Analyzing Network Traffic to Detect E-Mail Spamming Machines

Prasanna Desikan and Jaideep Srivastava
Department of Computer Science
University of Minnesota, Minneapolis, MN-55455
{desikan,srivasta}@cs.umn.edu

Abstract

E-Mail spam detection is a key problem in Cyber Security; and has evoked great interest to the research community. Various classification based and signature based systems have been proposed for filtering spam and detecting viruses that cause spam. However, most of these techniques require content of an email or user profiles, thus involving in high privacy intrusiveness. In this paper, we address the problem of detecting machines that behave as sending spam. Our approach involves very low privacy intrusion as we look at only the border network flow data. We propose two kinds of techniques for detecting anomalous behavior. The first technique is applicable for single instance network flow graph. The second technique involves analyzing the evolving graph structures over a period of time. We have run our experiments on University of Minnesota border network flow. Our results on this real data set show that the techniques applied have been effective and also point to new directions of research in this area.

KEYWORDS: E-Mail Spam Detection, Privacy and Security

1. Introduction

Cyber Security has emerged as one of the key areas of research interest with increase in information stored online and the vulnerability to attacks of such an information infrastructure. Over the years, the dependency on information infrastructure has increased, and so has their sophistication and potency. There have been intelligent and automated tools that exploit vulnerabilities in the infrastructure that arise due to flaws in protocol design and implementation, complex software code, mis-configured systems, and inattentiveness in system operations and management. The most common exploit seen is the buffer-overflow attack [4].

Technological advancements on the Internet have contributed very significantly in making information exchange very easy across the globe. E-Mail is the most popular medium for individuals to communicate with each other. However, such an effective communication medium is being increasingly abused. According to a recent survey, the number of spam mails has increased from 8% in 2001 and 50% in 2004 [8]. This alarming increase in the rate of spam mails is of concern for operational as well as security reasons. The total estimated cost incurred due to spamming was around $10B/yr in US (2002) [8]. To the cyber-security community, this is of concern, especially when machines inside a sensitive network are sending spam or huge amounts of information to the outside. Also, of interest are machines from outside the network that try to scan to use the exploits in the machines inside the network. It is very critical to differentiate such machines from those that are sending mail normally.

In this paper, we address the issue of identifying the machines that are sending spam, or machines that have been compromised and are being used as a spam relay. Note that our focus is not on identifying individual users who send spam, or filtering an e-mail as spam based on its content. There has been work in such areas which is not directly related to ours [10, 11, 15]. Recent work on detection of spam trojans suggests the use of signature and behavior based techniques [12]. However, using signatures will fail to detect novel attacks at an early stage and require looking into message content. Dealing with such problems would require availability of data that would be sensitive with respect to security and privacy which limits the applicability of these techniques. We have implemented our techniques as a part of the MINDS project [7].

In section 2 we describe the various kinds of data that can be analyzed from e-mail traffic, and the levels of privacy involved. Section 3 gives a brief overview of link analysis techniques that can be applied for network security. Our approaches are explained in detail in Sections 4, 5. Results of experimental evaluation of our approaches are presented in Section 6. Section 7 discusses other works that are related to this topic. Finally, we conclude in Section 8 and point to future directions.

2. E-Mail Architecture and Privacy Issues

Electronic Mail is technically a file transfer from one machine to another and is initiated by the sender. The architecture of this service is illustrated in Figure 1. The Mail Client is responsible for creating the message files and sending and receiving them at the host level.

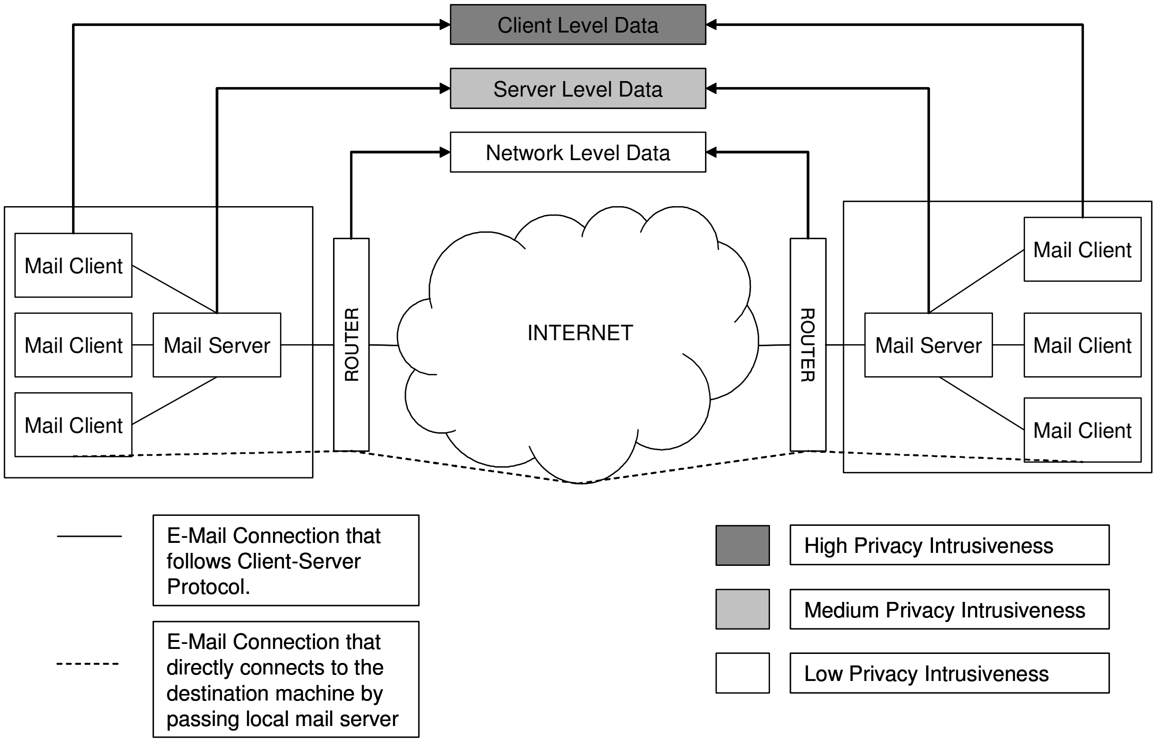

Client Level Data
Server Level Data
Network Level Data
Mail Client
Mail Client
Mail Server
Mail Client
ROUTER
INTERNET
ROUTER
Mail Client
Mail Server
Mail Client
Mail Client
E-Mail Connection that follows Client-Server Protocol.
E-Mail Connection that directly connects to the destination machine by passing local mail server
High Privacy Intrusiveness
Medium Privacy Intrusiveness
Low Privacy Intrusiveness

Figure 1. Architecture of Electronic Mail

The *Mail Client* handles the part of transferring a file to or from a mail server. The Mail Server handles the message files received from various mail clients within its network, and transfers them to the Internet where other mail transfer agents transfer the files to the mail servers of respective destinations. A receiving *Mail Server* is responsible for putting the received message files in mailboxes of the respective users. The *Mail Client* at the recipient end can retrieve the message files from the *Mail Server.* The transfer of messages between a mail server and other mail transfer agents within the Internet takes place via a TCP connection using the SMTP protocol. The transfer between a client and the local mail server uses protocols such as POP or IMAP. It should be noted that all emails do not necessarily pass through the mail server and a client can open a connection on a different port and communicate directly to another machine[1]. The *border router* collects all information about the network connections made in and out of the network.

It can be seen that with this architecture, data can be collected at different points. Data collected at such point reveals different kinds of information and with different granularity and privacy levels. We now discuss the kinds of information that can be extracted, and the respective levels of privacy intrusion. The darkness of the shaded boxes indicates the level of privacy intrusion in Figure 1.

Mail Client Data: The data that can be collected at this level is primarily the files that have been transferred and received. These files contain information about all the people the user sent mail to or received mail from, the date and time of such transfer. Mail clients also contain meta data such as the folders in which these files are stored, the mails that been replied to, forwarding, and more recently introduced concept of ‘conversations’. Other interesting information that can be obtained at a meta level is the contact information from the address book. Such data has high level of privacy intrusiveness.

Mail Server Data: The data that can be obtained at this level is the set of all files that have been transferred. These files can reveal who communicated with whom, when and about what topic. The level of granularity is fine, as we know everything that has been exchanged between the sender and receiver of email. The main difference between the data at the Mail Server level versus the Mail Client Level is the meta-data for each user discussed earlier. The level of privacy intrusion still

[1] However, such email is the rare exception rather than norm

remains high, as all information about the content of the file exchanged is available.

Network Level Data: These include data that can be collected at the network interface levels. The two main kinds of such data are the Tcpdump data and Netflow data. Tcpdump data contains a log of all the packets that passed the network sensor, including the packet content. Thus, the data provides a fine level of information granularity, which can lead to high level of privacy intrusiveness, though analyst may not be able to figure out the exact conversation if secure protocols such as SSL are used. Netflow data on the other hand is collected from routers (e.g. Cisco, Juniper). Each flow is a summary of traffic traveling in one direction in a session. When the router tears down a flow, a flow record is created. This flow record contains basic information about the connection, such as source/destination IP/ports, number of packets/bytes transferred, protocol used, and cumulative OR of TCP flags. However, flow records do not contain payload information. An email service connection that uses the SMTP protocol typically has the destination port as 25. The Netflow data has medium granularity of information and the privacy intrusiveness is at a much lower level as compared to the data obtained at the client level or the server level.

3. Link Analysis Techniques for Network Security

An interesting kind of information infrastructure that can be constructed from the types of data discussed previously is a 'link graph'. Link graphs can be used to represent information from a single source of data or from multiple sources. Interaction between different systems can be understood better by modeling them as link graphs. The key idea to modeling a given data as a link graph is to represent an agent of information or a given state as a node and the link as the connection or transition between them. For example, nodes can be IP addresses, ports, usernames or routers and the links the different connections between them. Once a link graph is generated, link analysis techniques can then be used to identify all interaction based behavioral patterns that are causes of possible threats.

Link analysis techniques have been popular in various domains and the significance and emergence of these techniques has been discussed by Barabasi in his book [1]. Link analysis has been successfully applied to mine information in domains like web [5], social networks [10] and computer security [15]. In our earlier work we have surveyed the existing link analysis techniques to the web domain and introduced taxonomy for research in this area [4]. A consequence of this was to develop a methodology to adopt link analysis techniques to different applications.

Link analysis can be thus been viewed as primarily used for two purposes namely, integration of different data sources, and profiling the system or user interactions. Accordingly, the kind of analysis performed varies depending on the data available. For example, Netflow data gives traffic flowing in one direction and hence a directed graph can be built at the level of an IP address or port. If we use TCP dump data, additional information about the content will be available and we can weigh the nodes and links accordingly to get a better picture of actual traffic. The traffic data will help in building graphs that reflect system interactions. Link analysis can then be used to find 'communities' of systems that have similar interactive behavior patterns. At the host level, syslogs can be used to model the sequence of commands (or the applications executed one after other can be connected by a link) as a graph and profile the host based on the command-command graphs. A mapping between the user (or a machine) and the list of commands issued (executed) will enable the profiling of users (machines) that execute these commands (run the applications) frequently. For example, analysis of a bipartite structure, with users (machines) as one set and the commands (applications) as the other set, would identify a group of users (machines) with similar behavior patterns. Information from server logs such as the web server or the database server can also be integrated. Link analysis techniques can be applied BGP router information to identify communities of networks that have similar usage pattern, and also key router locations that need to be monitored. The trade-off in privacy for the various kind of data was discussed in the earlier section.

Most techniques in link analysis have so far concentrated on identifying prominent normal behavior [9]. Other techniques such as attack graphs[16] have modeled possible plans based on a formal logic approach and have an underlying assumption that all events are observable. This makes them incapable of detecting novel attacks. Hence, there is a need to define measures for anomalous behavior in the link graph terminology to help detect attacks. Furthermore, most techniques developed so far have been related to static graphs. However, the network topology keeps changing and so do user patterns, and hence there is a need to develop robust techniques for evolving graphs. For long-term analysis, historical data of attacks or anomalous behavior can be collected and used to identify nodes that have been prominent 'perpetrators' and nodes that have been most 'vulnerable'. In summary Link Analysis Techniques for Network Security can be used to:

- Identify nodes (machines) and edges (connections) that are anomalous in behavior.
- Identify nodes highly likely to be possible sources of attack or are vulnerable over a period of time.

- Identify 'communities' of machines involved in 'normal' as well as 'anomalous' connections.
- Study the changing behavior of connections by analyzing temporal behavior of graphs.

4. Our Approach

E-mail servers traditionally send and receive mails from other e-mail servers. Thus, e-mail servers among themselves form a community due to interactions with each other. More precisely, they form among themselves a dense bipartite graph. We utilize this behavior of e-mail servers to profile normal versus anomalous behavior. In the following sub-section, we describe an existing approach to identify such bipartite graphs that has been used in other domains such as the web. We will then describe a way to utilize this to detect anomalous behavior of e-mail servers.

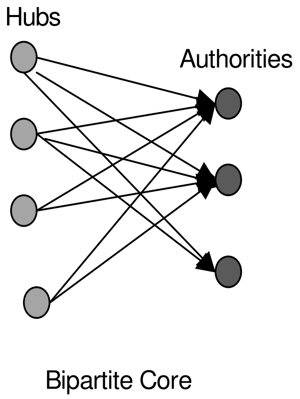


Figure 2. Hubs and Authorities

4.1 Hubs and Authorities

Identifying bipartite cores has been of interest in Web Mining domain. A bipartite core (i, j) is defined as a complete directed bipartite sub-graph with at least i nodes from one set of nodes to at least j nodes from another set of nodes. Figure 2 illustrates this concept.

With reference to the Web graph, *i* pages that contain the links are referred to as 'hubs' and *j* pages that are referenced are the 'authorities'. For a set of pages related to a topic, a bipartite core can be found that represents the Hubs and Authorities for the topic can be found using HITS algorithm [9]. Hubs and Authorities are important since they serve as good sources of information for the topic in question. In the domain of e-mail traffic flow, 'hubs' are equivalent to machines that send mails and 'authorities' are machines that receive mails and together they form a bipartite core. Such a behavior is typical of e-mail servers that send and receive mails from other servers. E-mail servers serve as both good hubs and good authorities. Hence, a bipartite graph captures the behavior of machines that are typically E-Mail Servers.

We will briefly describe the idea behind HITS algorithm. Let A be an adjacency matrix such that if there exists at least one connection from machine i to machine j, then Ai, j = 1, else Ai, j = 0. Kleinberg's algorithm, popularly known as the HITS algorithm [9], is described in Figure 3. This is a recursive algorithm where each node is assigned an authority score and a hub score. Hence we see that hub scores will be higher if it points to many nodes or nodes with high authority. Conversely, authority scores will be higher if it is pointed to by many nodes or pointed by good hubs.

The recursive nature of the iterations in the matrix computation will result in the convergence of authority and hub score vectors to the principal eigen-vectors of A^TA and AA^T respectively.

```
HITS ALGORITHM

Let $a$ is the vector of authority scores and $h$ be
the vector of hub scores
$a$=[1,1,….1], $h$ = [1,1,…..1] ;
do
        $a=A^Th$;
        $h=Aa$;
        Normalize $a$ and $h$;
while $a$ and $h$ do not converge(reach a
convergence threshold)
$a^* = a$;
$h^* = h$;
return $a^*$,  $h^*$
The vectors a* and h*represent the authority and
hub weights
```

Figure 3. HITS Algorithm

4.2 Identifying Potential Perpetrators

Existing link analysis techniques fail to detect machines that send spam or are used to relay spam. Most techniques are used to mine for behavior that is normal and dense within a community, as opposed to anomalous or rare behavior. To detect e-mail spamming machines we need to differentiate their behavior from those of the e-mail servers. Both of them will tend to have high outgoing traffic. However, an e-mail server tends to send e-mails to only other e-mail servers whereas a spamming machine sends mail to all machines. We make use of this behavioral aspect to detect the potential perpetrators.

We follow the following sequence of steps:

1. Pre-process the netflow data and construct the graph for e-mail connections.

   - *Graphs can be constructed for patterns that represent other kind of services like ftp.*
   - *Node can be an IP or AS or port or any combination depending on the problem. We do our analysis at an IP Level.*
2. Perform the HITS Algorithm on the generated graph.
   - *The nodes with top hub and authority scores represent typical e-mail servers*
3. Remove edges between top *k%* of hubs to top *k%* authorities.
   - *These top k % connections correspond to normal e-mail traffic between regular mail servers that have high hub and authority score.*
4. Perform the HITS algorithm on the resultant graph.
   - *A simple outdegree also works fine on the resultant graph.*
5. The new scores are the ***Perpetrator Scores***.
   - *Spamming machines obtain high rank compared to other e-mail servers.*

It can be seen that our approach is two-fold. Firstly, it identifies connections between regular mail servers. Such connections form a dense bipartite graph between servers, assigning them high *hub* and *authority* scores. All such connections that contribute to normal e-mail traffic are then removed. Note, only the edges are deleted and not the nodes. This eliminates normal e-mail server behavior. The second step identifies machines that behave like servers and have high traffic that does not correspond to regular e-mail connections. These machines are most likely spamming, since they send mails to a lot of other machines that do not take part in regular e-mail connections. Since no node is deleted, such an approach also helps to identify e-mail servers that are affected and sending spam. Figure 4 illustrates this concept clearly.

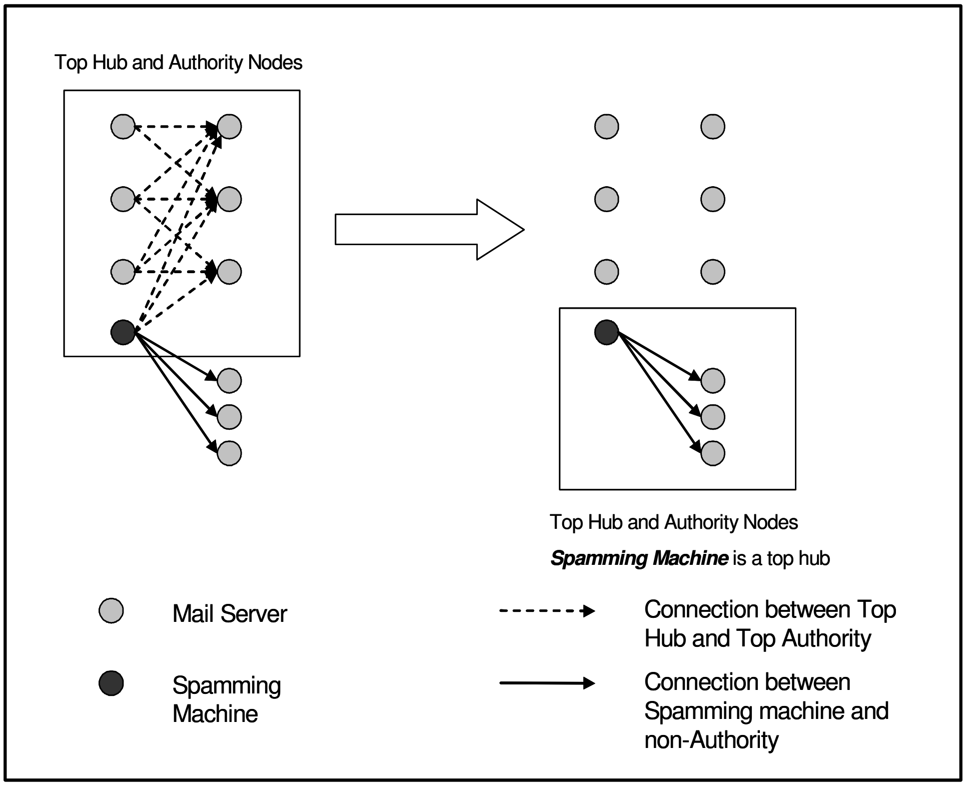

Top Hub and Authority Nodes
Top Hub and Authority Nodes
Spamming Machine is a top hub
Mail Server
Spamming Machine
Connection between Top Hub and Top Authority
Connection between Spamming machine and non-Authority

Figure 4: Identifying spamming machines

5. Temporal Evolution of Graphs

Link Analysis techniques have primarily focused on analyzing a graphs at a single time instance. However, graphs evolve over time, and much information can be gained by understanding their evolution. In earlier work, we have shown the significance of mining information from such evolving graphs in the web domain [5]. Graphs such as network graphs based on e-mail connections change rapidly, and there is a need to define properties that need to be measured and develop techniques capture the changing behavior. The sequence of steps for such an analysis is described below:

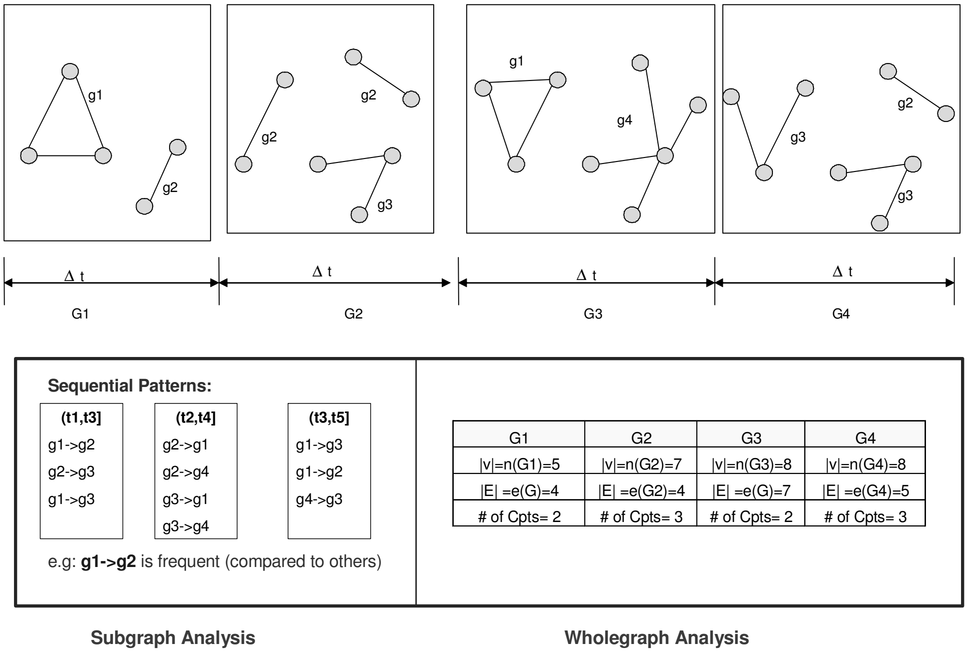
- Decide the Scope of Analysis: Single Node, Subgraph, Whole Graph.
- Develop Time Aware Models (e.g. Graph Models + Time Series Models).
- Define Time Aware Measures and Metrics.
- Design Efficient Algorithms (Incremental and Parallel) for computing metrics for all graphs.

In the following subsection we will describe the three levels of scope of analysis in detail. Figure 5 illustrates an example of an evolving graph. G1, G2, G3, G4 represent the snapshots of the graph taken at the end of consecutive time periods. The different subgraphs in each snapshot are represented as g1, g2, g3, g4. Each time period is of length, Δt. The start and end time instances of each time period are represented from t1 to t5. The order and size of graph are represented as |v| and |E|.

5.1 Analysis Scope

The models and techniques developed will also depend on the scope of analysis. The temporal behavior of the Web graph can be analyzed at three levels:

- **Single Node:** Studying the behavior of a single node across different time periods. Over a period of time, inherent properties of a node, such as machine configuration, can change, signifying the change in functionality of the node. Also, structural changes of a node over a time period can be analyzed by studying the variation of properties. Typical examples of properties based on link structure are indegree, outdegree, authority score, hub score and PageRank score. Such behavior will also serve as useful feedback to a network analyst. Finally, study of usage data of a single node across a time period, will reflect the activity of a node during the given time period. The temporal dimension will helps to identify current trends and helps in the prediction of active machines.
- **Sub-graphs:** At the next hierarchical level, changing sub-graph patterns evoke interest. These sub-graphs may represent different communities or connection patterns, representing services like e-mail, ftp, p2p, etc. that evolve over time. The idea of mining frequent sub-graphs has been applied with a large graph, or a set of small graphs, as input [16]. However, with addition of a temporal dimension, we look at an evolving graph, which may have different sets of sub-graphs at different time instances. Figure 5 illustrates an example of an evolving graph, and the sequential patterns that can be mined. In the example it is seen that if a subgraph pattern, g1, occurs during a time interval, the probability that a subgraph, g2, will occur in the next time period is higher than any other sequence of subgraphs over adjacent time periods. It can be seen that mining of sequential patterns of sub-graphs might provide useful information in profiling the changing behavior. Sequence mining may also help in predicting an emerging trend or predict an abnormal behavior in network traffic.



Sequential Patterns:

(t1,t3]	(t2,t4]	(t3,t5]
g1->g2	g2->g1	g1->g3
g2->g3	g2->g4	g1->g2
g1->g3	g3->g1	g4->g3
	g3->g4	

e.g: **g1->g2** is frequent (compared to others)

G1	G2	G3	G4
|v|=n(G1)=5	|v|=n(G2)=7	|v|=n(G3)=8	|v|=n(G4)=8
|E| =e(G)=4	|E| =e(G2)=4	|E| =e(G)=7	|E| =e(G4)=5
# of Cpts= 2	# of Cpts= 3	# of Cpts= 2	# of Cpts= 3

Figure 5: Analysis of evolving graphs

- **Whole graph:** While analysis of single nodes and sub-graphs tends to give specific information, analysis at the level of the whole graph will reveal higher level concepts. For each graph at a given time instance, a vector of features consisting of basic properties and derived properties can be built. Choosing the appropriate components of such a vector and its variation in time is an interesting area of research. Figure 3 illustrates the concept of the graph evolving and how the different graph properties change with time. Modeling such an evolving vector space and analyzing its behavior over time poses interesting challenges.

5.2 Rank Evolution

We analyze the evolution of the network graph at a single node level. For each node, we determine its rank based on its *Perpetrator Score(PScore)* and call it *Perpetrator Rank*. We then define another metric based on its *Perpetrator Rank(PR)* called *Perpetrator Height*. The height is a measure of 'how far' a node is from an infinitely low ranked node. For a node *i* at a time *t*, its *Perpetrator Height* can be expressed as:

$PHeight_{it}=\log_2(1+1/PR)$

Here we note that for a top ranked node, $PR=1$ and its $PHeight=1$. For a node with almost infinite rank, $PR=\infty$, and its *PHeight* would be zero. We then study rate of change in the rank of a node over time. The change for a time period Δt can be defined as:

$v = \Delta\ PHeight/\Delta\ t$

Since we are interested only in the change and not in a negative or a positive change in the rank (for the present work), we take the square of *v* for our analysis of how the node behaves. We also weigh the node according to the perpetrator score, *PScore*. We do this since a small change in a highly ranked node or a big change in a low ranked node is more interesting than a small or moderate change in a low ranked node. We can now define a quantity *Rank Energy* of a node as:

$Rank\ Energy = Weight * v^2$

This measure would be a good indicator of any rapid changes in the network behavior of machines. Such a rapid change would be of particular interest to the security analyst as it may indicate machines suddenly spamming or a mail server going down. Also, though we presently use *PScore* to weigh the node, the node can be weighed on other factors such as inside the network versus outside the network. The weight factor can be a vector of properties inherent to the node. The strength of the approach lies in its ability to detect anomalous behavior at an early stage.

6. Experimental Evaluation

Experiments were performed to evaluate two kinds of analyses. Firstly, we focused on identifying potential perpetrators given netflow data for a 10 minute time window. Second, we observed at a 3 hour time period and analyzed the rank evolution of each node. We discuss the details in the following subsections.

6.1 Analysis at a Single Time Instance

The first dataset was netflow data for the University for a 10 minute window from 07:10 to 07:20 hrs on June 17th, 2004. The total number of flows during this time period was 856470, with 228276 distinct IPs. Of these the number of connections that used SMTP protocol for E-Mail was 10368, with 1633 distinct IPs.

Using our approach described in section 4, we ranked the nodes according to their perpetrator scores. It was found that all main email servers were ranked low. Among those that were ranked on the top were, small e-mail severs that did not have traffic to the scale of the main e-mail servers. Most importantly, we were able to detect a machine, at address 134.84.S.44, that was known to be sending spam during that time period. This particular machine was ranked 2nd when ordered according to *Perpetrator Score.* We also noticed that once we remove the edges between the top hub and top authorities, a simple outdegree of the resultant graph also gave a fair measure of anomalous behavior. The rank of this machine according to authority scores was 1563, indicating that it was sending mails and not receiving them. The results are shown in Figure 6.

6.2 Analysis of Rank Evolution

The second dataset was netflow data for the University for a three hour time period from 7am to 10am on July 21st. We constructed graphs for each ten minute period, to obtain a set of eighteen graphs for this time period. The results are depicted in Figure 7.

We first generated *Perpetrator Scores* for each time instance and determined the rank of each node for that time period. The shading is a reflection of node rank. The top ranked node has a darker shade. Each column indicates one time period, and each row is an IP. For an IP not present in a time period we assign a default score of zero. Thus, the picture on the left indicates the variation of rank of the nodes. The last column is ranking of the node for the aggregated time period.

Total Flows: 856470
Email Flows: 10368
Distinct IPs (Total): 228276
Distinct IPs (Email): 1633

At this time, **134.84.S.44** was known to be sending spam. All of the other hosts were known, good email servers that were sending email

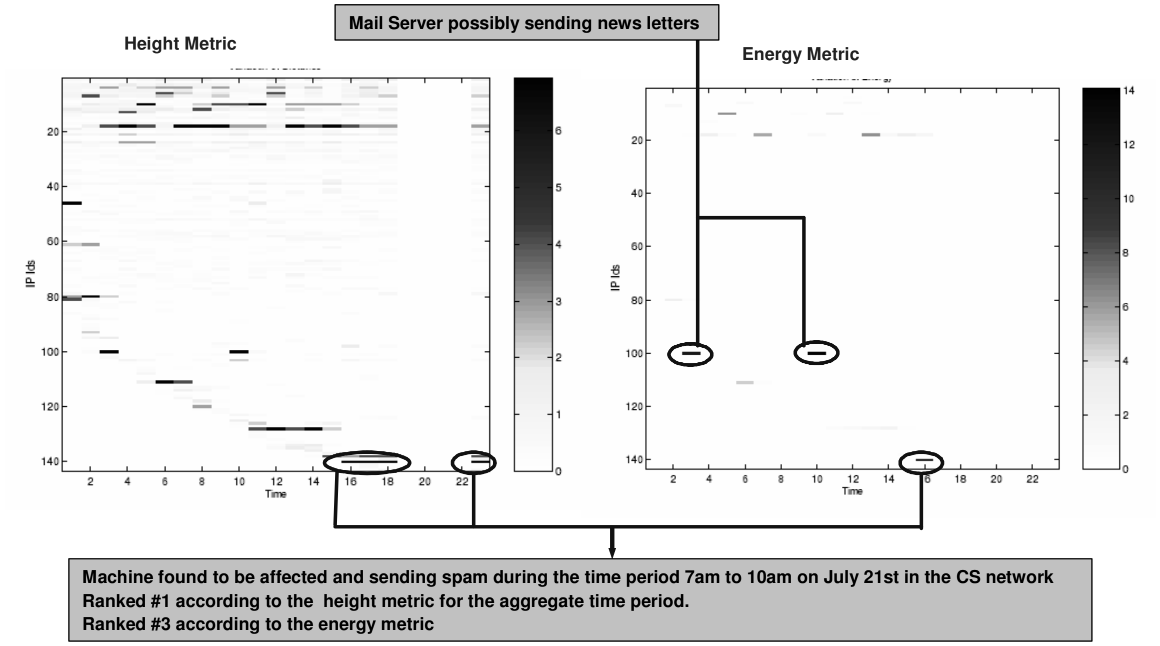
Sorted by Hub Score

IP Address	Authority Score	Hub Score
128.101.X.109	0	0.728289
134.84.S.44	**0**	**0.033964**
160.94.X.36	0	0.02685
160.94.X.35	0	0.02016
160.94.X.35	0	0.016173
160.94.X.36	0	0.014935
160.94.X.36	0	0.014778
128.101.X.119	0	0.013571
160.94.X.67	0	0.011118
160.94.X.33	0	0.010552
160.94.X.35	0	0.007896
160.94.X.33	0	0.006688
134.84.X.117	0	0.006529
128.101.X.10	0	0.005942
134.84.X.172	0	0.005282
134.84.X.4	0	0.005127
128.101.X.21	0	0.005016
128.101.X.1	0	0.004601
160.94.X.33	0	0.004492
160.94.X.100	0	0.004374

Sorted by Outdegree

IP Address	Indegree	Outdegree
128.101.X.109	0	363
160.94.X.36	1	176
134.84.S.44	**0**	**147**
160.94.X.35	1	112
160.94.X.36	1	106
160.94.X.36	1	103
128.101.X.119	0	99
160.94.X.35	1	92
160.94.X.35	1	60
160.94.X.33	0	45
160.94.X.33	0	45
160.94.X.33	0	36
128.101.X.10	0	33
134.84.X.4	0	28
134.84.X.2	0	26
128.101.X.2	0	26
134.84.X.172	0	25
160.94.X.11	0	24
160.94.X.34	0	22
128.101.X.104	0	21

Figure 6. Identifying Perpetrators



Mail Server possibly sending news letters

Machine found to be affected and sending spam during the time period 7am to 10am on July 21st in the CS network
Ranked #1 according to the height metric for the aggregate time period.
Ranked #3 according to the energy metric

Figure 7. Analysis of Rank Evolution

It can be seen that the sudden changes in the node ranks, for certain machines (such as mail server sending newsletters as shown in Figure 7), can be eclipsed by high change in one node, when computed for an aggregated time period.

In the second part, we computed the Rank Energy of each node by computing the change in the rank across consecutive time periods. This measure helps in eliminating most noise occurring due to changes in lesser important nodes in terms of anomaly behavior. The picture on the right depicts the energy of the nodes across the three hour time period.

7. Related Work

E-Mail Spamming has been a prominent area of research and different approaches have been taken to solve this problem. The two main class of problems studied have been 'spam email filtering' and 'detection and prevention of virus/worm intrusion and spreading'. Spam analysis can be broadly classified into content based techniques and flow statistics based techniques. There are commercial products that use signatures developed by analyzing the content [2,13]. Collaborative filtering approaches have also been developed by analyzing the content [3]. Classification based approaches that use heuristics or rules such as SpamAssasin [14] are also popular. MSN8[11] uses Bayesian based approaches to classify e-mails as spam. However, all these techniques have high privacy intrusiveness as they analyze the e-mail content.

Behavior based techniques such as the E-Mail Mining Toolkit [15] use user profiles to construct user cliques and analyze the e-mail attachment statistics for detection of e-mail worms or viruses. However, such techniques also need to obtain data at least at the mail server level and have a medium level of privacy intrusiveness. Sandvine Incorporated [12] suggests the use of behavior based techniques coupled with signature based techniques for detection of spam trojans. However, signature based methods fail to detect novel attacks at an early stage and such an approach would require looking into message content, raising privacy concerns. Also, the technical details of behavior based approach in the work are not clearly described.

Our goal in this work is not to identify individual users sending spam or classifying an individual email as a spam. Instead, we focus on detecting machines that are sending spam and we capture e-mail traffic that does not necessarily pass through an e-mail server or use a particular user id or a mail client. Compared to 'receiver based' approaches such as content filtering, and 'sender based' approaches such as IP blocking; our approach is in the complementary area of 'transport based' approaches where the e-mail is suppressed by stopping the mis-behaving mail system machine. In addition to being less privacy intensive, we believe this is also a new and complementary approach to spam reduction.

8. Conclusions

We have presented in this paper the different levels of privacy involved in analyzing e-mail behavior. We have proposed an approach to detect anomalous behavior in E-Mail traffic at the network level, with *low privacy intrusiveness*. Finally, we have presented a framework for studying evolving graphs and how it can be applied to network traffic for *early detection* suspicious behavior. We have restricted our work to a level of single node for the present work.

Further research in this area would be to develop models and measures to mine information from evolving graphs at the level of subgraphs and whole graphs.

9. Acknowledgements

We would like to thank Prof. Vipin Kumar and MINDS Research group at the Department of Computer Science for providing valuable suggestions. We would also like to thank Paul Dokas for providing the results for the University of Minnesota network. This work was been partially supported by the ARDA Agency under contract F30602-03-C-0243 and Army High Performance Computing Research Center contract number DAAD19-01-2-0014. The content of the work does not necessarily reflect the position or policy of the government and no official endorsement should be inferred. Access to computing facilities was provided by the AHPCRC and the Minnesota Supercomputing Institute.

10. References

[1] A.L. Barabasi, "*Linked: The New Science of Networks*. Cambridge, Massachusetts: Perseus Publishing, 2002.

[2] BrightMail, http://www.brightmail.com/

[3] CloudMark, http://www.cloudmark.com/

[4] C.Cowan, P.Wagle, C.Pu, S.Beattie, and J. Walpole, "Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade", DARPA Information Survivability Conference and Expo (DISCEX), Hilton Head Island SC, January 2000.

[5] P.Desikan, J. Srivastava, V. Kumar, P.-N. Tan, "Hyperlink Analysis – Techniques & Applications", Army High Performance Computing Center Technical Report, 2002.

[6] P.Desikan, J. Srivastava, “Mining Temporally Evolving Graphs”, WebKDD 2004, Seattle.

[7] L.Ertoz, ,E. Eilertson, A. Lazarevic, A., P.Tan, J. Srivastava, V. Kumar, P. Dokas, The MINDS - Minnesota Intrusion Detection System, "Next Generation Data Mining", MIT /AAAI Press 2004.

[8] J. Goodman, G. Hulten, “Junk E-mail Filtering”, Tutorial , KDD 2004

[9] J.M.Kleinberg, “Authoritative Sources in Hyperlinked Environment”, 9th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 668-667, 1998.

[10] V.Krebs, “Data Mining Email to Discover Social Networks and Communities of Practice”, http://www.orgnet.com/email.html, 2003

[11] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A Bayesian Approach to Filtering Junk E-mail” Learning for Text Categorization: Papers from the 1998 Workshop.

[12] Sandvine Incorporated, “Trend analysis: Spam trojans and their impact on broadband service providers”,http://www.sandvine.com/solutions/pdfs/spam_trojan_trend_analysis.pdf, June 2004

[13] O.Sheyner, J.Haines, S.Jha, R.Lippmann, and J. M. Wing, “Automated Generation and Analysis of Attack Graphs”, IEEE Symposium on Security and Privacy , April 2002.

[14] SpamAssassin, http://spamassassin.apache.org/

[15] S.J. Stolfo, et al. "A Behavior-based Approach to Securing Email Systems". "Mathematical Methods, Models and Architectures for Computer Networks Security", Proceedings published by Springer Verlag, Sept. 2003

[16] SurfControl http://www.surfcontrol.com/