International Journal of Science Engineering and Advance Technology

Anomaly Detection on Firewall Logs from Multifold System Based on Email Classification

*Pothuru Alekhya
Assistant professor, Raghu Institute of Technology,
Visakhapatnam, alekhya.pothuru@gmail.com

Abstract

The greater part of the current frameworks sort the report or firewall logs-corpus in light of the term similitude by discover the archive term relationship. It can't distinguish the calculated comparability or connection among them. Be that as it may, proposed framework concentrates on both term savvy and additionally reasonable shrewd comparability to discover the firewall logs insights to tag the firewall logs with appropriate sort. Classification of firewall logs information is multi crease in proposed framework. Real phases of proposed approach are portrayed in the accompanying segment. Order of the messages in light of their applied closeness as opposed to visually impaired term savvy likeness alongside firewall logs header, html substance and connection investigation.

Key words- Anomaly detection, Firewall logs, virus.

Introduction

For the previous few years, we've got determined an eternal increase of malicious traffic within the web in style of distributed denial of service attacks, virus and worm's propagation, intrusions, etc. actually recent studies ([2], [4]) have disclosed the vital rise in malicious traffic volume within the entire web. This rise is during a large proportion caused by the propagation within the web of worms like CodeRed [3] [5], Nimda [5], the Slammer worm [6], Msblaster and Funlove. Consequently, defensive networks against such malicious traffic may be a day by day incessant activity for network operators. Plenty of techniques are developed so as to discover, determine and stop propagation of malicious traffic over networks. we tend to differentiate between 2 categories of intrusion detection techniques: Misuse Detection and Anomaly Detection. TheMisuse Detection Systems attempt to discover intrusions by comparison the present activity of the audited resource to a information of noted attack eventualities. Those techniques can't discover unknown attacks. However, the Anomaly Detection Systems (ADS) attempt to discover intrusions by comparison the present activity of the audited resource to a longtime "normal activity" diagrammatical in style of a profile. The bulk of those techniques ought to keep per-connection or per-flow state over one link or node. Thus, they have to be wide deployed altogether nodes so as to be effective. What are more they need plenty of computing resources creating their value unaffordable for several ISPs? During this work, we tend to tried to develop AN anomaly discover tool able to detect attacks while not keeping a per flow state. This tool doesn't commit to determine the various styles of attacks or their origins. Thus it is often helpful as a first line anomaly detection tool. In fact, this tool is often wont to indicate once an additional refined intrusion detection system, supported per-flow knowledge assortment, should be started. In fact, we tend to developed AN Anomaly Detection System (ADS) derived from the anomaly detection schema bestowed and supported SNMP knowledge. When evaluating this technique against some common attacks, we tend to exploit it for the detection of traffic anomalies within the TNUN network. Finally we tend to studied some temporal patterns of network traffic anomalies. This paper is organized as follow. First, within the second section, we tend to discuss previous connected work. Within the third section, we tend to describe the anomaly detection technique employed by our ADS system. Then we tend to gift the analysis methodology and discuss analysis results. In section four we tend to describe the TNUN network. After that, we tend to discuss some temporal characteristics of traffic anomalies within the TNUN network within the fifth section. Finally we tend to conclude with an outline of the themes developed throughout our study.

This paper mainly focus on the analysis of data generated through server logs detection and possible forecast of anomalies associated to the monitored servers. The issue is applicable in many mission-critical systems contains various servers.

There it is favorable to be able sense and even predict problems to be able to react promptly and apply required corrections to the system. We have had at our discarding log data recorded from real operating server machines. The data contain a range of attributes correlated to the use of CPU cores, memory, disks and file system as well as processes. Solid example of the attributes analyzed includes CPU usage, amount of free available memory and number of processes. The indicator values are store with definite time intervals, e.g., one minute intervals. This may limit in some cases the ability to detect anomalies they may be get hidden due to the average of the indicator values over the time interval. The anomalies happening in the system could be due to several different reasons including programming errors (software errors), component failures, high load, and manpower errors. Some indicate the values scaled directly a property that we are accepted.

PROPOSED APPROACH

Proposed system focuses on both term wise as well as conceptual wise similarity to find the firewall logs statistics to tag the firewall logs with suitable type Categorization of firewall logs data is multi fold in proposed system. Major stages of proposed approach are described in the following sections.

Machine Learning

Machine learning is a process of giving knowledge to the proposed learner. For that purpose initially proposed system is trained with some existing firewall logs corpus with heterogeneous content. A novel unsupervised machine learning algorithm named as “Adaptive Hierarchical Leader follower with Evolutionary computing algorithm” is used by the proposed system for effective handling of concept drift firewall logs streams. Before enter into details of algorithm we need to finish some pre processing steps described as follows.

PRE PROCESSING STEPS

Firewall logs pre-processing having following stages:

Firewall logs Header processing

Generally firewall logs header contains following information: From, Subject, Date, To, Return-Path, Envelop-To, Delivery Date, Received, Message-id, Content-type, Content-Length, X-spam status and Message Body.

Among these fields proposed system utilizes Subject, Date, To, Delivery Date, Message Id, Received, Content type, Content length and Message Body fields for firewall logs statistics and analysis.

Initially Historical or training firewall logs corpus is loaded and extract the above mentioned fields from each firewall logs. Then these fields are store as a transaction table in the database for offline or online querying. For this transaction table a view is created with specified group by statements. This view is used to get frequency of particular path and content. So it is more helpful to filter the mails in basic level. Based on this statistics initial decision is made and mark some of the firewall logs paths or ids are susceptible. Later body processing stage is initiated.

Body processing

In this stage body is identified based on the content type. It may be html, plain or other MIME type. Based on the content type process is also divided. It is having two stages of work again.

1. HTML Processing: In this body text is scanned for <A> and <SCRIPT> tags. Because, in most of the cases hyperlinks and script elements causes to phishing and virus injection. Every hyper link is basically scanned to identify the source of that link. This can be achieved through processing of host part of link with “trace route” like tools. The results of the “trace route” are compared with”phish tank” database. If any one of the address in the trace route belongs to phish tank database, proposed system removes that hyper link. This phish tank database is publicly available.
2. Text processing of firewall logs body includes number of stages. This is heart of the conceptual document mining. It includes some natural language processing steps. They are described in following modules. Body is identified with <P> tag as well as <TD>tag. After that all the individual <P> [<TD>] plain text parts are merged into single document for each firewall logs and placed in a firewall logs body store locally. Here we assume that this local store is maintained in the firewall system.

Conceptual Document Processing

This stage is executed in parallel by utilizing recent advancements in processor technology. The following procedure is executed in parallel. Before this procedure to be executed entire firewall logs body corpus is divided into multiple partitions. Each Partition is submitted to different processors as multiple Fork-Join Tasks. In every partition procedure executed as follows.

- Each document is split into multiple sentences
- Each Sentence is split into multiple terms.
- Each term is applied to stemming algorithm such as Porter Stemming Algorithm [1]. This

algorithm finds the stem word the given term to get the generalization of that term.

- To find the conceptual similarity there is a need to identify the negation phrases. For example, “Rama is not bad boy” means “Rama is good boy”. According to the example presented both of the sentences are conceptually similar. If we only consider term wise similarity then they may be treated as different. To overcome such type of problems, identify the negation words in each sentence and try to replace it with antonym words. For this purpose we have deployed WordNet [6] tool with Java-WordNet-Interface developed by MIT Team members. It will return not only antonyms but also synonyms, hypernyoms and hyponyms those are also used in our text processing.
- Similarly some words or sentences are also conceptually exhibit similar meaning. For example, “you have to pay your bill” may conceptually equal to “you are supposed to pay the bill”. Other example could be, “you need to submit documents” conceptually equals to “you required submitting document”. To achieve this proposed system find the synonyms, hypernyoms and hyponyms of each term and compared with the respective list. But this is not that much of scalable and memory efficient solution to maintain all the terms and their synonyms in the list in the memory.

For this purpose Synset ids acquired from WordNet are maintained in a hash map. Every a new word found, the respective index word will be found and related synset id will be retrieved. If indexed word id under synset id is found process will ignore that word otherwise it will insert the word into hash map. This is ongoing process till the end of stream. This type of mechanism required to adaptively cluster the documents based on the conceptual similarity.

PROPOSED SYSTEM ARCHITECTURE

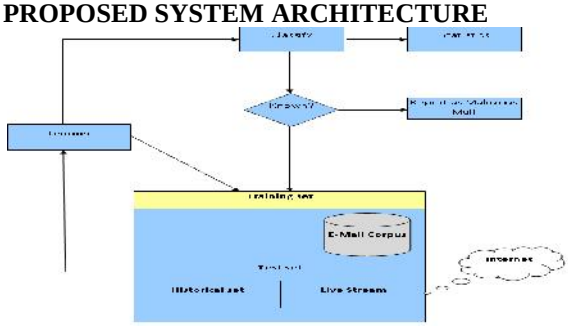

Figure 1: System Architecture

Sometimes some words are treated as nouns, verbs or adjectives. In that case, there is a need to consider the sense of those words in the given sentences for given context. This can also be achieved through auto tagging or words in a given sentence with natural language processing tools such as OpenNLP [3]. It has as POS (Parts-of-Speech) utility to know the parts-of-speech of term.

Finally document vectors are also generated and normalized by other processes simultaneously. Initially document vectors are generated with equal size term frequencies. Document-Term joint probability matrix is also generated prior to the clustering.

RESULTS

Before starting the process to build data base through queries which is process to run a file from sqlite server. After completion of building the data base we have to identify what are the source and destination ip’s along with this we have to check what are the ip’s accepted by the system and denied the system along with sending and receiving statistics. Finally we have to show two clusters which ip’s are denied, malicious traffic in the network.

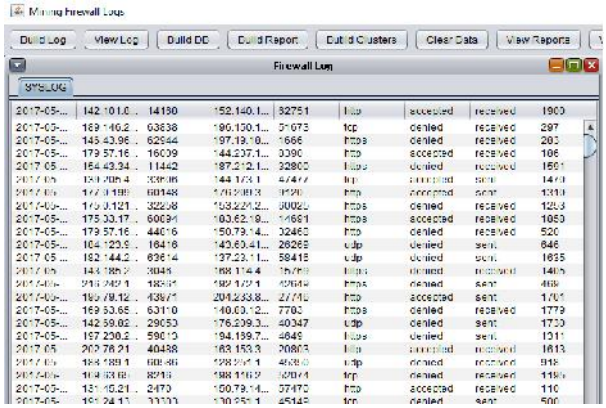

Figure 2: System Log file which we are build through sqlite

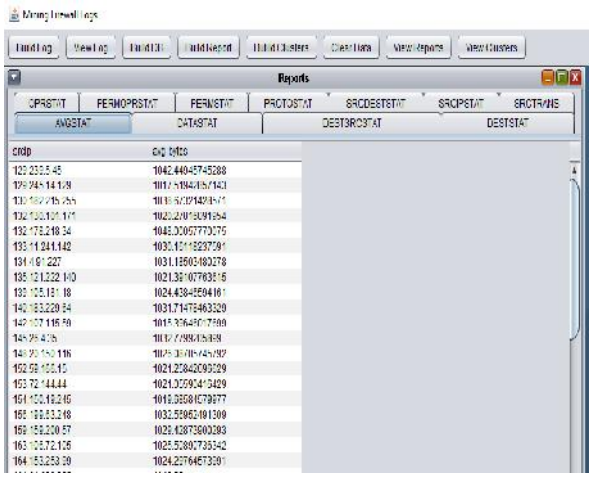

Figure 3: System average statistics of source IP address

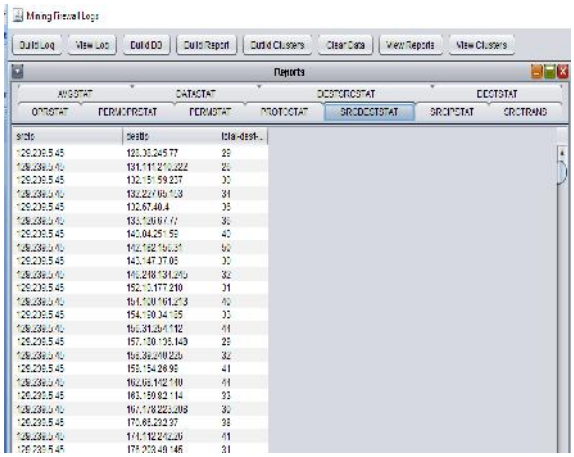

Figure 4: System average statistics of source and destination IP address

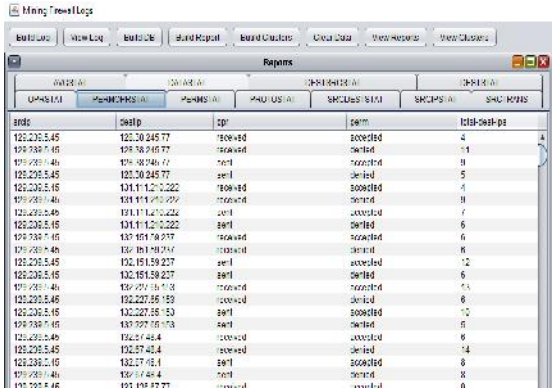

Figure 5: System performance report with notification

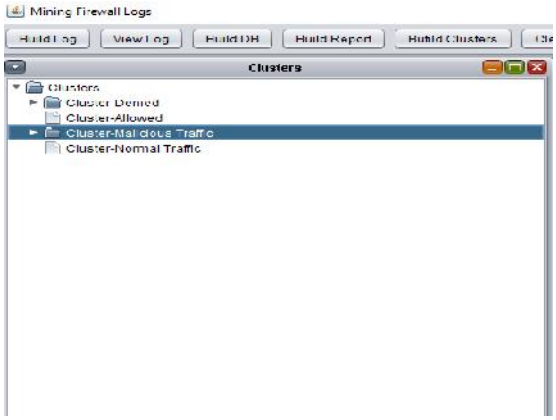

Figure 5: Clusters based on notification

CONCLUSION

The process described in this paper were used rapidly identify a number of machines associated with anomalous network activity. A significant number of the identified machines were victims of nefarious activity or mis-configuration. The ability to control these issues is a challenge for network, server and device management in part because of the vast amount of log data involved. Through the use of data mining processes and techniques we have demonstrated an accurate way of handling burdensome firewall logs. In this scope of this effort a full analysis of the technique could not be performed. However, additional analysis is warranted and may lead to promising results. No effort was made to identify false negatives by correlating classification with other systems that may detect similar activity, such IDS or net flow systems. This would be an important part of future investigation into the utility of applying these techniques to firewall log data. There may also be significant benefit to correlating the features derived from the firewall logs with other features obtained from net flow and IDS data. This may lead to a richer feature vector that might have greater utility than the data acquired during the project.

REFERENCES

[1] Winding, Robert, Timothy Wright, and Michael Chapple. "System anomaly detection: Mining firewall logs." Securecomm and Workshops, 2006. IEEE, 2006.

[2]. D. Moore, G. Voelker, and S. Savage: Inferring Internet Denial of Service activity. In Proceedings of the 2001 USENIX Security Symposium, Washington DC, August 2001. 5.

[3]. D.Moore, C.Shannon and J.Brown: Code-Red: a Case Study on the Spread and Victims of an Internet Worm. In Internet Measurement Workshop (IMW); 2002

[4]. R. Pang, V. Yegneswaran, P. Barford, V. Paxon, L. Peterson: Characteristics of Internet Background Radiation. In IMC'04, Italy, October 2004.

[5]. S. Staniford, V. Paxson and N. Weaver: How to 0wn the Internet in Your Spare Time, In Proc. USENIX Security Symposium 2002.