Using a Graph-Based Approach for Discovering Cybercrime

William Eberle
Department of Computer Science
Tennessee Technological University
Cookeville, TN USA
weberle@tntech.edu

Lawrence Holder
School of Electrical Engineering &
Computer Science
Washington State University
Pullman, WA USA
holder@wsu.edu

Jeffrey Graves
Department of Computer Science
Tennessee Technological University
Cookeville, TN USA
jagraves21@tntech.edu

Abstract

The ability to mine data represented as a graph has become important in several domains for detecting various structural patterns. One important area of data mining is anomaly detection, but little work has been done in terms of detecting anomalies in graph-based data. While there has been some work that has used statistical metrics and conditional entropy measurements, the results have been limited to certain types of anomalies. In this paper we present a graph-based approach to uncovering anomalies in applications containing information representing possible cybercrime activity: network activity and employee movements. We use three algorithms for the purpose of detecting anomalies in all three types of possible graph changes: label modifications, vertex/edge insertions and vertex/edge deletions. Each of our algorithms focuses on one of these anomalous types and uses the minimum description length principle to discover those substructure instances that contain anomalous entities and relationships. We then show the usefulness of applying these graph theoretic approaches to discovering anomalies in a real-world-type domain, the Visual Analytics Science and Technology (VAST) mini-challenge involving badge and network traffic. In addition, we present the results of this approach on synthetic graphs of varying sizes, in order to demonstrate the applicability of this approach as a real-world application.

Introduction

Cybercrime is one of the leading threats to company confidential data and resources. A recent study by the Ponemon Institute surveyed 577 IT practitioners, who rated the issue of cybercrime as the top trend in their industry for the next few years, over such hot topics as cloud computing, mobile devices, and peer-to-peer sharing [Ponemon 2009]. The U.S. Department of Justice, in its Computer Crime & Intellectual Property Section reported six incidences in the last month alone, ranging from trafficking in counterfeit computer programs to accessing government databases [USDOJ 2009]. News stories detail how insiders have bilked corporations out of millions due to their ability to access sensitive information – sometimes after they have resigned from a company that did not immediately remove their confidential access [Vijayan 2009]. There have even been studies that suggest that the economy has impacted, or will impact, the surge in cybercrime [Kirk 2009][Bush 2009].

For the last several years, companies have been analyzing their IT operations and processes for the purpose of uncovering insider threats and cybercrime. Most approaches have either been statistical in nature, leading to various data mining approaches, or a visualization of their resources where they can monitor for illegal access or entry. However, recently, the ability to mine relational data has become important for detecting *structural* patterns. The complex nature of heterogeneous data sets, such as network activity, e-mail, payroll and employee information, provides for a rich set of potentially interconnected and related data. Graph-based data mining approaches analyze data that can be represented as a graph (i.e., vertices and edges), yet little work has been done in the area of *graph-based anomaly detection*, especially for application to cybercrime.

In partial response to this issue, we propose the use of a *graph-based approach* for discovering cybercrime. As part of the IEEE Symposium on Visual Analytics Science and Technology (VAST) for 2009 [VAST 2009], one of the mini-challenges consisted of various aspects of an insider threat, where an employee is leaking information. Using data that consists of "badge swipes" and network IP traffic over a month, we used the Graph-Based Anomaly Detection (GBAD) tool to analyze anomalous instances of structural patterns in data, where the data represents entities, relationships and actions in graph form [Eberle and Holder 2007]. Input to GBAD is a labeled graph in which entities are represented by labeled vertices and relationships or actions are represented by labeled edges between entities. GBAD embodies novel algorithms for identifying the three possible changes to a graph: modifications, insertions and deletions. Each algorithm discovers those substructures that match the closest to the normative pattern without matching exactly. As a result, GBAD is looking for those activities that appear to match normal patterns, but in fact are structurally different.

We hypothesize that such a system can aide in the discovery of knowledge in a graph representation of this type of data by (1) showing the normal structure of the employee movements and activity, and (2) showing anomalies in employee behavior, indicating a possible insider threat. In addition, while previous work with graph-based approaches have demonstrated mixed results when it comes to performance, we will show that GBAD's performance on these graphs, as well as synthetic graphs of similar size and complexity, can provide reasonable response times for real-world analysis.

Related Work

The ability to mine relational data has become important in several domains for detecting various structural patterns. One important area of data mining is anomaly detection. The ability to mine data for nefarious behavior is difficult due to the *mimicry* of the perpetrator. If a person or entity is attempting to commit fraud or participate in some sort of illegal activity, they will attempt to convey their actions as close to legitimate actions as possible. For instance, the United Nations Office on Drugs and Crime states the first fundamental law of money laundering as "The more successful money-laundering apparatus is in imitating the patterns and behavior of legitimate transactions, the less the likelihood of it being exposed" [Hampton and Levi 1999].

Recently there has been an impetus towards analyzing relational data using graph theoretic methods [Holder and Cook 2007]. Not to be confused with the mechanisms for analyzing "spatial" data, graph-based data mining approaches analyze data that can be represented as a graph (i.e., vertices and edges). While most graph-based data mining research has dealt with intrusion detection [Staniford-Chen 1996], there has been some research in the area of graph-based anomaly detection. In Priebe et al's work, they used what are called "scan statistics" on a graph of the e-mail data that is represented as a time series [Priebe et al 2005]. While their approach detects statistically significant events (excessive activity), without further analysis, they are unable to determine whether the events are relevant (like insider trading). Martin et al. examined what they called "behavioral features" of a particular user's network traffic in order to discover abnormal activity [Martin et al 2005]. Through various clustering approaches, and comparisons to methods such as Support Vector Machines and Naives Bayes Classification, they group sets of users into single behavioral models. Diesner et al. applied various network analytic techniques in their exploration of the structural properties of the Enron network. They used various graph structural metrics, such as betweenness centrality, eigenvectors and total degree in order to identify key players across time [Diesner and Carley 2005]. In 2007, Kurucz et al. used hierarchical spectral clustering to evaluate weighted call graphs [Kurcz 2007]. They analyzed several heuristic approaches using phone calls made over an eight-month period. However, their purpose was not to expose anomalies in phone traffic, but instead to address the issues associated with processing large graphs. In Swayne et al's work, they used graph techniques to explore AT&T phone records [Swayne et al 2003]. While their approach was able to provide for the analysis of phone traffic, it was entirely based upon a graph-visualization, rather than any graph theoretic approaches. In fact, when it comes to generating graphs of information, much research has dealt with only the visual aspects of what is represented, rather than the structural aspects of the graphs themselves

The advantage of graph-based anomaly detection is that the relationships between elements can be analyzed, as opposed to just the data values themselves, for structural oddities in what could be a complex, rich set of information.

Graph-Based Anomaly Detection Approaches

The idea behind the approach used in this work is to find anomalies in graph-based data where the anomalous substructure in a graph is part of, attached to, or missing from a *normative substructure*.

Definition: *A graph substructure S' is anomalous if it is not isomorphic to the graph's normative substructure S, but is isomorphic to S within X%.*

X signifies the percentage of vertices and edges that would need to be changed in order for *S'* to be isomorphic to *S*.

GBAD (Graph-based Anomaly Detection) is an *unsupervised* approach, based upon the SUBDUE graph-based knowledge discovery method [Cook and Holder 1994]. Using a greedy beam search and Minimum Description Length (MDL) heuristic [Rissanen 1989], each of the three anomaly detection algorithms in GBAD uses SUBDUE to find the best substructure, or normative pattern, in an input graph. In our implementation, the MDL approach is used to determine the best substructure(s) as the one that minimizes the following:

$$M(S,G) = DL(G \mid S) + DL(S)$$

where *G* is the entire graph, *S* is the substructure, *DL(G|S)* is the description length of *G* after compressing it using *S*, and *DL(S)* is the description length of the substructure.

There are three general *categories of anomalies*: insertions, modifications and deletions. Insertions would constitute the presence of an unexpected vertex or edge. Modifications would consist of an unexpected label on a vertex or edge. Deletions would constitute the unexpected absence of a vertex or edge. We have developed three separate algorithms: GBAD-MDL, GBAD-P and GBAD-MPS. Each of these approaches is intended to discover one of the corresponding possible graph-based anomaly categories as set forth earlier. The reader should refer to [Eberle and Holder 2007] for a more detailed description of the actual algorithms.

Cybercrime Scenario

One particular example of cybercrime is the leaking of information by employees with access to confidential and sensitive information. As part of the IEEE Symposium on Visual Analytics Science and Technology (VAST) for 2009, three mini-challenges and one grand challenge were posted as part of their annual contest. Each of the mini-challenges consists of various aspects of a fictional insider threat, based upon the leaking of information. The goal of these challenges is to allow contestants to apply various visual analysis techniques so as to discover the spy and their associated actions.

While our GBAD approaches are not "visually based", we chose to apply our algorithms to the mini-challenge that consists of badge and network IP traffic. The proxy data set is comprised of employee "badge swipes" during the month of January in 2008, and the IP log consists of all network activity to and from the facility. One of the goals of this mini-challenge was to determine what computers the "spy" used to send the sensitive information.

Discovering Cybercrime

We can separate the cybercrime discovery process into three separate tasks:

1. Discover the anomalous network activity,
2. Create *targeted* graphs for just those days and people that might be involved in the anomalous activity, and
3. Use GBAD to discover what employees participate in anomalous activity.

Discovering Unusual Activity

The first stage of this process is to discover the network activity that is unusual – or the source of illegal transmissions. Rather than apply a graph-based approach to the discovery of what would be numerical/statistical anomalies (i.e., non-structural anomalies), we can do a simple analysis of the actual records. Sorting the IP logs by packet size, one discovers that the top five transmissions by packet size are all to the same destination IP, 100.59.151.133 on port 8080:

```
…
Synthetic Data   37.170.100.31   2008-01-15T17:03   100.59.151.133  8080     9513313     14324
Synthetic Data   37.170.100.20   2008-01-24T17:07   100.59.151.133  8080     9732417     42347
Synthetic Data   37.170.100.13   2008-01-22T08:50   100.59.151.133  8080     9984318     42231
Synthetic Data   37.170.100.56   2008-01-29T15:41   100.59.151.133  8080    10024754     29565
Synthetic Data   37.170.100.8    2008-01-31T16:02   100.59.151.133  8080    13687307    485421
```

In the IP log file, the first column is the type of data, the second column is the source IP, the third column is the date and time, the fourth column is the destination IP, the fifth column is the destination port, the sixth column is the size of the transmission packet, and the last column is the size of the response record. In fact, 17 of the 32 highest-transmission records have this same destination IP - clearly an unusual volume of traffic to a single, external destination.

In addition, with our graph-based approach, we can *verify* the anomalousness of the traffic based upon the relationship of the activity within the graph. For example, knowing that employee 31's computer is one of the computers that sent the supposedly illegal transmissions (see the top record above), we can analyze the subgraph of that employee's activity on that day. Figure 1 is a partial graphical representation of that subgraph.

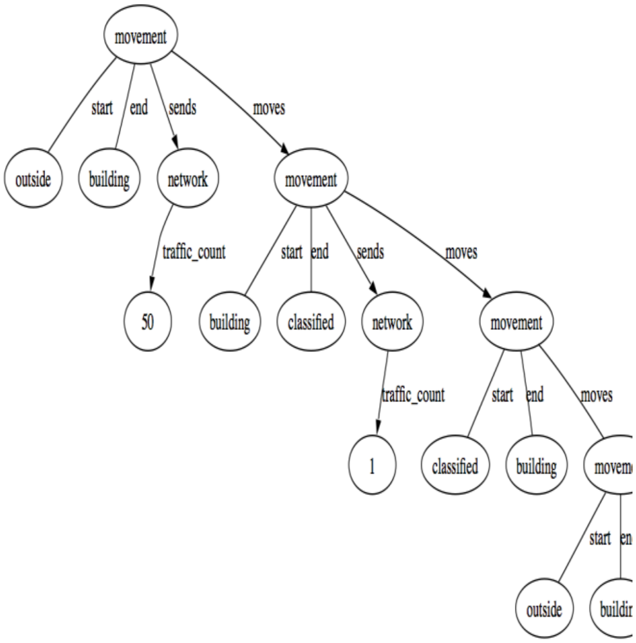

Figure 1. Example movement and activity.

Notice that there is a transmission (sends->network) when the location is "classified" – which is not possible because the employee's computer is not in the classified area. This type of *structural* anomaly supports the claim that unusual network activity occurred from this computer (on this day).

Targeting Access to Computers

In order to discover an insider committing this form of cybercrime, we make two assumptions:

1. The insider never uses their own computer (for fear of their actions being traced back to them), and
2. The insider only uses someone else's computer when they are in the classified area (as that is the only time we know that they are not in their office).

Using these two assumptions, we can then focus on the generation of graphs that (1) exclude people whose computer was compromised from being considered as suspects, and (2) reduce the graph search space to only those days where the illicit transmissions took place. In this data set, 10 employees are removed from being considered as suspects, and only the activity of other employees during the anomalous network activity are represented in the graph. This will enable us to analyze abnormal structure in the graph during the times of the crimes.

So first we create graphs consisting of subgraphs that represent employee movements for *each* targeted day (i.e., the days when the illicit transmissions took place), as well as graphs that represent the movements for each employee over *all* of the targeted days. Each subgraph will contain a "backbone" of movement vertices. Attached to the movement vertices will be two vertices representing where the person was before entering the current location and the current location (i.e., outside, building, classified). The edges will be labeled start and end, respectively. Then, if network traffic is sent before the person moves again, a network vertex will be created and linked to the movement vertex via a sends edge. The network vertex will also be linked to a vertex with a numerical label, representing how many messages are sent before the next movement occurs. The result is a graph topological representation as shown in Figure 2.

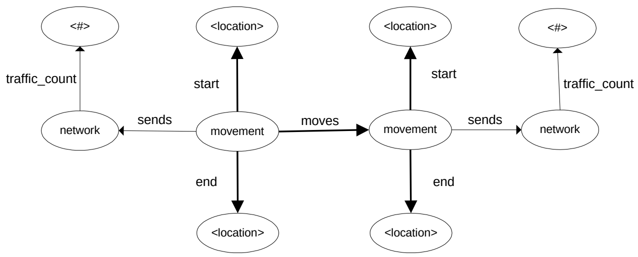

Figure 2. Graph representation.

In the partial example shown in Figure 3, visualized using the GraphViz tool [www.graphviz.org], a person enters from the outside, transfers some data across the network, and then moves into the classified area.

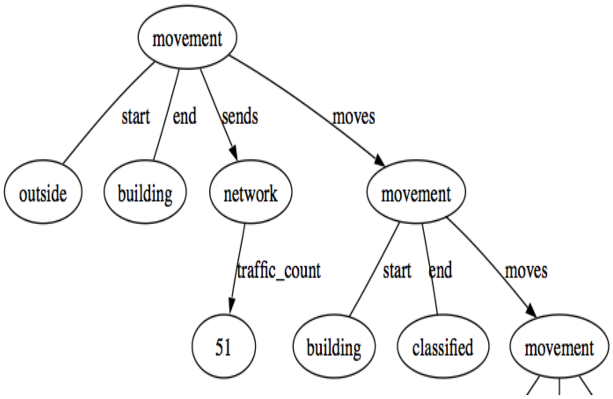

Figure 3. Example movement and activity.

A graph input file for GBAD is an ASCII text file that defines the vertices using sequential numbering and the edges using numbered vertices. We created a tool to process the comma-delimited proxy log and IP log files and output a graph file for use with GBAD. Once the graph files are created, GBAD can then be used to obtain (1) the normative pattern discovered in the specified graph input file and (2) the top-N most anomalous patterns.

Using this graph representation, Figure 4 shows a visualization of a normative pattern.

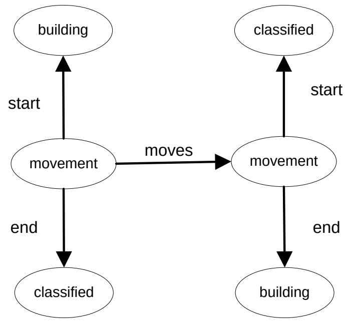

Figure 4. Normative pattern.

After uncovering the normative pattern, GBAD can then use its three algorithms to discover all of the possible structural changes that can exist in a graph (i.e., modification, deletions, and insertions).

The VAST data set consists of the activities of 60 employees at an embassy over the month of January in 2008. As stated earlier, there are 17 transmissions to the suspect IP. Based upon our first assumption, we can remove 10 employees from the list of suspects (some employees were compromised more than once). We can also reduce our data set down to just the days where the anomalous transmissions took place, which consists of 8 of the 31 available days worth of information. This subset of the data is then the baseline for our GBAD analysis.

Discovering the Suspicious Employee

Using these targeted graphs (8 day graphs and 50 people graphs), we ran the GBAD algorithms using default parameter settings, where it would report only the most anomalous instances, rather than the top-K instances. On the graphs that represent individual people and their movements and network activities across all targeted days, the GBAD-MDL algorithm discovers 12 employees as having anomalous movements and activities, and the GBAD-MPS algorithm reports 8 employees as anomalous. On the graphs that represent all movements and activities for each targeted day, GBAD-MDL reports 6 employees as anomalous while GBAD-MPS reports 2 employees. However, there is an interesting commonality across all four experiments. If you take the overlap (intersection) between them, in other words which employees are reported in ALL of the experiments, one discovers that there are only 2 employees that are very suspicious: employee 49 and employee 30.

We can further distinguish a difference between these two employees by analyzing the graphs and GBAD results. From the GBAD results, employee 30 is reported as the most anomalous (score-wise) on 6 of the 8 days, with employee 49 being the most anomalous on the other 2. Also, employee 30 is the only employee with the structural anomaly shown in Figure 5.

In Figure 5 (for space reasons, only the part of the graph necessary for this observation is shown), one will notice that the employee initially moves from the outside into the building. However, their next move is from the classified area into the building – with no movement into the classified area before that. This is called "piggybacking", where an employee does not use their badge, but instead follows on the heels of another employee. Employee 30 is not only the only employee to piggyback into the classified area, but they do it several times. Perhaps their intent is to gather classified information without a trace of ever entering the area. Unfortunately (for them), they had to badge-out of the area – resulting in a structural anomaly in their movement.

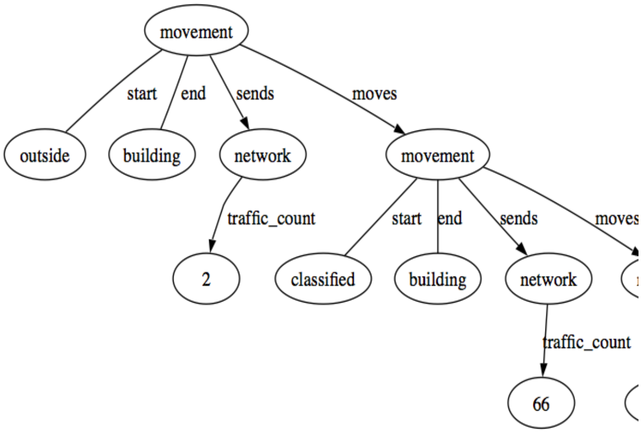

Figure 5. Anomalous structure.

It should also be noted that the GBAD-P algorithm does not report any significant movement or activities as anomalous, but does report the differences in network packet sizes. In addition, it is interesting to note that all of the anomalous activity takes place on Tuesdays or Thursdays. Perhaps some future work in anomaly detection could be the detection of structural patterns in the anomalies themselves.

Performance

Of course, the ability to discover anomalies is critical to the success of any anomaly detection system. However, in order to be useful as a real-world application, the performance of the algorithms must be viable for real-time analysis.

For the VAST data set presented in this paper, the average time for GBAD runs on the graphs representing all employee movements for a single day (~3,500 vertices and edges) is 1.07 seconds, with a maximum time of 1.91 seconds. The average time for runs on the graphs representing each employee's movements over all of the targeted days (~600 vertices and edges) is 0.06 seconds, with a maximum time of 0.12 seconds.

We also looked at the running times of the GBAD approach on the complete VAST data set, where the data was not reduced to a targeted set of individuals or days. The average running time for GBAD on a single graph representing the movements of all 60 employees over all 31 days was 443.44 seconds. This single graph consists of ~40,000 vertices and ~38,000 edges.

In this domain, the graphs are relatively sparse, as evident by the number of the edges versus the number of vertices, with a tree-like topology. So, to further analyze the performance of GBAD on various graph sizes, we generated various synthetic, sparse graphs with random anomalies created. Figure 6 shows the performance of GBAD on our synthetic experiments as a linear log-log plot.

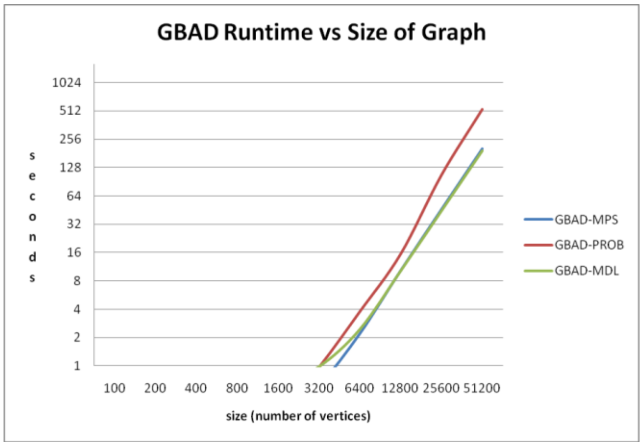

Figure 6. GBAD running times on synthetic graphs.

These results show that the running time is polynomial to the size of the graph – an acceptable performance for an application that needs results in a timely manner.

Conclusions and Future Work

Using a graph-based approach, we have been able to successfully discover anomalies in graphs that represent cybercrime. In addition, the performance of GBAD on an example domain, as well as graphs of varying sizes, indicates that such an approach can be used in real-world situations. Some future directions that we are exploring include the incorporation of traditional data mining approaches as additional quantifiers to determining anomalousness, as well as applying graph-theoretic algorithms to dynamic graphs that change over time. Using tools that can simulate network activity and personnel actions and movements, such as the OMNeT++ simulator [OMNeT], we can create limitless numbers and varieties of simulations modeling cybercrime. These can then be used to evaluate GBAD both systematically and on models of known cybercrimes, like those reported by CERT [Kowalski et al. 2008]. The importance of this technology lies in the necessity of being able to detect cybercrime before further damage has been perpetrated. Organizations need advanced tools to detect cybercrime, and they need to be able act in a timely fashion.

Acknowledgement

This material is based upon work supported by the Department of Homeland Security under Contract No. N66001-08-C-2030. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Homeland Security.

References

Bush, J. *Survey suggests economy could lead to cybercrime increase*, Purdue University News Service, March 19, 2009.

Cook, D. and Holder, L. *Graph-based data mining*. IEEE Intelligent Systems 15(2), 32-41, 1998.

Diesner, J. and Carley, K. *Exploration of Communication Networks from the Enron Email Corpus*. Computational and Mathematical Organization Theory, Volume 11, Issue 3, pp. 201-228, October 2005.

Eberle, W. and Holder, L., *Anomaly Detection in Data Represented as Graphs*, Intelligent Data Analysis, An International Journal, Volume 11(6), 2007.

Hampton, M. and Levi, M., *Fast spinning into oblivion? Recent developments in money-laundering policies and offshore finance centres*. Third World Quarterly, Volume 20, Number 3, pp. 645-656, 1999.

Holder, L. and Cook, D., *Mining Graph Data*. John Wiley and Sons, 2007.

Kirk, J. *In poor economy, IT pros could turn to e-crime*, IDG News Service, March 25, 2009.

Kowalski, E., Cappelli, D., Conway, T., Willke, B., Keverline, S., Moore, A. and M. Williams, *Insider Threat Study: Illicit Cyber Activity in the Government Sector*, January 2008. URL: http://www.cert.org/.

Kurcz, M., Benczur, A., Csalogany, K. and Lukacs, L. *Spectral Clustering in Telephone Graphs*. Joint 9th WEBKDD and 1st SNA-KDD Workshop '07 (WebKDD-SNA-KDD '07), August 12, 2007.

Martin, S., Sewani, A., Nelson, B., Chen, K. and Joseph, A. *Analyzing Behaviorial Features for Email Classification*. CEAS 2005: Conference on Email and Anti-Spam, July 21-22, 2005.

OMNeT++. http://www.omnetpp.org/.

Ponemon, L. *Cyber Crime: The 2009 MegaTrend.* CSO, 2009.

Priebe, C., Conroy, J., Marchette, D. and Park, Y. *Scan Statistics on Enron Graphs*. Computational and Mathematics Organization Theory, Volume 11, Number 3, p229 - 247, October 2005.

Rissanen, J. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Company, 1989.

Staniford-Chen, S. et al., *GrIDS – A Graph Based Intrusion Detection System for Large Networks*. Proceedings of the 19th National Information Systems Security Conference, 1996.

Swayne, D., Buja, A and Lang, D. *Exploratory Visual Analysis of Graphs in GGobi*. Proceedings of the 3rd International Workshop on Distributed Statistical Computing, DSC 2003, March 20-22, 2003.

United States Department of Justice, Computer Crime & Intellectual Property Section, http://www.justice.gov/criminal/cybercrime/.

VAST 2009 Challenge: www.cs.umd.edu/hcil/VASTchallenge09.

Vijayan, J. *Insider at Cal Water steals $9M and runs*. Computerworld Security, May 22, 2009.