferred activity. The better the activity fits the model (in the context of other inferred activities), the less suspicious the activity is. On the other hand, when an activity does not fit the model well, it is considered suspicious, but not necessarily malicious. The suspiciousness values will be used by the CDMI module for further analysis.

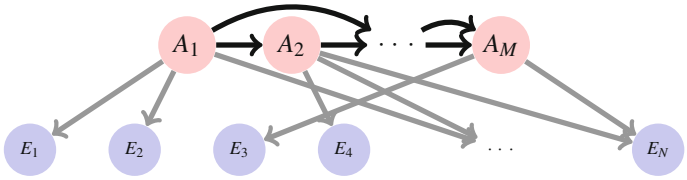

Fig. 1 An illustrative example of E2A hidden Markov model

3.2 Cross Dimension Mission Identification Module

The Cross-Dimension Mission Identification (CDMI) module processes the higher-level abstraction, i.e., the dimension specific activity level information to estimate the contribution that each set of hypothesized activities has made toward the completion of the insider mission. It is expected that tens of activity types will be used to represent hundreds of or more event types. CDMI aims at analyzing the activity suspiciousness value across insider dimensions to estimate the overall insider mission progress over time. An insider mission score, ranging between 0.0 and 1.0, will be produced to reflect the threat level of any ongoing insider mission.

Three major algorithms are developed to determine the mission score from the suspiciousness values of activities in different insider dimensions. Dynamic Activity Discovery (DAD) selects suspicious activities based on the E2A outputs, Intra-Dimension Fusion (IDF) aggregates the activity scores within each insider dimension to produce a completeness score for each dimension, and Cross-Dimension Fusion (CDF) takes the completeness scores and generates the final mission score.

The inputs to CDMI include the probability values (p_{ij}) that associate each event e_i to an activity a_j and the corresponding suspiciousness values (L_i) produced by the E2A module. Note that each event observable is now treated by CDMI as a potential insider activity with a suspiciousness value. The term 'suspiciousness' is emphasized because the goal is not to determine whether an event is truly an insider activity or not. Instead, the goal is to use the suspiciousness values to aggregate potential insider activities in different dimensions to determine a mission score.

The suspiciousness value, which is the log-likelihood, represents how much the corresponding individual event/activity deviates from the normal behavior given the contexts occurring before it. DAD further calculates the exponential weighted moving average (EWMA) of the suspiciousness values, to reflect how the sequence has been gradually deviating from normal behavior. The EWMA of log-likelihood is compared to a threshold derived based on the training set (i.e., the normal behavior). The events/activities that exceed the threshold will be used to produce the activity score for each activity type. The process of DAD is given in Algorithm 1.

The main objective of IDF is to evaluate how *complete* each insider dimension is given the suspicious activity level observed in each time window. A completeness score for a given dimension is determined by fusing the suspicious activities in the same insider dimension. The first step of this process is to determine a Suspicious

Algorithm 1 Dynamic Activity Discovery Algorithm

Given EWMA parameter α, log-loss threshold L, E2A probability p_{ij} and suspiciousness L_i
Set filtered log-likelihood $L_f(0) = L(0)$
for all Event e_i in the corresponding time window **do**
  EWMA log-likelihood $L_f(i) = \alpha L(i) + (1 - \alpha)L_f(i - 1)$
end for
Initialize Suspicious Activity Matrix M
for all Event e_i in the corresponding time window **do**
  if Filtered Log-likelihood $L_f(i) > L$ **then**
    Get the probability distribution vector $\mathbf{P} = (p_{i1}, p_{i2}, \ldots, p_{im})$;
    Append $\mathbf{P}$ to $\mathbf{M}$
  end if
end for
return Suspicious Activity Matrix $\mathbf{M}$

Activity Vector $\mathbf{V_A}$ by combining the suspiciousness values of the events in an observation window for each activity type. The combination process is based upon the Suspicious Activity Matrix $\mathbf{M}$, and used a filtering mechanism as shown in Algorithm 2.

Algorithm 2 Suspicious Activity Vector Generation Algorithm

Given Suspicious Activity Matrix $\mathbf{M}$, Threshold T, and parameters $\alpha_1 < \alpha_2$
for all Activity type a **do**
  Set $\mathbf{M}'(0, a) = \mathbf{M}(0, a)$
  for all Event e_i that has non-zero value **do**
    if $\mathbf{M}(i, a) < T$ **then**
      Set $\alpha = \alpha_1$ for less suspicious activities
    else
      Set $\alpha = \alpha_2$ for more suspicious activities
    end if
    $\mathbf{M}'(i, a) = \alpha \mathbf{M}(i, a) + (1 - \alpha)\mathbf{M}'(i - 1, a)$
  end for
  $\mathbf{V_A}(a) = 1 - (\prod_{i \in \text{observation window}}(1 - \mathbf{M}(i, a)))$
end for
return Suspicious Activity Vector $\mathbf{V_A}$

The Suspicious Activity Vector $\mathbf{V_A}$ represents the overall likeliness of an insider activity occurring in a time window. The filtering mechanism shown in Algorithm 2 is used to capture sudden surges of suspicious activities while exhibiting slow decays to maintain the lasting effects of insider activity across time windows. From here, the system evaluates the 'percent effort' spent in each activity type as compared to the overall effort within each dimension while accounting for the criticality of the activity types. The higher the percent effort (with lasting effect) is observed and/or the more critical the activity type is, the higher the 'completeness' score is for each dimension.

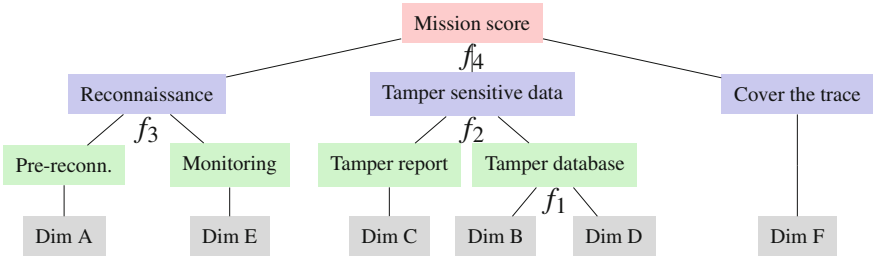

Fig. 2 Cross-dimension fusion structure

The design of the completeness functions for individual insider dimensions is based on the general framework of Ordered Weighted Average (OWA) [16].

To this end, CDF utilizes a hierarchical structure of fusion algorithms to combine the completeness scores of the various insider dimensions. Figure 2 shows the specific structure for the insider mission scenario described earlier. The hierarchical fusion structure consists of four fusion functions (f_1–f_4) and processes insider dimensions that behave similarly or tend to work together. Dimensions B and D both involve modification of the sensitive data; Fusing the two (f_1) gives an indication of the insider actions occurring on the database containing sensitive information. Alternative to tampering the database, the software agent can tamper the intelligence report of the sensitive information. Fusing the two types of tampering (f_2) shows the level of completeness with respect to the primary task of the insider mission, i.e., tampering the sensitive information. The design of f_1 and f_2 utilizes OWA framework to reflect that accomplishing either Dim B, D, or C is indicative to data tampering, but accomplishing more dimensions should still exhibit a higher overall mission score than just accomplishing one.

Similar concept is used to implement f_3 using OWA, since Dim A and E both reveal insider activity in learning about the sensitive data processing workflow process and system configurations. Dim F stands alone by itself to represent insider activities in covering evidences of data tampering. Function f_4 makes use of Sugeno Fuzzy Inference System [15] to integrate expert's knowledge for the final fusion and mission score generation. The Fuzzy system is designed so that not all of reconnaissance, data tampering and covering trace are needed to exhibit a high mission score. Based on expert recommendations, the fuzzy system emphasizes more on data tampering and covering trace. Particularly if sufficient and high confidence is shown for covering the traces of malicious activities, a sufficiently high mission score should be reached as activities in that dimension is not commonly observed.

The parameters used in the above DAD, IDF, and CDF algorithms are primarily derived based on qualitative recommendations from the domain experts for the specific insider mission scenario. Particularly, the weights used in EWMA as part of DAD, the weights used in OWA, and the fuzzy rules as part of CDF are determined by soliciting the relatively importance of and the relationship between the different activity types and insider dimensions. The threshold T and α_1/α_2 used in

IDFare designed to reflect how sensitive and how fast the system reacts to suspicious activities, respectively.

4 Experiments and Results

4.1 Experiment Design

A key obstacle in the way to insider threat research is the lack of real world data. There are two major challenges in obtaining such data. First, organizations are reluctant to share insider data due to business and security concern. Insider mission data usually contains sensitive information, such as organization security policies, sensor deployment, firewall configurations, etc. Moreover, in order to collect real insider data, all the network events should be monitored and logged, the process of which can incur expense that impact business bottom line. Second, not all the ground truth is known or tagged in real data. Users cannot tell which audit log entries are due to insider mission behavior, especially when the mission is at an early stage. Such knowledge is important for the analysis of insider's motivation and attacking strategy. In addition, it is challenging to obtain real data that reflects a variety of obfuscation techniques and colluding behaviors with different configurations of software agents. To address these challenges, this work elects to simulate insider mission process and generate the insider data set.

Insider event generation involves three distinct steps: First, each activity is decomposed into a partial order of event types. Second, the partial order of event types is verified to fulfill the activitys goal and to identify the corresponding constraints that must be satisfied. These constraints would clarify the data and control dependencies and invariants among the possible instantiations of the involved event types. Third, based on these data and control dependencies and invariants, a set of state-machines are used to automatically generate nondeterministic instantiations of the partial order of event types.

Normal background events are generated by a different set of state machines, each of which implements concurrent normal business operations. In order to build realistic normal behavior models, each workflow is decomposed into a partial order of activity types, which are mapped into a sequence of events. The insider mission events and normal background events are then interleaved to satisfy reasonable causality relationships between them, as well as taking into account that the malicious insider may attempt to hide the malicious events as much as possible.

This work considers an example of insider mission that aims at penetrating and potentially altering sensitive data, which involves a database managed by a DB administrator, a web application that allows Security Analysts to access and modify the sensitive data, and distributed file systems that store intelligence reports and review documents. The Security Analysts have the authority to directly update the data through the privileged accounts. The software agents could potentially possess

the privileges to access a combination of the above victim systems. There are seven dimensions of activity categories, including those shown in Fig. 2 and one that collects irrelevant activities.

A number of event sequences are generated for HMM training to establish the baseline normal behavior, and to test a variety of colluding insider software agent behaviors. The normal background data is generated by analyzing the common business processes for intelligence organizations. A set of state machines are developed to reflect the business processes and used to generate such data. On the other hand, the insider attack data is constructed based on the insider mission scenario described above with built-in control dependencies. Both the normal business processes and insider mission processes are then simulated, with the event instances recorded in a sequential manner in the log files as the data sets to be used for training and testing. The data generation process is kept unknown from the insider mission detection algorithm development, so that the exact states and state transitions used to generate the simualted data is not known by the algorithm. Only the high level insider mission scenario and the type of business processes are known by both the data generation team and the algorithm development team.

Figure 3 shows an example of an event instance generated. Note that many of the attributes, e.g., `activity`, `user-id`, `insider-event`, and `obfuscation`, provide the ground truth information for verification purposes only, and are hidden from the insider mission detection algorithms. The attributes, `account` and `OS-account` provide information on whcih application accountand operating system account are used for the observed event, respectively. In addition to the accounts, the primary attributes used by the E2A/CDMI modules for insider mission detection are `type-id`, showing the event type, and the `asset-id`, showing the specific asset (e.g., a webpage, a file, an entry in the database) being accessed, along with the time stamps and IP/Port information.

```
<event-id="10"
  type-id="V0200"
  description="successfully login the TAS web UI"
  activity="D2"
  account="web1025"
  OS-account="a02005"
  user-id="a02005"
  asset-id="1.7.1"
  insider-event="no"
  obfuscation="no"
  start-time="2012/01/01 07:44:41"
  end-time="2012/01/01 07:44:41"
  source-ip="10.28.46.48"
  dest-ip="10.16.48.62"
  dest-port="22" />
```

Fig. 3 An example event instance

Each data set contains about 45,000–50,000 events. The trained HMM includes 32 states, 104 observable symbols, 466 transitions between states and 125 emissions from states to observable symbols. Additional sets of normal behavior are also generated for testing against the data sets containing insider activities. The insider test sets contain approximately 1,200 insider events each and are mixed with normal event behaviors. Individual event observables for insider activity and normal activity are not distinguishable. It is the state machines that cause the slight variation between the insider and the normal activities. Additional eight types of obfuscation techniques are also included in the insider test sets. They are defined as follows:

- Noise injection: inject noise into the insider mission process, such as conducting mission irrelevant events, accessing useless assets, etc.
- Event re-timing: change the time stamp of critical insider mission events to help disguise insider mission. One possible approach could be tampering the system clock.
- Self-throttling: lower the pace of mission execution by increasing time intervals between insider events.
- Event reordering: reorder insider events without dependency relationships to disguise insider mission.
- Activity splitting: repartition the insider mission or reorganize the original activities to get a new set of activities to fool detection modules such as E2A module.
- Leveraging equivalent event sequence (event renaming/event merging): replace a series of insider mission events with equivalent event sequence. For example, modifying a file in the file system can be achieved in two ways: either editing the file and saving the modified file into the file system or deleting the original file and creating a new file with the same file name with modified content.
- Removing traces: With escalated privilege automated agents can be configured to remove or modify mission critical event logs.

The performance of the mission identification system is evaluated based on the following metrics:

- False Positive Rate: the number of false positives divided by the total number of datasets each of which does not include an insider missions. The mission identification result for a dataset is false positive if it reports the detection of an insider mission even though the dataset does not contain an insider mission.
- False Negative Rate: the number of false negative results divided by the total number of datasets each of which does include an insider missions. The mission identification result for a dataset is false negative if it reports that there is no insider mission yet the dataset does include an insider mission in the ground truth.
- Precision: the number of true positive results divided by the total number of datasets identified as dataset including an insider mission. The mission identification result for a dataset is true positive if it reports that there is an insider mission and the dataset does include an insider mission in the ground truth.
- Recall (a.k.a. Detection Ratio): the number of true positive results divided by the total number of datasets that include an insider mission in the ground truth.

- Detection Time: the time period from the start of the first insider event to the time the insider mission is detected.

4.2 Experiment Results

Consider first the performance of E2A module. Ten datasets are used for testing. Table 1 shows the false positive rate (FPR), false negative rate (FNR), precision and recall using the suspiciousness value to determine whether an event is observed from an insider activity or not. In general, one would like to see high recall so that no insider events are dropped (for later mission identification tasks) and can tolerate a relative low precision (as later mission identification modules can mitigate this). Using the mean suspiciousness value from the training dataset as a threshold, one achieves very good recall but very poor precision. By adding standard deviation of the suspiciousness value, the precision is improved at the cost of reduced recall, while achieving reasonable false positive rate and false negative rate. On the other hand, using the maximum suspiciousness value from the training dataset sacrifices the recall too much. The weighted average between the maximum and the minimum suspiciousness values becomes usable only when the weight is leaned towards the minimum value. Indeed, the maximum value might be an outlier, and hence the weighted average can be too large to obtain a high recall. The above results recommend using the mean plus standard deviation approach, as commonly used in statistical control theory, to give low FNR and high Recall.

The poor performance of E2A module is expected, since this work builds upon the premise that differentiating legitimate from insider actions is not viable. However, the activity suspiciousness values produced by E2A help CDMI to analyze the insider activity levels across different dimensions, and thus to assess whether an insider mission is ongoing.

Table 1 E2A accuracy for insider event determination

Threshold	FPR	FNR	Precision	Recall
Max	0.00	0.64	0.94	0.36
Mean	0.60	0.03	0.02	0.97
Mean + stdandard	0.23	0.07	0.06	0.93
Mean + 2*standard	0.11	0.15	0.10	0.85
Mean + 3*standard	0.08	0.20	0.13	0.80
Min*0.0625+max*0.9375	0.00	0.64	0.94	0.36
Min*0.125 + max*0.875	0.00	0.64	0.94	0.36
Min*0.25 + max*0.75	0.00	0.64	0.94	0.36
Min*0.5 + max*0.5	0.02	0.49	0.25	0.51
Min*0.75 + max*0.25	0.13	0.12	0.09	0.88

Table 2 Detection accuracy

Configuration	TP	FP	TN	FN
1	5	0	5	0
2	5	0	5	0
3	5	1	4	0
4	5	0	5	0
5	5	0	5	0
6	5	0	5	0
7	5	0	5	0
8	5	0	5	0
9	5	0	5	0
10	5	0	5	0

To measure the robustness of the overall mission identification system, ten different sets of configurations are used. For each configuration, two training sets and ten testing sets, five with only normal events and five with mixed insider and normal events, are generated. Each dataset on average includes around 50,000 events. The detection results are shown in Table 2. True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) are collected. As shown in the table, all insider missions are detected, with only one false positive reported. The false positive is due to the conservative setting of thresholds used in CDMI.

This excellent performance is achieved due to a few reasons. First, the objective here is to detect the existence of an overall insider mission, instead of differentiating insider events from the normal ones. The collective evidence, even with significant false negatives output from the E2A module, can still be indicative of the existence of an insider mission. In addition, the proposed system is built on an integrated data-driven and knowledge-driven approach. The definition of the activity types, the dimensions, and their relative importance (which translates to the parameters used in the fusion algorithms within the CDMI module) are derived by soliciting knowledge from domain experts for the specific insider mission scenario. The specific system developed here will not automatically work to detect a very different insider mission on a different organization, but the framework will apply. The results shown in this chapter suggest the promising outlook to detect insider mission using the novel integrated data and knowledge-driven approach, which is not conceivable in other existing works. The following shows additional results to demonstrate the robustness of the framework in the presence of missing observables and advanced obfuscation techniques.

For each configuration, the insider testing sets are reduced with each event independently dropped with a fixed probability. Five fixed probabilities, 5, 10, 15, 20, and 25 %, are used for different robustness tests. Table 3 shows the detection ratios for the cases with increasing dropping ratio. All insider missions are detected in this experiment even though 25 % of the events are dropped. This superior performance is

Table 3 Detection with missing observations

Drop probability	Detection ratio (%)	Average # alerts generated
0	100	122.69
0.05	100	102.70
0.1	100	98.74
0.15	100	94.20
0.2	100	93.51
0.25	100	93.27

due to that CDMI examines the suspiciousness values of activities in each time window, and raise an 'alert' as long as there is sufficient evidence suggesting an insider mission is ongoing in that time window. For a given test set, an insider mission is considered detected when the first alert is raised. As a result, as long as the removed events do not eliminate the possibility to raise alerts in ALL windows, the insider mission can be detected. Table 3 shows the averaged number of alerts generated for each case. While all insider sets are detected, not all time windows will generate alerts showing insider activities due to the dropped events.

Table 4 shows the detection ratios when different obfuscation techniques are used. As in the previous experiment, all insider missions are detected regardless of the obfuscation techniques used. The average number of alerts (i.e., time windows with insider mission detected) varies, however, when different obfuscation techniques are used. One interesting observation is that the 'event re-timing' and 'self-throttling' techniques result in significant different performance in terms of average number of alerts generated. This is because these two techniques spread the insider events over a longer time span, and thus more time windows see raised alerts.

Table 4 also shows the detection time when different obfuscation techniques are used. Other than 'event re-timing' and 'self-throttling', the mission identification system detects the insider mission around the same time if there were no obfuscation techniques. The longer detection time is due to the same reason as that for larger average number of alerts; that is, the spreading the same number of insider events

Table 4 Mission detection against obfuscation techniques

Obfuscation	Detection ratio (%)	Average # alerts	Average detection time
Noise-injection	100	61.9	125.5
Event-retiming	100	126.7	196.2
Self-throttling	100	126.7	196.2
Event-reordering	100	68.5	124.0
Activity-splitting	100	62.1	123.8
Equivalent-sequence	100	60.7	123.9
Trace-removing	100	53.1	124.1

over a longer period of time makes it more challenging to gather sufficient evidence to declare the existence of an insider mission. Obviously these types of evasion are difficult to detect due to the event sparsity.

5 Conclusion

Going beyond the classical intrusion detection, this work developed hierarchical data processing for insider mission identification by abstracting activities from lower level events, estimating level of suspiciousness, all of which have been evaluated for a final mission score that relies on both the data abstraction and domain knowledge. The emphasis is to show how one can reveal the insider mission while activities performed by automated software agents were hidden among the legitimate activities.

The integrated approach of data driven (E2A) and knowledge driven fusion of insider activity (CDMI) has been shown to be highly successful to differentiate cases where colluding autonomous agent activities are present versus those with no insider activity. Hierarchical fusion allows to account for the completion of individual insider dimension, driven by suspicious level of insider activities, and, thus, robust to obfuscation techniques attempting to hide the autonomous agent activities.

References

1. Ali, G., Shaikh, N.A., Shaikh, Z.A.: Towards an automated multiagent system to monitor user activities against insider threat. In: Proceedings of International Symposium on Biometrics and Security Technologies, pp. 1–5 (2008)
2. Bertino, E., Ghinita, G.: Towards mechanisms for detection and prevention of data exfiltration by insiders. In: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, pp. 10–19 (2011)
3. Buford, J.F., Lewis, L., Jakobson, G.: Insider threat detection using situation-aware MAS. In: Proceedings of 11th International Conference on Information Fusion (2008)
4. Hu, Y., Panda, B.: Two-dimensional traceability link rule mining for detection of insider attacks. In: Proceedings of the 43rd Hawaii International Conference on System Sciences (2010)
5. Kohli, H., Lindskog, D., Zavarsky, P., Ruhl, R.: An enhanced threat identification approach for collusion threats. In: Proceedings of Third International Workshop on Security Measurements and Metrics, pp. 25–30 (2011)
6. Liu Y., Cobett, C., Chiang K., Archibald, R., Mukherjee, B., Ghosal, D.: SIDD: a framework for detecting sensitive data exfiltration by an insider attack. In: Proceedings of the 42nd Hawaii International Conference on System Science (2009)
7. Mathew1, S., Petropoulos, M., Ngo, H.Q., Upadhyaya, S.: A data-centric approach to insider attack detection in database systems. In: Proceedings of the 13th international Conference on Recent advances in intrusion Detection, pp. 382–401 (2010)
8. Maybury, M., Chase, P., Cheikes, B., Brackney, D., Matzner, S., Hetherington, T., Wood, B., Sibley, C., Marin, J., Longstaff, T.: Analysis and detection of malicious insiders. Technical report, MITRE (2005)

9. Parveen, P., Weger, Z.R., Thuraisingham, B., Hamlen, K., Khan, L.: Surpervised learning for insider threat detection. In: Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence, pp. 1032–1039 (2011)
10. Pfleeger, S.L., Predd, J.B., Hunker, J., Bulford, C.: Insiders behaving badly: addressing bad actors and their actions. IEEE Trans. Inf. Forensics Secur. **5**(1), 169–179 (2010)
11. Raissi-Dehkordi, M., Carr, D.: A multi-perspective approach to insider threat detection. In: Proceedings of IEEE Military Communications Conference, pp. 1164–1169 (2011)
12. Salem, M.B., Hershkop, S., Stolfo, S.J.: A survey of insider attack detection research. Insider Attack Cyber Secur. **39**, 69–90 (2008)
13. Santos, E., Nguyen, H., Yu, F., Kim, K., Li, D., Wilkinson, J.T., Olson, A., Jacob, R.: Intent-driven insider threat detection in intelligence analyses. Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. **2**, 345–349 (2008)
14. Singh, S., Silakari, S.: A survey of cyber attack detection systems. Int. J. Comput. Sci. Netw. Secur. **9**(5) (2009)
15. Wang, L.X.: A Course on Fuzzy Systems. Prentice-Hall press, USA (1999)
16. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decision-making. IEEE Trans. Syst Man Cybern. **18**(1), 183–190 (1988)
17. Yang, J., Ray, L., Zhao, G.: Detect stepping-stone insider attacks by network traffic mining and dynamic programming. In: Proceedings of the 2011 International Conference on Advanced Information Networking and Applications, pp. 151–158 (2011)