Intent-driven insider threat detection in intelligence analyses

Eugene Santos, Jr.[*], Hien Nguyen[+], Fei Yu[*], Keumjoo Kim[*], Deqing Li[*], John T. Wilkinson[*], Adam Olson[+], and Russell Jacob[+]

[*]Dartmouth College
Thayer School of Engineering
8000 Maclean
Hanover, NH 03755
Eugene.Santos.Jr@Dartmouth.EDU

[+]University of Wisconsin-Whitewater
Dept. of Math and Computer Science
800 W. Main Street
Whitewater, WI 53190
nguyenh@uww.edu

Abstract

In intelligence analysis, information plays an important role in solving problems and making decisions. However, an increasing number of malicious behaviors, most of which come from insiders, are threatening information security. An insider in the intelligence community is either a current or previous member who has access to privileged resources and whose intelligence decisions have impacts on the decision makers. With malicious intent, an insider may alter, fabricate, or hide critical information in order to interfere with a decision making process, In this paper, we focus on detecting abnormal behaviors that can lead to identifying a malicious insider. Malicious actions such as disinformation tend to be very subtle and thus difficult to detect. Therefore, we employ a user modeling technique because its dynamic and incremental nature provides unique capabilities to reflect the cumulative effects from malicious actions. We created a computational model for each insider and applied our detection technique to monitor and analyze the user models as they change over time. By monitoring the changes we can be alerted to any deviation of behavior. A pilot test revealed that the deviations had a high correlation with analysts' cognitive styles. Based on this finding, we designed a framework that minimized the impacts from cognitive styles. The evaluation showed that four out of five simulated malicious insiders were successfully detected.

1. Introduction

An insider in the intelligence community is either a current or previous member who has access to privileged resources and whose intelligence decisions have impacts on decision makers. A typical decision making process involves the following steps: First, insiders are assigned to carry out an intelligence analysis on given problems. Next, they collect and analyze information in order to organize and support hypotheses. At the end, every insider submits a final report that impacts decision making directly or indirectly. In the decision making process, any malicious insider may carry out insider attacks that result in irreversible damage. To date, much effort has been focused on detecting cyber insider threats [1,5]. However, little effort has been placed on automatically detecting the malicious insiders who attempt to interfere with a decision making process.

We observe that there are quite a few challenges in preventing and detecting these insider threats. The first challenge is the variety of cognitive styles among analysts. Even though analysts are usually trained to carry out their analyses in a fairly rigid fashion [2], previous research has shown that analysts have their unique cognitive styles in achieving goals [11]. Their cognitive styles are reflected both in the way they carry out their analyses and in their final deliverables such as reports. Any deviation caused by different cognitive styles should be accepted as legitimate variations. Therefore, a detection methodology should account for unique cognitive styles in achieving analysts' goals, and different analytical styles in retrieving information as well as different reporting habits. Another challenge is that the detection method should not be constrained by the way an insider carries out a malicious attack. For example, a malicious insider can spread disinformation simply using his intelligence reports without breaking into any system or stealing any critical information. This kind of the malicious actions are subtle in nature and difficult to model. Furthermore, any detection method based on a priori taxonomy of malicious behaviors is vulnerable to attacks that it has not accounted for. The third challenge is the inappropriateness of using historical information as a baseline to detect abnormal behaviors. Unlike jobs in other disciplines, the assigned problems to analysts are rarely repetitive. The data that an analyst accesses varies from task to task, thus past history cannot form a sound reference to detect deviations of behavior.

We propose a unified framework for intent-driven insider threat detection. The heart of the framework is the IPC user modeling technique [7,9,10] which captures the analyst's interests, knowledge context, and preferences over time. Another key element is an insider detection methodology that analyzes the user models and sends warning signals when an insider is suspected.

We conducted an empirical evaluation using the APEX '07 collection. The APEX dataset was created by the National Institute of Standards and Technology

(NIST) to simulate an analysis task in the intelligence community. The APEX '07 collection included 8 analysts, their recorded actions over time, and their research reports as well as assessment reports generated on their analysis. Five malicious insiders were simulated each based off of one of the original 8 analysts. We measured the similarities between the final user model and different hypotheses in the assessment reports for all analysts. In order to analyze these similarity values, three different metrics were proposed to compare the deviation values between multiple hypotheses either in each section of the assessment report or in the entire report for identifying suspicious insiders. The experimental results showed that the framework was effective in identifying insider threats. The first and third metrics detected four insiders with malicious intent. The third metric did not raise any false positives while the first and second metrics had false positives on two benign analysts.

The remainder of this paper is organized as follows: We introduce some previous research in detecting insider threats. Next, we detail the proposed methodology in Section 3. The design of our experiment is described and the experimental results are analyzed in Section 4. Finally we conclude with description of our future work.

2. Background

Traditional insider threat detections have focused on three main methods: action analysis, social networks and document analysis. In action-based detection, the actions of the analyst are monitored until an irregular action falls outside of an analyst's pre-defined role. With social networks, a static social network is created beforehand for an analyst and this network is used to detect whether an analyst is making abnormal behaviors. A document-based method focuses on the documents that an analyst accesses or produces. These methods are not able to detect insiders until they make an irregular action, and the detection parameters usually have to be determined and input beforehand, manually. This means that these methods are often too slow or too late to capture the malicious insiders as they rely on static information:

Natarajan and Hossain (2004) and Park and Ho (2004) use social networks to detect insider threats. In the former research, by creating a social network for not only an analyst, but also of the system around them, one can detect abnormal behaviors by monitoring their actions. The latter focuses on both a social network of an analyst and the actions they make. Roles are created for an analyst a priori, which are system permissions related to the type of work they are assigned. The actions that the analyst takes in every role are logged, and compared to a social network of their expected behavior. Assigning a role to an analyst may assist in catching insider actions since there is an additional check that can be made. However, both roles and social networks must be created manually beforehand and is not very flexible. If a piece of information is added or removed, the social network must be completely redone. If an analyst takes a malicious action that is not considered when the network was created, he will not be caught. Likewise, if an analyst has access to the software that monitors him, he may be able to hide his actions. Moreover, detecting insider threats by using social networks is difficult to implement as the differences between two analysts do not explicitly mean either is an insider, only that they are different from each other. Efforts to automatically separate analysts can identify which ones do research differently, not the validity of their data.

Symonenko et al (2004) attempts to validate the data an analyst accesses. The documents obtained when a group of analysts are working on the same project are processed using natural language processing techniques and clustered afterwards. The clustered documents show how similar the documents an analyst views are to the documents the other analysts access. The research on the documents themselves, however, does not factor in an analyst's personal variations. Like with the role-based method in [8], the focus in Meza-Aleman et al (2005) is on determining how legitimate the document access is with respect to the access given to the analyst. The analyst role is taken into account, but it is assumed that the information on the analyst has been built beforehand.

In our approach, as an analyst works, we build up their user model, which can be referenced later or even as it is being built. We assume that even though analysts may vary in their experience and expertise, they all share the same role and access rights. They all have access to the same data set and their reports are weighted equally. This is our main difference from Thompson (2004) who uses sets of user models for each analyst. While he does recognize the need to focus on an analyst's actions, documents viewed and the content of their produced work, their model is based on having pre-compiled information available to build models.

Our work focuses on dynamically detecting insider threats in a way that is not constrained by the way an analyst works or the data an analyst handles. The user modeling techniques we use provides unique capabilities in recognizing various classes of insider threats.

3. Methodology

3.1 Intent-Driven Insider Threat Detection

We propose an intent-driven framework to detect insider threats in the intelligence community. The framework consists of a user model and insider detection metrics. It begins with collecting textual observables from search queries, information content accessed and saved, and final deliverable reports. All textual observables are categorized into behavioral observables and decisional observables. Behavioral observables can be any information an analyst has accessed when he/she was performing the analysis task, while the decisional

observables can be analytical decisions, supporting arguments and evidence that an analyst has concluded about the analytical task. Next, a user model is built based on the sequential order of the behavioral observables. A user model is a computational model that allows us not only to detect the change of a user's behavior over time but also to predict a user's conclusion. Since such predictions are able to anticipate future decisions, they can also be used to detect deviations from the final decisions. Based on this nature of user modeling, we first conducted a pilot test to investigate how the deviation of user models from final reports is correlated with the validity of the analysts' data. . Last but not least, we aim to identify the main factors of the deviations. Our main hypothesis is to detect the malicious insiders by finding out whose deviation from their own final conclusions are beyond a given threshold. Each insider's deviation value is compared against all other insiders in order to distinguish suspicious/malicious ones. A simple deviation to use in the test is obtained by computing how similar the final user model is with the decisional observables of an insider. The test results reveal that the computed deviation values have a high correlation with analysts' cognitive styles. Without normalizing the effects from different cognitive styles, the malicious insiders are not distinguishable from benign insiders due to these influences. Based on the findings in the pilot test, we designed an insider detection technique to normalize the deviation by different cognitive styles. The similarities between the final user model and the different hypotheses in the decisional information are computed. Different metrics are used to further analyze these similarities. The details of the design of the pilot test and the detection method as well as their respective experiment results can be found in Section 4.

3.2 User Models and Document Graphs

Our user models were based on the IPC model [7,9] which contain an interest list, a preference network and a context network. An interest list is a list of important terms with associated weights reflecting a user's current interests. A preference network is a Bayesian network that keeps track of how a user forms queries. A context network is a document graph used to model a user's knowledge context. Old knowledge will be faded out in the context network once it is not referred for a while. How a fading method works is described as follows: When nodes are added to a context network, we keep track of the number of times that they occurred. If the frequency is under a threshold after a few iterations of being in the context network then the nodes will be removed from a context network.

We generate a document graph (DG) for each document from the textual deliverables. A Document Graph (DG) is a directed acyclic graph consists of concepts nodes and relations between them. Two kinds of relationships between conceptual nodes are defined -- "Is a" relations and "Related to" relations [14]. An "Is a" relation is a generalizing relation between parts of a noun phrase. A "Related to" relation is between adjacent noun phrases in a sentence and between noun phrases and the noun phrase portion of a prepositional phrase. The parsing for the sentence "Military members support nuclear weapons." is depicted in Figure 1.

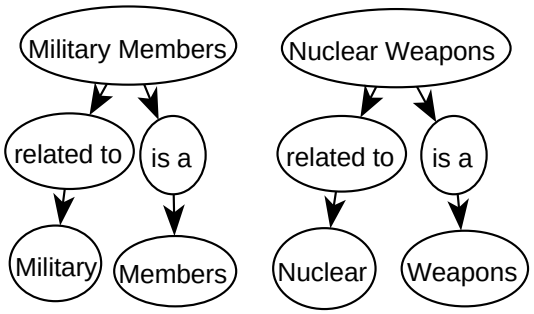

Figure 1 An example of a Document Graph

We use user-centric user models to track analyst intent and generate DGs for all of the documents an analyst looked at. The generated documents are stored for future diagnosis. The method we use to compare two document graphs is modified from Montes-y-Gómez et al. [4]: checking for a sub-graph of one DG in another DG (Figure 2). This method gives us similarities between 0 and 1, 1 meaning completely similar and 0 meaning not similar at all.

$$Similarity(DG_1, DG_2) = \frac{n}{2N} + \frac{m}{2M}$$

Figure 2 Equation to compare a pair of DGs. n denotes the number of entity nodes shared by DG1 and DG2. N denotes the total number of entity nodes in DG1. Likewise, m and M are parallel to n and N except they count the relation nodes instead of entity nodes.

4. Experimental procedure

4.1 Objectives

The main objectives of the experiment are to:

1) Investigate the effects of predicting insiders' intent by modeling their behaviors
2) Evaluate the effectiveness of the framework to detect malicious insiders
3) Identify future directions and improvements

4.2 Testbed

To conduct the experiments, we used the APEX '07 data set collected by the National Institute of Standards and Technology (NIST), which was originally used to evaluate Collaboration and Analyst/System Effectiveness (CASE) tools for tacit collaboration.

Eight analysts were involved in this experiment and each was requested to assess the following two hypotheses: "Where does the Iranian clerical community stand on Ayatollah Khamenei and President Ahmadinejad's policies with regards to Iran's civilian and military nuclear program?" (We refer to this as hypothesis 1 or H_1) and "Are there fissures in the clerical community and do they represent a deepening divide among the clerics loyal to the Iranian revolution?" (We refer to this as hypothesis 2 or H_2).

The collected data for each analyst is follows:

1) Analysis Log Events (ALEs) that contain recorded information about search events, retain events, save/delete events and so on.
2) An assessment report that includes an analyst's arguments, supporting evidence that is organized in an ACH matrix, and analytical decisions.
3) A research report that answers scripted questions.

It is important to mention that each of the analysts are required to write their assessment report using Analysis of Competing Hypothesis (ACH) [2], which involves detailed arguments of his analysis decision, and matrices with evidence and associated ratings.

Five malicious insiders were simulated each individually based on one of the original analysts. The malicious insiders may alter, fabricate or hide critical information in order to interfere with a decision making process. Two out of five are novice malicious insiders. They are constructed according to a scenario that each novice insider attempts to fabricate evidence supporting one of the two hypotheses when writing their assessment reports. Their malicious behaviors are overt so that they are easier to detect by human judges. The other three are "expert" malicious insiders whose malicious actions are more subtle to detect. Some examples of the malicious actions that an expert malicious insider may carry out are as the follows: when searching for information, the insiders may use more support queries than nonsupport queries and put more constraints on nonsupport queries; when retaining information, the insiders may ignore nonsupport documents, use outdated documents when support documents are not sufficient, and even fabricate snippets of information; when drafting reports, the insiders may over cite the same documents, exaggerating evidence ratings.

All of these different malicious actions are used as guidance for simulating a malicious insider but do not limit what malicious actions can be carried out.

4.3 Procedures and Results:

4.3.1 A Pilot Test

A pilot method was used to compare an analyst's user model with his analysis decision. First, we built a series of user models based on the sequential actions of an analyst. Next, the analyst's assessment report was converted into a document graph which represents the semantic meaning of the report. Lastly, we compared the analyst's final user model with the DG of his assessment report. The similarity values obtained from the experiment are presented in Table 1. Each analyst is given a code name and the code names of the malicious insiders are simply extended from the names of their original analysts.

Analyst	Similarity value
APEXK	0.136016946
APEXKEXPERT	**0.136073091**
APEXE	0.145556773
APEXHNOVICE	**0.212040946**
APEXP	0.344164617
APEXL	0.345929973
APEXH	0.499434069
APEXFNOVICE	**0.51028939**
APEXPEXPERT	**0.534264547**
APEXFEXPERT	**0.581233332**
APEXC	0.761673709
APEXF	0.890060268

Table 1 Similarity values computed by comparing the final user models with the assessment reports.

A user's behaviors are modeled from recorded actions using our user modeling technique. The decisional observables such as assessment reports can be used to model a user's expected outcomes of the behaviors. Our pilot hypothesis is that a malicious insider's similarity value between his user model and the assessment report should be lower compared to a benign insider's, indicating a higher deviation of behaviors from expected outcomes. However, according to the sorted similarity values in Table 1, the malicious insiders are not distinguishable from the benign insiders. Therefore, the hypothesis is rejected. For example, the similarity of APEXK is only 0.136016946 which is the lowest among the similarities of all analysts. It means that the final user model of APEXK is extremely dissimilar with his assessment report. However, APEXK is not a malicious insider whose similarity value is expected to be high according to our hypothesis. On the other hand, the similarity value of APEXFEXPERT is **0.581233332,** which means that his user model is highly similar with his assessment report. However, his similarity value is expected to be low because he is a malicious insider.

A further analysis into the dataset reveals some of the reasons. One reason is that the unique cognitive styles of analysts have a large impact on how well their user models can predict their decisions. An analyst's cognitive style is a preferred way of perceiving, reading,

researching and writing. The uniqueness of cognitive styles is reflected in the followings:

1. Different analysts demonstrate different styles in retrieving information. Some analysts like accessing many documents before saving a snippet as evidence while others do not. Because our user model technique incorporates a fading facility which fades out old knowledge if it does not appear frequently enough, the order of actions make a difference.
2. Different reporting habits. The most influential factor is the different styles in citing evidence. Some analysts tend to quote a snippet out of the cited documents directly in the ACH matrix while others tend to summarize their own opinions in the matrix. What's more, some analysts tend to write a report directly from their past experiences which cannot be found in the recorded actions, while others prefer to justify their opinions just using what they found.
3. Different sizes of ALEs, different lengths of the assessment reports, and so forth.

We have also observed that there is a strong correlation between a malicious insider with his basis analyst. For example, the similarity between APEXK and APEXKEXPERT is almost 1 which indicates that the cognitive styles of APEXK are preserved in APEXKEXPERT. In this case, the influence from the cognitive style on the similarity value is much stronger than variance due to malicious actions. It can be explained intuitively because only a few malicious actions occurred while the cognitive style affects all actions.

4.2.2 Insider Detection Method

Based on the finding in the pilot test, a detection method was designed along with three different metrics to detect malicious insiders. The method takes inherent cognitive styles into account to ensure that the deviation is mostly reflected by malicious actions. The main procedures can be described as following.

Step 1: Decompose each assessment report into four components, which are arguments on Hypothesis 1, arguments on Hypothesis 2, evidence cited about Hypothesis 1 in the ACH matrix, and evidence cited about Hypothesis 2 in the ACH matrix.

Step 2: Convert the four components into DGs.

Step 3a: For each analyst, compute the similarity between each of the four DGs and the final user model.

Step 3b: For each analyst, also compute the similarity between the entire assessment report and the final user model.

Step 4: Normalize the similarities computed in Step 3a by dividing them by the similarity computed in Step 3b. The target of Step 3 and 4 is to minimize the influences from different cognitive styles in different components by normalization.

Step 5: Apply the metrics below on the similarity values to capture insiders.

Metric 1: Compute the deviation in each analytical stage. One stage is to write up arguments assessing two hypotheses and the other is to construct an ACH matrix.

A formula to compute the difference between Hypothesis 1 and Hypothesis 2 in the analytical stage of writing up arguments is as following:

$$Deviation_{arg}(H_1,H_2) = Sim_{arg}(H_1) - Sim_{arg}(H_2)$$

where $Sim_{arg}(H_1)$ is obtained by computing the similarity value between the final user model and the arguments.

Our hypothesis is that if arguments on Hypothesis 1 largely deviate from arguments on Hypothesis 2, or if evidence cited about Hypothesis 1 largely deviates from evidence cited about Hypothesis 2, then the analyst can be identified as a malicious insider.

Analyst	$DEVIATION_{ARG}(H_1, H_2)$	$DEVIATION_{ACH}(H_1, H_2)$
APEXKEXPERT	**0.32758599**	0.150971083
APEXK	0.04445526	-0.000714383
APEXC	-0.06463396	-0.132235344
APEXHNOVICE	**-0.61152796**	**-1.295942635**
APEXH	0.10343892	-0.052325706
APEXL	0.0425942	**0.352047403**
APEXE	0.063317	-0.081047
APEXP	-0.16347768	**0.282958345**
APEXPEXPERT	0.18957924	-0.096895267
APEXFNOVICE	**0.57762307**	-0.019518343
APEXFEXPERT	**0.34366445**	**0.566688317**
APEXF	-0.00934664	0.034651211
Threshold	**0.21177036**	**0.255499586**

Table 2 Deviations between Hypothesis 1 and Hypothesis 2 in different analytical stages

The threshold is simply an average value of the absolute values of all deviations which will be further explored in future work. All deviation values that exceed the threshold are bolded in Table 2. The results show that a high deviation is detected for four malicious insiders in either of the two analytical stages. The deviation values from novice malicious insiders tend to be higher than those from expert malicious insiders. This is intuitive because the malicious actions carried out by expert malicious insiders are assumed to be more subtle thus more difficult to capture. Another observation is obtained by comparing the deviation values from APEXE against the pilot test. In the pilot, APEXE had a similarity value of 0.145556773 which is in the top 3 indicating a large deviation. From the results obtained in this method, both deviation values from APEXE are extremely low (0.063317 and -0.081047). This supports our hypothesis

that inherent cognitive styles have a large impact on the experiment. The results also demonstrate that the impacts are minimized using the normalization technique (Step 3a/b).

Metric 2: This metric is extended from Metric 1 to further examine whether two deviation values in different analytical stages also indicate abnormality. If the difference between arguments on Hypothesis 1 and arguments on Hypothesis 2 is highly different from the difference between evidence cited about Hypothesis 1 and evidence cited about Hypothesis 2, then the analyst is identified as a malicious insider.

A formula to compute the deviation between Hypothesis 1 and 2 between two analytical stages is

$$\text{Deviation (Arg, ACH)} = \text{Deviation}_{arg}(H_1,H_2) - \text{Deviation}_{ACH}(H_1,H_2)$$

where $\text{Deviation}_{arg}(H_1,H_2)$ and $\text{Deviation}_{ACH}(H_1,H_2)$ are values computed in Metric 1.

Analyst	*DEVIATION(DEVIATION$_{ARG}$, DEVIATION$_{ACH}$)*
APEXKEXPERT	0.176614904
APEXK	0.045169643
APEXC	0.067601386
APEXHNOVICE	**0.684414672**
APEXH	0.155764622
APEXL	**-0.309453203**
APEXE	0.144364
APEXP	**-0.446436028**
APEXPEXPERT	**0.286474507**
APEXFNOVICE	**0.597141416**
APEXFEXPERT	-0.223023867
APEXF	-0.043997855
Threshold:	**0.265038**

Table 3 Deviations between different analytical stages

As shown from the results in Table 3, Metric 2 does not perform as well as Metric 1. Metric 2 also identified the same two benign analysts APEXL and APEXP as malicious insiders. However, Metric 2 was only able to detect three out of five malicious insiders. One possibility for the poorer performance is that a certain amount of deviation from one analytical stage to another should be accepted as legitimate variation.

According to our observations obtained from the first method, the legitimate deviation should also be different due to different cognitive styles.

Metric 3: Based on the observations from Metric 2, a new metric is designed to examine whether the difference between everything written about Hypothesis 1 and everything written about Hypothesis 2 is too high.

Our hypothesis is that if the combination of arguments and evidence on Hypothesis 1 largely deviates from the combination of arguments and evidence on Hypothesis 2, then the analyst is identified as a malicious insider.

A formula to compute the deviation between hypothesis 1 and 2 is as follow:

$$\text{Deviation } (H_1,H_2) = (\text{Sim}_{arg}(H_1) + \text{Sim}_{ACH}(H_1)) - (\text{Sim}_{arg}(H_2) + \text{Sim}_{ACH}(H_2))$$

Analyst	*DEVIATION(H_1, H_2)*
APEXKEXPERT	**0.239278535**
APEXK	0.021870438
APEXC	-0.098434651
APEXH NOVICE	**-0.953735299**
APEXH	0.025556605
APEXL	0.197320801
APEXE	-0.008864863
APEXP	0.059740331
APEXPEXPERT	0.046341987
APEXFNOVICE	**0.279052364**
APEXFEXPERT	**0.455176384**
APEXF	0.012652283
Threshold:	**0.199835**

Table 4 Deviations between hypothesis 1 and hypothesis 2 on all analytical stages

Compared to the results of Metrics 1 and 2, Metric 3 performs the best. According to the results in Table 4, four out of five malicious insiders are identified but none of the benign analysts are suspected. The results support our hypothesis that the computed deviation from one analytical stage to another is influenced by inherent deviations of analysts. This influence is canceled out when we compute a deviation without splitting the content into different analytical stages.

5. Conclusions

The malicious actions that are carried out to spread disinformation in the intelligence community are subtle in nature. In addition, a series of such actions are usually required to cause damage. Little research has been done that develops an approach as well as evaluates its effectiveness. In this paper, we propose a unified framework which builds user models for all of the analysts. The dynamic and incremental nature of user models provides unique capabilities to reflect the cumulative effects from the subtle malicious actions. A

key finding is that unique cognitive styles have large impact on computed deviations. Without minimizing the effects from different cognitive styles, the malicious insiders are not distinguishable from benign analysts due to these influences. An insider detection method was thus designed in order to solve this problem. The experiment results for the method showed that the derivation degrees are normalized by different cognitive styles. Our method performed well at identifying insiders without false positives.

In the future, we intend to pursue more extensive testing and we would like to examine the feasibility of normalizing out cognitive styles if the information is not all available such as ACH. Also, we wish to further expand our existing framework so that it is not only able to identify suspicious insiders but is also able to list the abnormal behaviors of the suspected insiders as evidence.

Acknowledgments. This work was supported in part by Air Force Office of Scientific Research, Grant No. FA9550-07-1-0050 and by a grant from IARPA.

References

1. Anderson, D.F; Cappelli, D.M.; Gonzalez, J.J; Mojtahedzadeh, M.; Moore, A.P.; Rich, E.; Sarriegui, J.M.; Shimeall, T.J.; Stanton, J.M.; Weaver, E. & Zagonel. A. (2004). Preliminary System Dynamics Maps of the Insider Cyber-Threat Problem. In *Proceedings of the 22nd International Conference of the System Dynamics Society.*

2. Heuer, R.J. (2001). *Psychology of Intelligence Analysis.* Center for the Study of Intelligence, Central Intelligence Agency.

3. Meza-Aleman, Boanerges; Burns, Phillip; Eavenson, Matthew; Palaniswami, Devanand & Sheth, Amit (2005). An Ontological Approach to the Document Access Problem of Insider Threat. *Lecture Notes in Computer Science*, Volume 3495. Pages 486-491.

4. Montes-y-Gómez, M., Gelbukh, A., and Lópes-López, A. (2000). Comparison of Conceptual Graphs. In *Proceeding of MICAI-2000, In 1st Mexican International Conference on Artificial Intelligence.*

5. Moore, Andrew P.; Cappelli, Dawn M. & Trzeciak, Randall F. (2008). The "Big Picture" of Insider IT Sabotage Across U.S. Critical Infrastructures. Technical Report CMU/SEI-2008-TR-009

6. Natarajan, Anand & Hossain, Liaquat (2004). Towards a Social Network Approach for Monitoring Insider Threats to Information Security. *Lecture Notes in Computer Science*, Volume 3073. Pages 501-507.

7. Nguyen, H.; Santos, E Jr.; Zhao, Q.; and Wang, H. (2004b). Capturing User Intent for Information Retrieval. *Proceedings of the 48th Annual Meeting for the Human Factors and Ergonomics Society (HFES-04), new Orleans, LA.* Pages 371-375.

8. Park, Joon S. & Ho, Shuyuan Mary (2004). Composite Role-Based Monitoring (CRBM) for Countering Insider Threats. *Lecture Notes in Computer Science*, Volume 3073. Pages 201-213.

9. Santos, E Jr.; Nguyen, H.; Zhao, Q. & Wang, H (2003a). User modelling for intent prediction in information analysis. *Proceedings of the 47th Annual Meeting for the Human Factors and Ergonomincs Society*. Pages 1034–1038.

10. Santos, E Jr, Zhao, Q, Johnson, G, Nguyen, H & Thompson, P. (2005b). A Cognitive Framework for Information Gathering with Deception Detection for Intelligence Analysis. *Proceedings of 2005 International Conference on Intelligence Analysis*.

11. Santos, E Jr.; Zhao, Q.; Nguyen H.; and Wang H. (2005a). Impacts of User Modeling on Personalization of Information Retrieval: An Evaluation with Human Intelligence Analysts. *4th Workshop on the Evaluation of Adaptive Systems, in conjunction with UM'05*. Pages 27-36.

12. Symonenko, Svetlana; Liddy, Elizabeth D. Liddy; Yilmazel, Ozgur; Zoppo, Robert Del; Brown, Eric & Downey, Matt (2004). Semantic Analysis for Monitoring Insider Threats. *Lecture Notes in Computer Science*, Volume 3073. Pages 492-500.

13. Thompson, Paul (2004). Weak Models for Insider Threat Detection. Proceedings of SPIE.

14. Zhao Q.; Santos Jr. E.; Nguyen H.; and Mohammed A. (2006). What Is Needed for a Good Summary?? Two Different Types of Document Sets Yet Seemingly Indistinguishable to Human Users. In *Proceedings of HICSS-39, IEEE Press. Maui, HI.*