A User Behavior-based Approach to Detect the Insider Threat in Distributed Diagnostic Imaging Systems

Hassan Sharghi, Kamran Sartipi
Department of Electrical, Computer and Software Engineering
University of Ontario Institute of Technology
Oshawa, ON, L1H 7K4, Canada
{*Mohammadhassan.Sharghigoorabi, Kamran.Sartipi*}@uoit.ca

***Abstract*—Modern diagnostic imaging systems integrate several PACS (Picture Archiving and Communication System) through datacenters that allow a large community of users to access and share sensitive patients medical images. As the user access to the medical images in the non-local PACS systems is based on a trust model, the data integrity and privacy in such systems are vulnerable to any malicious user behavior. Moreover, the limited scope and precision of the existing policy-based access control solutions prevent them from detecting such malicious user behaviors after they are authenticated to use system resources freely. In this paper, we propose an approach for analyzing the user behaviors that allows the administrators to identify the users whose behaviors may jeopardize the data privacy and system integrity. In this context, the system administrator can define any pattern of the suspicious user behavior using our new behavior pattern language; a constraint-based pattern-matching engine will identify the instances of the suspicious behavior pattern in the system's audit-log repository; and a decision support system will present the excerpt findings to the system administrator with the overall goal of refining the access control policy rules. We present a case study which indicates our proposed approach provides promising results.**

***Keywords*-User Behavior; Security; Pattern Matching; Insider Threat; Diagnostic Imaging System; Event-log Repository.**

I. Introduction

Healthcare information systems are attractive targets for cyber threats due to confidential patient information in the electronic health record (EHR) systems. Medical data disclosure takes the second place in the list of highly reported data breach cases [1]. Therefore, maintaining the confidentiality of the patients' clinical information is crucial for the healthcare organizations. The IBM Cyber Security Intelligence Index reports that 55 percent of attackers are insiders [2], and the number and extent of damage of insider breaches continue to rise year by year [3]. The increase in the number of successful insider attacks on healthcare systems in recent years shows that the current access control mechanisms are inefficient in protecting against insider threat. For instance, Utah's health data breach led to the disclosure of nearly 1.1 million records of personal information [4]. The University of Iowa Hospitals and Clinics reported that the electronic health records of thirteen University of Iowa football players were inappropriately accessed by five employees [4].

Policy-based access control is nowadays a common mechanism to protect resources in distributed systems and helps enterprises to enforce policies that define who should have access to which resources, and under what circumstances [5]. The access control mechanism is responsible for authenticating and authorizing the system users who issue access request for specific resources. However, authenticated and authorized users can affect the security level of a service-based distributed system through misusing the resources. Therefore, the behavior of users should be constantly monitored and the policy of the system should be regularly updated by the administrators to prevent such incidents. Monitoring and analyzing the activities of users are not easy tasks for administrators, particularly in a large distributed system that supports a large number of users with different levels of authorization. Moreover, the changing nature of user behaviors imposes significant challenges with respect to system monitoring, auditing, and fault diagnosing.

Threats from the inside of an IT framework are extensively costly for an organization, since insiders often have significant information about the system's critical assets and enough permission to perform intentional or unintentional damaging activities. The ”insider threat” is cited as one of the most pressing security challenges, and it is also considered as a hard problem to deal with because of the difficulty of distinguishing the harmful behavior of insiders who are legitimately granted the authorization to access the resources. After granting permission to distributed system's users, they have been legitimately empowered with the right to access or decide about one or more resources of the system. Consequently, detecting legitimate users who misuse resources is a challenge in large distributed systems.

Apart from salient features of the PACS systems, the proprietary design and implementation of the PACS systems limit their capabilities to interact with modern technologies. With respect to the security aspects, currently access to the medical images is based on a ”trust model”, where each

PACS system uses a local authentication and authorization mechanism to grant the users to access the patient images. Such a trust model for a distributed Diagnostic Imaging (DI) system lacks federated capabilities to manage user authentication, authorization and the consistency of security policy rules. To deal with the issue, we designed a federated authentication mechanism for distributed DI systems in our previous work [6].

In this paper, we propose a behavior based analysis approach in order to discover the instances of any kinds of suspicious behavior pattern defined by the analyst. Once the user behavior pattern is defined, a constraint-based pattern matching engine will search the system's event log repository to discover the instances of the defined behavior pattern that satisfy different hard and soft constraints. The discovered behavior instances will be provided to an intelligent decision support system for extracting meaningful results. The result will be offered as suggestions and guides that allow the administrators to identify the gaps and flaws in the security policy rules. Overall, this paper presents the following contributions: (i) provisioning a behavior monitoring mechanism for an agent-based security middleware that enables the administrator to monitor the behavior of legitimate users; and (ii) designing a model for behavior-based pattern matching and searching the instances of a particular behavior pattern.

The remaining of this paper is organized as follows: Section II provides an overview of related works. Our proposed approach for behavior monitoring and analyzing is presented in Section III, followed by a case study in Section IV. Finally, Section V provides some concluding remarks and outlines the direction for future research.

II. Related Work

Designing an efficient and scalable infrastructure for monitoring and processing behavior patterns has been a major research interest in recent years. Such infrastructure provides a mechanism for the enterprise to monitor the behavior patterns and anomalous behaviors of their customers. Consequently, behavior modeling has been increasingly recognized as a challenge for associating semantics with the human's actions to be used in different environments and for different purposes. Cao et al. [7, 8] consider behavior as individual activities represented by events as well as activity sequences conducted by the user within certain context. They defined four dimensions, as: actor, action, environment and relationship to represent abstractly the user behavior. The assigned attributes allow for describing features of each dimension and temporal logic is used to express the properties and relations between elements of a desired behavior.

In [9] the authors introduce a semantic model for representing and computing behavior in online communities. An ontology was defined that represents all involved entities and their interactions. Representing the behavior pattern by semantic rules has been also proposed in literature so that in [10] Event-Condition-Action (ECA) rules was considered to represent the frequent behavior patterns.

Using behavior to deal with the problem of insider threats has been explored in several papers. For example, [11] utilized a sequential pattern mining technique in order to identify patterns, by comparing the current generated pattern with the one stored in the user profile to detect the deviation of users from their normal behavior. Authors in [12] proposed a comprehensive threat assessment approach using a predictive modeling framework that integrates a diverse set of data sources from the cyber domain, as well as inferred psychological factors.

Monitoring insider misuse in the Document Control Domain (DCD) by analyzing the behavior of insider to access the documents was proposed in [13]. The authors assumed that each user has her typical access patterns (time and click ratio) to the documents during performing her duties. They utilized apriori algorithm in order to monitor the insider misuse.

The issue of insider threat can be mitigated through supervised analysis of event log to reveal suspicious behavior and extract facts about user behavior patterns. We represent abstractly the user behavior as a sequence of finite number of events. An event is represented as a tuple of attributes that describe the characteristics of interaction between a user and a system resource. Our proposed behavior pattern language enables the analyst to describe the suspicious user behavior for further exploration in order to detect the access control policy violation and specify the subsystems and users commit such behavior.

III. Proposed framework

Diagnostic Imaging repository (DI-r) as a main component of the modern distributed diagnostic imaging system provides a central repository to record medical images. Furthermore, it has the capability to make a non-proprietary image sharing system compliant with XDS-I profile [14] across health care network. The lack of federated capabilities to manage authentication and authorization and define consistent rules for access control policy are challenges in the distributed diagnostic imaging systems.

In [6], we proposed an agent-based middleware and distributed generic agent for each PACS system to make a federated identity management service. This service allows users to authenticate themselves with the infrastructure that provides a common set of policies for all participated applications in order to share the resources securely and seamlessly. Figure 1 shows the proposed architecture where the middleware has been extended by adding a user behavior analysis feature that allows the administrators to identify the authenticated and authorized users whose behaviors may jeopardize the data privacy and system integrity.

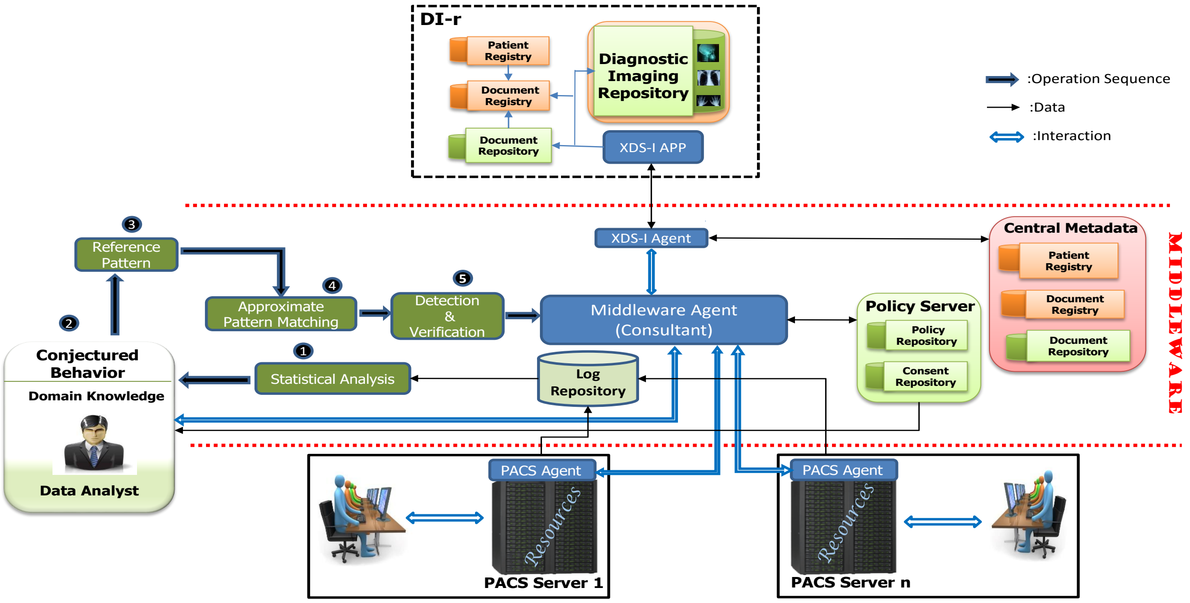

Figure 1: Overall view of the proposed infrastructure including a user behavior investigation mechanism to make a secure and seamless connection between DI-r and PACS systems.

Investigation into abnormal activities in the environment is initiated by the analyst through generating the specification of a conjectured behavior. Such specifications will be produced by examining the policy rules, patient consent forms, procedural workflows, and statistical information generated by the log analyzer. The conjectured behavior is described using our proposed Behavior Pattern Language (BPL) that drives the pattern matching process. The described behavior will be converted into a sequence of events called reference pattern and a set of constraints that represent the correlation between events. Then, the pattern matching engine searches the Log Repository to discover an ordering of events that approximately match with the reference pattern. The result will be reported to the Consultant Agent (Middleware Agent) for analysis in order to reveal important facts about user behaviors in terms of access to resources and actions performed.

A. *Behavior Pattern Language*

The user behavior presents characteristics, relationships, structures, and effects of a sequence of events in a specific application domain. An event consists of a tuple of attributes whose values represent an observation of the behavior. We propose the BPL language to describe precisely the behavior elements (events) and their relationships using a set of predefined operators.

The BPL takes advantage of important concepts in high-level programming languages such as *”yield”*, *”iteration”*, and *”block”*. It also provides operations such as: *negation, frequency of occurrences*, *time & location separations between events*, and *event sequence* that provide semantics for the behavior patterns. In this context, the *user behavior patterns* are generated through interactions between users and system resources. Different categories of behavior patterns can be identified based on: event sequences, attribute associations, and specific constraints between events based on their attribute values. Figure 2 shows an association between two sets of events that share the attribute values of location (L1, L2) and resource (S1, S2). In Section IV, we present a BPL code to describe a behavior pattern including a sequence relation.

B. *The process of detecting malicious user-behavior*

The process of seeking the instances of the defined behavior utilizes different techniques for each stage, as explained below.

Step 1: Statistical analysis
All interactions of users to access resources are recorded in a central log repository. Statistical analysis of event-log provides primitive knowledge about the usage percentage of different resources, usage frequency during different time intervals, the time spent for each resource and the frequency of a particular activity on a resource, users' demography, etc. Log data analysis can supplement the understanding of

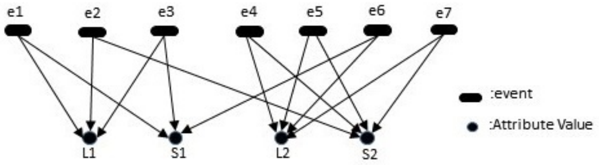

Figure 2: Association between a sequence of events

users' behavior with more concrete data, and it provides a means to investigate the users' interest in existing resources in a distributed system. The statistical information provides guidance for the analyst to explore about particular behavior in the system.

Step 2: Conjectured Behavior
The analyst always needs to perform a full exploration of the suspicious behavior in the system in order to recognize the user and subsystem in which a particular behavior occurs. The analyst describes a behavior based on initial information provided by the statistical analysis (step 1), the workflow of operations in the organization and the policy rules. The reference behavior is described by a new Behavior Pattern Language (BPL) that we proposed and developed. The BPL provides a template to describe the high level behavior that is considered as a sequence of events as well as constraints that represent the correlation of events.

Step 3: Reference Pattern
The composed behavior by BPL should be parsed in order to extract the sequence of events and constraints. The output of the parser will be a set of constraints and a sequence of events representing the abstract behavior. We call it as a reference pattern. The reference pattern plays the role of a reference which is used by pattern matching engine in order to collect similar behavior patterns to the reference pattern. This step drives a behavior pattern matching process where we intend to discover the instances of the given reference pattern. The discovered instances can be more general or more specific based on the number of concrete attribute values defined in the reference pattern.

Step 4: Approximate Pattern Matching
Approximate behavior pattern matching is proposed for the problem of identifying the instances of the defined behavior. Our proposed pattern matching mechanism searches the event log to discover the sequences of events that approximately satisfy the constraints presented by the reference pattern. Due to the nature of the behavior pattern matching, we modeled the behavior pattern matching mechanism as a Valued Constraint Satisfaction Problem (VCSP). Meanwhile, our representation for behavior pattern makes it relatively easy to express the approximate behavior pattern matching problem as a VCSP problem. Practically, this means generating a set of variables and a set of constraints for a given behavior pattern, and searching the event log to find the sequence of events whose attribute values closely satisfy the constraints. A VCSP problem is generally solved using a search algorithm. Algorithm 1 "Behavior Pattern Matching" receives the users events, size of reference pattern, set of bound attributes, set of constraints and corresponding weights, and returns the approximate instances of the reference pattern. The algorithm utilizes the function "*Propagate ()*" to propagate unary hard constraints to reduce the search space by grouping the events, and function "*Search ()*" to find appropriate events from event-groups to satisfy the constraints.

Algorithm 1 Behavior Pattern Matching

```
Input:
    A: set of attributes
    Size: number of events in the reference pattern
    X: set of bound attributes
    C: set of unary or binary constraints
    W: set of weights
    E: set of users events
Output:
    P: set of instances of behavior pattern
Local Variable:
    g: an event-group that satisfies unary hard constraints
    relevant to an event-placeholder
    G: set of all event-groups
    s: an instance of the behavior pattern
$P = \varnothing$
for i in [1 .. Size]  do
    for each unary constraint $c_{\mathrm{i}}$ in C  do
        $g_{\mathrm{i}}$ = Propagate($c_{\mathrm{i}}$, E)
        insert $g_{\mathrm{i}}$ in G
    end for
end for
for step in [1 .. SizeOf ( E)]  do
    s = Search (G, C, W)
    if $s \neq \varnothing$ then insert s in P end if
end for
```

Step 5: Detection and Verification
The result generated by the VCSP solver will be a set of instances of the conjectured behavior defined by the analyst. For each instance, the attribute values of the events (i.e, role, actions, and resources) will be examined in order to verify that the conjectured behavior is actually a normal behavior or a malicious behavior from the access control policy point of view. The Consultant Agent receives the extracted attributes in order to identify which policy rules have been affected. For this purpose, the Consultant Agent communicates with the PACS Agent located in each subsystem in order to determine the policy rules pertinent to the extracted attributes.

The Consultant Agent checks whether the determined policy rules have been violated according to the attribute values. The result will be sent for the analyst to make a decision about refining the policy rules or adding new rules. Moreover, after extracting the unacceptable facts about a user and approving them by the administrator, the Consultant Agent sends special commands to the PACS Agent in order to set some restrictions for the at fault user.

IV. Case study

We consider a distributed diagnostic imaging system whose authenticated users are enabled to create, read, update, and delete system's resources such as: documents, images, audio files, etc. The system's authorization mechanism grants or denies access requests based on a set of policy rules. Moreover, the log repositories record all interactions of users according to the RFC 3881 standard that defines the format of data to be collected and the minimum set of attributes that need to be captured for security auditing. We assume that every user's transaction is represented by a sequence of events. Each event is represented by a tuple, including some attributes such as User, Role, Location, Operation, Resource type, Time, Date, etc.

Table I: Access control policies

Policy #	Description
1	Users have no right to access images without specifying the reason, e.g., checkup, diagnosis, update, studies.
2	Users are allowed to access the images only within the specified medical environments.
3	Physicians can view their patient images solely within their respective departments.
4	Lab technician has no right to access medical images of patients.
5	Nurses can only view the images and diagnostic reports of those patients that are assigned to them.
6	Nurses and radiologists cannot access to three resources in order: patient information, MRI, and CTscan during a working day.
7	Only physicians can delegate nurses to view certain medical images of the patients as a part of diagnosis.

We present an example for a Diagnostic Imaging repository (DI-r) system to demonstrate how a suspicious behavior pattern can be used as a trigger for investigation in order to improve the security. Table I presents the access control policies that are defined by the security administrator for diagnostic report and medical image retrieval from the DI-r.

The analyst performs a statistical analysis on the event log, and she generates the following hypothesis for more investigation: *"If a user accesses to several resources in order: patient information, MRI, and CTscan of different patients during a working day, then it may be an abnormal behavior.*"
The hypothesis should be converted to a sequence of events as a reference behavior pattern in order to find the instances of such behavior issued by different users.

BPL Code 1: The BPL description of the reference pattern

```
PATTERN Example-1;
BEGIN
DEFINE Event = < User, Action, Patient,
    ↪ Location, Resource, Date, Time >
E[] = new Event (3);
set1 = { E[1], E[2], E[3] };
attVal = { <Resource, ("patient information
    ↪ ","MRI", "CTscan")>};
CALL Op1 ( set1 , attVal) ;
END
/* define operation */
DEF Op1 ( set1, attVal)
Set1[E1].Resource = attVal[Resource].value
    ↪ [0];
Set1[E2].Resource = attVal[Resource].value
    ↪ [1];
Set1[E3].Resource = attVal[Resource].value
    ↪ [2];
ASSERT EQUALITY (set1, Date);
END
```

Parsing the BPL code of the defined behavior generates a JSON (JavaScript Object Notation) file that describes the structure of the behavior pattern in terms of length, bound attributes, and the specification of constraints.

Due to the lack of access to real-world audit logs in healthcare domain, we use a dataset generator toolkit [15] to generate an event log repository including the different kinds of user behavior patterns. The generator toolkit generates user-system interactions that display the realistic features of a user behavior. We consider three categories of user behavior pattern including sequence, association, and temporal proximity. The generator toolkit receives the parameters of each pattern in order to generate and inject the instances of the pattern into dataset. For each category of the user behavior, we generated a dataset containing around 30000 events of 100 users during 30 days.

In the defined behavior, access sequentially to three particular resources is considered as a suspicious behavior that needs more investigation. After describing the specification of such suspicious behavior and converting it to a reference pattern, the approximate pattern matching engine matches the reference pattern against patterns existing in the event log.

We use the Valued Constraint Satisfaction Problem (VCSP) to model the approximate behavior pattern matching problem. The VCSP is an extension of the CSP framework in order to handle over-constrained problems and the preferences on solutions. Formally, a VCSP framework for behavior pattern matching is defined by a four-tuple vcsp = (X, D, C, f), where:
X = $\{\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_\mathrm{n}\}$ is a finite set of variables. We consider every attribute in each event as a variable. For example, x_1 = e1.User ; x_2 = e1.Role; x_3 = e1.Action
D = $\{\mathrm{D}_1, \mathrm{D}_2, ..., \mathrm{D}_\mathrm{n}\}$ is a set of domains, one for each variable (i.e., attribute).
C is a set of unary or binary constraints. Unary constraints are defined to restrict the values of one variable. However, a binary constraint defines a restriction between the values of two variables.
f is a cost function.

To define the specification of the reference behavior pattern, hard constraints are used to formalize properties that cannot be violated. However, some desired properties can be considered as preferences, and are modeled by soft constraints whose violations should be avoided as much as possible. A VCSP is characterized by a set of hard constraints that must be satisfied, and a set of soft constraints whose satisfactions are desirable. Therefore, solving a VCSP means finding an assignment (set of events) that sub-optimally satisfies a set of hard and soft constraints based on attribute values.

We consider a cost $\mathrm{w_i}$ for each constraint so that a very large number (to simulate infinity) will be considered for a hard constraint. The associated cost for each soft constraint is defined based on the number of distinct values for each variable involving in a constraint. In the VCSP an assignment is the task of assigning a value to a variable such that some constraints may be violated. The total cost of an assignment is the aggregation of the costs of constraints that are violated by the assignment.

$$f(C_1, ..., C_n) = \sum_{i=1}^{n} w_i$$

f: cost function; n: number of violated constraints; w: a positive integer number assigned to each constraint to show the effect of a constraint violation.

Table II: Parameters and results of behavior pattern matching

Dataset	Sequence	Variable	Hard const.	Soft const.	Discovered pattern
30000	3000	2	3	2	54

Table II shows the parameters and the results of solving VCSP model for the defined behavior. The discovered instances shows that there are several users who access the three resources in order: patient information, MRI, and CTscan during a working day by changing their role. The results demonstrate that the policy number 6 (Table I) is not a strong rule to prevent effectively from misusing the mentioned resources because a user can violate this rule by changing her role.

V. Conclusion

We proposed a security middleware solution for the large distributed systems consisting of several autonomous subsystems whose securities are managed by a multi-agent system to provide federated authentication and authorization to share sensitive system resources. An example of a distributed PACS and DI-r's in the domain of diagnostic imaging systems was presented. In such systems, the authenticated and authorized users (as trusted users) can intentionally or unintentionally jeopardize the valuable system resources by performing malicious operations on the resources. To tackle this subtle problem, we extended the middleware using a behavior analysis mechanism to detect the malicious behavior of trusted users. A statistical analysis of log data provides usage statistics on different resources and the kind of actions performed on those resources. This retrospective view of the statistics on resource usage along with existing policy rules provide some guidelines for the analyst to define a suspicious behavior pattern using our proposed behavior pattern language that drives a behavior pattern matching process. An approximate pattern-matching engine searches the system's audit-log repository to identify the similar instances of the defined suspicious behavior pattern. We modelled such an approximate behavior pattern matching as a valued constraint satisfaction problem to match the suspicious behavior pattern against the users' event sequences. The discovered pattern instances are examined by the security analyst to detect the highly suspicious behaviours and the subsystem where the behavior was issued. We aim at extending our approach by provisioning an agent-based automated decision support and recommendation system that analyzes the instances of the suspicious behavior to detect the violated policies or specifying the existing gap in policy rules in order to effectively assist the administrator to update the access control policy rules.

References

[1] A. Appari and M. Eric Johnson. Information security and privacy in healthcare:current state of research. *International Journal of Internet and Enterprise Management*, 6(4):279–314, 2010.

[2] IBM Security. Battling security threats from within your organization. In *Research Report*, 2015.

[3] Vormetric Data Security. 2015 vormetric insider threat report, trends and future directions in data security. In *Research Report*, 2015.

[4] Suhair Alshehri, Sumita Mishra, and Rajendra Raj. Insider threat mitigation and access control in healthcare systems. In *Technical Report, Rochester Institute of Technology*, 2013.

[5] Grigoris Antoniou, Matteo Baldoni, Piero A. Bonatti, Wolfgang Nejdl, and Daniel Olmedilla. Rule-based policy specification. In *Secure Data Management in Decentralized Systems*, pages 169–216. Springer Science+Business Media, LLC, 2007.

[6] Hassan Sharghi, Weina Ma, and Kamran Sartipi. Federated service-based authentication provisioning for distributed diagnostic imaging systems. In *IEEE 28th International Symposium on Computer-Based Medical Systems (CBMS2015)*, pages 344 – 347. IEEE, 2015.

[7] L. Cao. In-depth behavior understanding and use: the behavior informatics approach. *Journal of Information Sciences, vol. 180, no. 17*, pages 3067- 3085, 2010.

[8] C. Wang and L. Cao. Modeling and analysis of social activity process. pages 21–35. Behavior computing: modeling, analysis, mining and decision, New York, Springer, 2012.

[9] S. Angeletou, M. Rowe, and H. Alani. Modelling and analysis of user behaviour in online communities. pages 35–50. Springer Verlag Berlin Heidelberg, 2011.

[10] H.-L. Bui. Survey and comparison of event query languages using practical examples. Ludwig Maximilian University of Munich, 2009.

[11] Lucky Nkosi, Paul Tarwireyi, and Matthew O Adigun. Insider threat detection model for the cloud. In *Information Security for South Africa*, pages 1 – 8. IEEE, 2013.

[12] Frank L. Greitzer and Ryan E. Hohimer. Modeling human behavior to anticipate insider attacks. *Journal of Strategic Security*, IV(2):25–48, 2011.

[13] Jung ho Eom, Sung hwan Kim, and Tai Myoung Chung. Analysis of insider access pattern for monitoring misuse in the dcd. *International Journal of Multimedia and Ubiquitous Engineering*, 8(3):431–440, 2013.

[14] Cross-enterprise document sharing for imaging. http://wiki.ihe.net/index.php,2016.

[15] W. Ma and K. Sartipi. Synthesizing scenario-based dataset for user behavior pattern mining. *International Journal of Computer and Information Technology, vol. 4, no. 6*, pages 855–866, 2015.