International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT)
Vol. 4, No. 2, June 2013

A FRAMEWORK TO DEFENSE AGAINST INSIDER ATTACKS ON INFORMATION SOURCES

Reza Asgari[1] and Reza Ebrahimi Atani[2]

[1]Department of Computer Engineering, Guilan University, Rasht, Iran
rezaasgari@msc.guilan.ac.ir
[2]Department of Computer Engineering, Guilan University, Rasht, Iran
rebrahimi@guilan.ac.ir

ABSTRACT

As for development and growth of information systems and security organizations, protecting information against probable attacks is of great importance. External raids on these organizations, for the most part, are not practicable due to high defensive layers. Therefore to intrude on such organizations, insiders are employed. In this paper, by introducing consequence and necessity of recognition of insider attacks perils, we intend to propose a new framework for detecting and preventing from insider attacks on information systems.

The suggested framework is defined according to ontology graphs, thus, a structure of so-called ontology is firstly explained. This composition represents data structure for saving and presenting information and is then practiced to detect user's behavioral patterns within such framework. The framework consists of three phases of construction, comparison and analysis such that it first receives a set of user's requests alongside with his legal access level and in case of encountering an attack it communicates an appropriate warning message to the organization administrative system.

KEYWORDS

Insider attacks, Ontology, Intrusion detection systems, Access control systems

1. INTRODUCTION

Technology development in today's organizations and agencies hinges on development in infrastructures of information networks in such organizations. Such infrastructures raise misfortunes like hackers' attacks or outsiders on organizations. Most of organizations take care of attacks established outside, however, statistics reveal almost all of damages to them are administered by insiders [2].

Incumbent or former employees, managers and contractors are treated as essential threats to organizations because of their superior knowledge, having access to other employees' computers and databases as well as legal pass through electronic or physical security devices [3]. Employees may have access to information from different sources and through various methods such as documents, plans, files and electronic communications. Larger part of employees, alas, is unaware of trickiness of grouped information. Hence an organization is responsible for a solid idea as to how and where to distribute sensitive information.

This paper attempts to suggest a framework to detect insider attacks established upon ontology. In this paper, an introduction to detecting insider attacks is arranged in section (II). Ontology-based graphs in order to present data are proposed in section (III) and a framework to detect insider attacks is then introduced and final conclusion on the issue is drawn in section (IV).

2. NECESSITY OF RECOGNITION AND DEFENSE AGAINST INSIDER ATTACKS

Several studies have been as yet conducted to prevent from illegal accesses to systems. Mechanisms such as Intrusion Detection Systems or IDS, firewalls and Access Control are only capable of thwarting illegal entrances but not recognizing misuse of privileges by a legal user [5]. IDSs are one of those systems employed to spot intrusions, although they suffer three disadvantages [6], they: 1) Usually are able to detect pre-defined attacks and are of low efficiency against unknown ones. 2) Treat attacks in a low level i.e., they are not fit to understand the logic behind the scene. 3) Usually just send signals and no more in case of intrusion.

On the other hand, firewalls are designed and then implemented to protect from dangers threatening us from outside the system, however they are not equipped with any special feature to detect insider attacks. Nevertheless for use of such systems in order to thwart insider attacks, an inside firewall is employed which brings several constrictions upon the system.

Application of Access Control so as to keep sensitive information safe is a way of lowering threats but it causes obstacles too, for instance, it may have an undesirable impact on job satisfaction since employees are constantly discontent with their information accessibility [4].

A damaging insider is an employee or someone else who interacts with the system, has access to network, system or data and may disrupt reliability, integrity and approachability of the system intentionally or accidentally. Such act is considered inside attack.

Lots of attacks, throughout a year, occur on information systems bringing plenty of losses on organizations particularly those of information, finance or security. Unfortunately statistics prove that nearly most of damaging behaviors to the systems are carried out by a legal user [3, 7]. In Fig. 1, (a) and (b) show respectively a rate of electronic attacks performed by different agents in 2011 and a rate of attacks causing major losses to organizations [7]. The pie charts are suggestive of a careful consideration to this type of attacks.

In order to illustrate the issue we broach a couple of previous insider attacks: in the beginning of the 1996 two employees worked for a loan organization preparing loan reports for financial payments transformations. They were also authorized to applying modifications on loan reports according to up-to-date information which the organization received. They took advantage of their legal privileges so that the organization lost $215000. In March, 2002, a digital bomb destructed 10 billion files of the commercial systems to an international service company. The act was committed by an ex-employee whom had been already dismissed [3].

Damages caused by insider attacks had an upward trend so that U.S. National Security Agency since 2001, in order to resolve the issue of attacks, started making multi-dimensional attempts with the help of CERT to detect, evaluate and oversee potential threats to crucial systems and vulnerable data [3].

3. THE SUGGESTED APPROACH BASED ON ONTOLOGY

In this section, we first introduce the data structure employed in the approach and then define the framework suggested to detect insider attacks. The section is concluded with an example of performance.

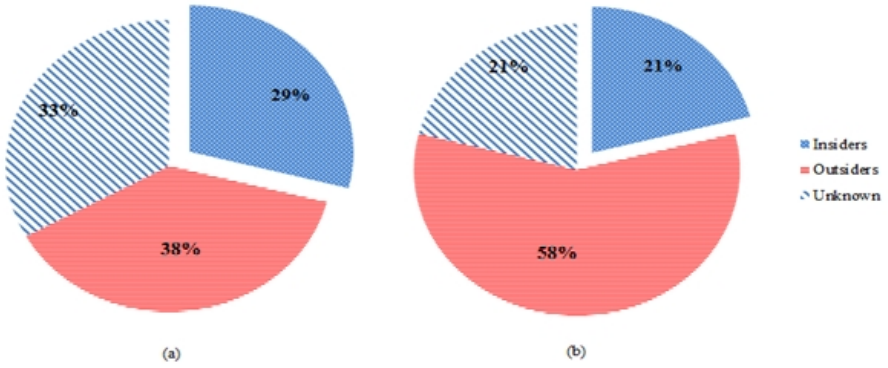

Figure 1. (a): A rate of electronic attacks performed by different agents. (b): A rate of attacks causing major losses to organizations in 2011.

3.1. The Used Ontology

In order to create the used ontology and present our model, we employ the rules in [8] to create ontology graphs by inserting essential elements to meet systems needs for presenting data. An ontology is generally shown through following factors [10]:

- Describing relationship between classes (concepts).
- Describing relationship between instances.
- Describing relationship between classes, instances and attributes.

So we define ontology as a definite set of O according to equation (1). We also use the set of R in equation (2) so as to define relationships.

$$O = \{o_1, o_2, \ldots, o_n\} \quad (1)$$

$$R = \{\text{isA, synOf, partOf, atr, val}\} \quad (2)$$

O, in equation (1), is a set creating the ontology of a database. o_i is employed ontology in database.

In equation (2) "isA" to describe relationships including inheritance between concepts, "synOf" to describe a set of all current synonyms for a concept, "partOf" to describe relationships of type Aggregation (relationships in which a concept is a subpart of a global concept). Although by omitting global concept subparts would not be deleted. For instance, imagine the relationship between a football player and a club. By deleting club, nature of players to it would not be deleted because they can be recruited in another club. "Atr" and "Ins" are used to respectively represent presentation of attributes to a concept and having access to values of a concept.

Each $o_i \in O$ and $r_i \in R$ are considered respectively a concept and relationship between two concepts. An ontology is defined as a graph G(V,E) [8] which V is a definite set of vertices and E is a definite set of edges. Every vertex is tagged with a concept and every edge is a connection between two vertices.

Each tag for node $n \in N$ is defined by $N(n) = o_i \in O$ mapping each node onto a string of o_i. Each edge $e \in E$ is tagged through a function $T(e)$ mapping each edge onto a string of R. Edges tagged as "atr" point to a head of link list including set of attributes of a concept so that values of concepts are approachable through tracing nodes with tag "val". The other nodes point to current

concepts in ontology. To illustrate the issue let's take a look at Fig. 2 which is part of products ontology. Finally a database in form of equation (3) is defined:

$$DB = \{G(V,E), O, R, N, T\} \quad (3)$$

One advantage of such graph is being simply updated. In case of inserting a new tuple into a table or deleting it, it is enough to modify the linked list connected to the table values. Such task is feasible at low cost and time consumption.

In order to compare two graphs and discover similarities between them, we apply the method [1] which we employed to reveal similitude in ontology graph to perform alignment operation. To do so, according to the method, we must take these three factors into account:

(I) Structural similarity.

(II) Semantic similarity.

Structural similarity generally examines graphs structure regardless to nature of concepts. Semantic similarity yields the number of namesake tags in total tags of ontology graph.

While many methods have been put forward to calculate each of the factors, we yet employ those in [9, 11, 12] with a couple of slight modifications for adjustment to defined concepts of ontology.

In order to structural similarity we define equation (4) as follows:

$$Structure(o_1, o_2) = \frac{num\_sim\_c + num\_sim\_p}{\max(num\_c1 + num\_p1, num\_c2 + num\_p2)} \quad (4)$$

Where $c \in$ {set of classes}, $p \in$ {set of attributes}, $c_1, p_1 \in O_1$ and $c_2, p_2 \in O_2$ so that O_1 and O_2 denote respectively the set of ontologies indicating current behaviors and the set of ontologies of user's behavior. Furthermore num_c_1 refers to the number of current concepts (classes) in O_1 containing sub-concept num_p_1 refers to the number of current attributes in O_1 containing sub-attributes and so are num_c_2 and num_p_2 defined. In order to determine num_sim_c and num_sim_p is practiced as (5) wherein $length\_root\_x_i$ and $length\_leaf\_x_i$ are respectively distances from the root and the leaf.

After discovering structural similarity, (6) is applied so as to calculate semantic similarity wherein $|C_1|, |C_2|, |P_1|$ and $|P_2|$ are numbers of total concepts and attributes in ontologies. num_equal_c and num_equal_p are also ,in order, numbers of similar tags among current concepts and attributes in two ontologies.

```
Begin
   x = p or c
   num_sim_x =0
   insert all x ▫ o1 in x1 and x ▫ o2 in x2
   while x1 and x2 is not empty do
      if (lengh_root_x_i is equal to lengh_root_x_j and lengh_leaf_x_i is equal to
lengh_leaf_x_j)
         num_sim_x = num_sim_x +1
         remove x1_i and x2_j from x1 and x2
      end if
```

```
   end while
End                                                    (5)
```

$$Lable(o_1, o_2) = \frac{num\_equal\_c + num\_equal\_p}{\max(|c1| + |p1|, |c2| + |p2|)} \quad (6)$$

After discovering semantic and structural similarity factors between user's behavior ontology and each ontology related to current behavioral patterns, the pseudo code (7) is employed so as to find the most similar pattern:

```
Begin
   for each (oi, oj) ∈ (O1, O2)
      add decision vector < Structure(), Lable() > to dv
      insert d0 in p and remove it from dv
 for each di in dv do
    if Dominate(p, di) do
       p = di
       remove di from dv
       continue
    else remove di from dv
  end for
      // p is pattern that is most similar to user behavioral pattern
End                                                    (7)
```

In algorithm (7), O_1 and O_2 symbolize the set of ontologies indicating current behaviors and the set of ontologies of user's behavior in order. Function Structure() returns a numeric value for structural similitude regarding equation (4) and function Label() returns a numeric value for semantic similitude regarding equation (6). Function Dominate(p, d_i) examines with respect to Pareto domination notation [10] if decision vector p excels decision vector d_i. In other words if decision vector p dominates decision vector d_i. For more information about Pareto optimization and the used algorithm you may refer to our previous study on alignment of ontologies in [1].

3.2. The Suggested Framework

The suggested framework consists of three phases of construction, comparison and analysis. Through construction phase, requisite information is collected and the ontology graph is constructed and then sent to the comparison phase. Through this phase, the degree of similitude between user's behavioral ontology and current patterns are measured and results are sent to analysis phase. In analysis phase it's determined whether the behavior is normal or is an attack on information sources of organization, however intentionally or accidentally. See Fig. 3. We describe the phases in detail as follows:

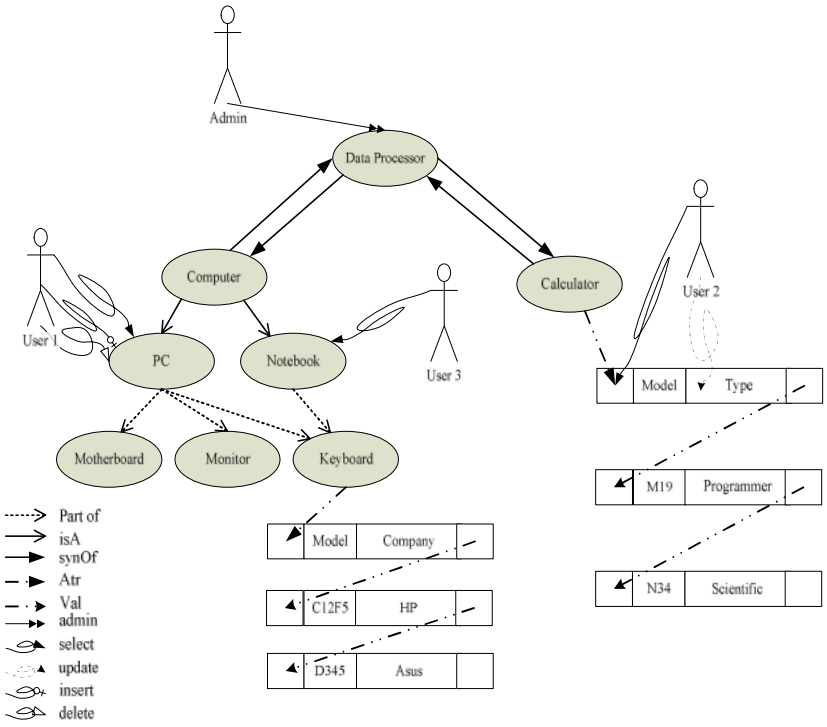

Figure 2. Ontology graph pertaining to parts of information belonging to computer equipment distribution firm products alongside the way four users communicate with different parts of the ontology.

During construction phase the ontology related to user's behavior and his legal access levels to information is created by means of observing user's behavior and his degree of authorization which is received by Access Control. In our system a user's access is defined this way: user is permitted to read tables data. The operation is called select. Operation of update, insert and delete are respectively defined as modifying a record in a table, adding a new record into table and removing a record from a table. Admin, or administrator, is someone else who is authenticated and authorized to perform all four operations on a table.

In Fig. 2 four examples for user's communication with part of the ontology related to products of a computer equipment distribution firm is brought in form of an ontology graph. According to the Fig. 2, admin is authorized to read data in Calculator class or to modify them as well as to insert a tuple into the table of the class and to delete a tuple from it.

i. User has all admin's authorization except for modification.

ii. User is only capable of reading or modifying data in the field TYPE.

iii. User is only capable of reading data in the field MODEL not any other operations.

After constructing user's behavior ontology, his certain request is transferred to Data Ontology (an ontology containing all requisite information in database). After verifying the request, the database response to user's behavior is given in form of ontology. In the end, the final ontology O resulted in from user's behavior, user's access level and database response is created.

In comparison phase, the ontology O is compared with current normal behavioral patterns in Normal Patterns Ontology and Attack Patterns Ontology. Such comparison is made according to the algorithm at the end of section of III part A. The degree of similarity between the ontology O and current patterns is transferred to analysis phase. The outcome of this phase is of two parts: the

first indicates to the similitude degree between the ontology O and normal behavioral patterns. And the second indicates to the similitude degree between the ontology O and current patterns in insider attacks.

In the analysis phase, it is determined if user's behavior and the corresponding system response is a potential attack? In case of being verified as a normal pattern, it is transferred to the update unit in order to update the unit of Normal Patterns Ontology (only those parts of patterns are registered that never before have).

If a user's behavior is recognized as an attack a message in accord with the level of the attack is signaled to the administrative system by the Alert System. Moreover, this pattern is transferred to the unit of Attack Patterns Ontology so as to update attack patterns. See Fig. 3.

Since sequences of user's behaviors are registered in the system, such serial records may be used, in case of an attack, to unveil if it is set to steal information or aimed at interruption so it is easier to troubleshoot current security difficulty and make the system safer against future attacks.

4. CONCLUSIONS

In this paper we attempted to offer a new method based on ontology graph in order to detect insider attacks in information system. To do so, we first put a structure forward to present data based on ontology. Furthermore, we used a method based on Pareto optimization in order to discover similitude degree of user's behaviors; we then introduced a framework based on user's behavioral patterns and with the help of ontology to detect insider attacks on the system.

By our method, not only is it possible to detect potential attacks on the system but we realize the logic behind the scenes and data streams brining on these behaviors such that they come to our aid to make the system more secured in the future.

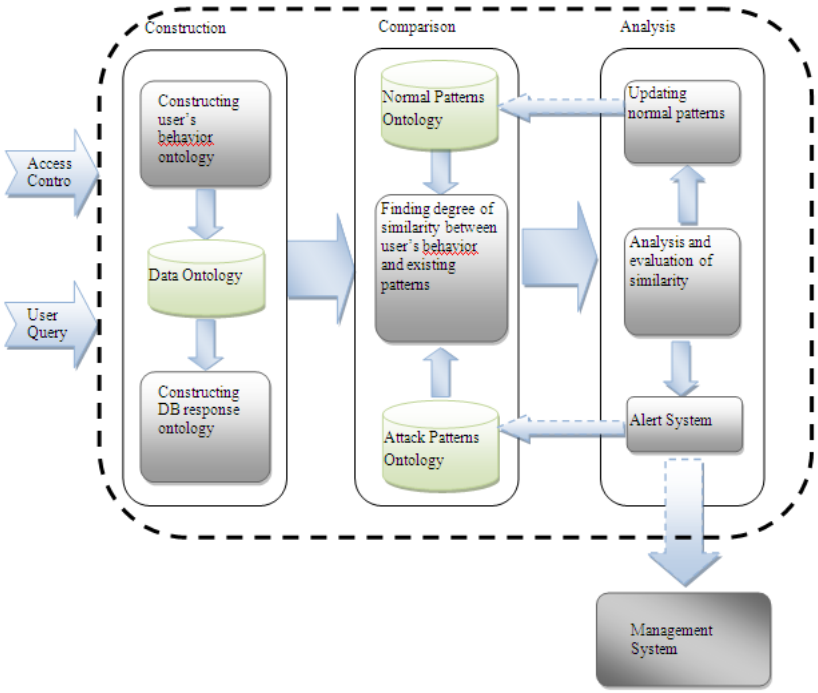

Figure 3. The suggested framework to detect insider attacks on information sources consisting of three phases of construction, comparison and analysis. It first receives user's requests and his legal access level and in case of detecting an attack it signals an appropriate message to the administrative system of the organization.

REFERENCES

[1] R. Asgari, M. Moghaddam, S. Sahraei, R. A. Ebrahimi, "*An Approach to Ontologies Alignment Based Upon Pareto Front*", 5th International Computer and Instructional Technologies Symposium, Turkey, 2011.

[2] N. Kanaskar, J. Bian, R. Seker, M. Nijim, N. Yilmazer, "*Dynamical System Approach to Insider Threat Detection*", IEEE International Systems Conference (SysCon), pp. 232-238 , 2011.

[3] M. R. Randazzo, M. Keeney, E. Kowalski, "*Insider threat study: Illicit Cyber Activity in the Banking and Finance Sector*", CMU/SEI-2004-TR-02 Key: citeulike:7861905 , 2005.

[4] D. Caputo, G. Stephens, "*Detcting insider theft of trade sectors*", curity & Privacy, IEEE , Volume 7 , pp. 14-21, 2009.

[5] S. Mathew, S. Upadhyaya, D. Ha, H. Ngo, "*Insider Abuse Comprehension Through Capability Acquisition Graphs*", 11th International Conference on Information Fusion, pp. 698-705, 2008.

[6] C. Colwill, "Human factors in information security: The insider threat e Who can you trust these days?", Information Security Technical Report, pp. 186-196, 2011.

[7] CERT, "*CyberSecuirtyWatch Survey*", CSO Magazine, U.S. Secret Service, Software Engineering Institute CERT Program at Carnegie Mellon University and Deloitte, January 2011.

[8] C.B. Necib, J. Freytag, "*Ontology based query processing in database management systems*", Proceeding on the 6th international on ODBASE, pp. 37-99, 2003.

[9] A. Mazak, M. Lanzenberger, B. Schandl, "*iweightings: Enhancing Structure-based Ontology Alignment by Enriching Models with Importance Weighting*", International Conference on Complex, Intelligent and Software Intensive Systems, pp. 992-997, 2010.

[10] K. Deb, "*Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design*" Kanpur Genetic Algorithms Laboratory (KanGAL), Depatment of Mechanical Engineering, Indian Institute of Technology, Kanpur, India, 1999.

[11] L.Juanzi, J. Tang, Y. Li, Q. Luo, "*RiMOM: A Dynamic Multistrategy Ontology Alignment Framework*", IEEE transactions on knowledge and data, Vol. 21, pp. 1218- 1232, 2009.

[12] C. Xie, "*Semantic Similarity-Based Ontology Alignment for Enterprise Ontologies*", 6th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 386-390, 2009

Authors

Reza Asgari Reza Asgari was born in Iran, Ghazvin. He recieved his BSc degree from university of Guilan, Iran in 2011. He is now MSc student at university of Guilan, Iran. His research interests in operating system and database security.

Reza Ebrahimi Atani Reza Ebrahimi Atani received his BSc degree from university of Guilan, Rasht, Iran in 2002. He also recieved MSc and PhD degrees all from Iran University of Science and Technology, Tehran, Iran in 2004 and 2010 respectively. Currently, he is the faculty member and assistant professor at faculty of engineering, University of Guilan. His research interests in cryptography, computer security, network security, information hiding and VLSI design.