RESEARCH ARTICLE OPEN ACCESS

Anomaly Extraction Using Association Rule Mining

Ms. Gargi Joshi*
* (Department of Computer Engineering, Dr. D.Y Patil College of Engineering, Ambi, Pune)

Abstract
Today network security, uptime and performance of network are important and serious issue in computer network. Anomaly is deviation from normal behavior which is factor that affects on network security. So Anomaly Extraction which detects and extracts anomalous flow from network is requirement of network operator. Anomaly extraction refers to automatically finding in a large set of flows observed during an anomalous time interval, the flows associated with the anomalous event(s). It is important for root cause analysis, network forensics, and attack mitigation and anomaly modeling. We use meta data provided by several histogram based detectors to identify suspicious flows, and then apply association rule mining to find and summarize anomalous flows. Using Histogram based detector to identify anomalies and then applying Association rule mining, anomalies will be extracted. Apriori and FP Growth algorithm will be used to generate the set of rule applied on metadata. Using traffic data from a network this technique effectively finds the flow associated with the anomalous event(s). it triggers a very small number of false positives, which exhibit specific patterns and can be sorted out by an administrator this anomaly extraction method significantly reduces the work hours needed for analyzing alarms, making anomaly detection systems more practical.

Keywords **–** Anomaly Extraction, Association Rules, computer network, data mining, Apriori Algorithm,FP Growth Algorithm.

I. INTRODUCTION

An anomaly detection system may provide meta-data relevant to an alarm that help to narrow down the set of candidate anomalous flows. For example, anomaly detection systems analyzing histograms may indicate the histogram bins that an anomaly affected, e.g., a range of IP addresses or port numbers. Such meta-data can be used to restrict the candidate anomalous flows to these that have IP addresses or port numbers within the affected range. To extract anomalous flows, one could build a model describing normal flow characteristics and use the model to identify deviating flows. However, building such a microscopic model is very challenging due to the wide variability of flow characteristics. Similarly, one could compare flows during an interval with flows from normal or past intervals and search for changes, like new flows that were not previously observed or flows with significant increase/decrease in their volume. Such approaches essentially perform anomaly detection at the level of individual flows and could be used to identify anomalous flows.

Anomaly detection techniques are the last line of defense when other approaches fail to detect security threats or other problems. They have been extensively studied since they pose a number of interesting research problems, involving statistics, modeling, and efficient data structures. Nevertheless, they have not yet gained widespread adaptation, as a number of challenges, like reducing the number of false positives or simplifying training and calibration, remain to be solved.

II. PROBLEM DEFINITION

Aim of this system is to identify an anomaly from the network traffic during a time interval and find the flows associated with the event(s) that triggered an observed anomaly.

2.1 EXISTING SYSTEM

Identifying network anomalies is critical for the timely mitigation of events, like attacks or failures that can affect the security and performance of network. Traditional approaches to anomaly detection use attack signatures built in an Intrusion Detection System (IDS) that can identify attacks with known patterns. Significant research efforts have focused on building IDS's and, therefore, related production systems are presently employed in many networks. Although signature-based detection finds most known attacks, it fails to identify new attacks and other problems that have not appeared before and do not have known signatures.

A number of studies have focused on developing volume-based anomaly detection schemes [2]–[7]. For example, Barford et al. [2] used wavelets to distinguish between predictable and anomalous traffic volume changes. More recently, Zhang et al. [6] introduced a general framework that aims to identify anomalies from network-wide link load traffic data. These studies are successful in identifying anomalies that result in (network-wide) traffic volume

deviations. However, they are not so effective in detecting stealth attacks, such as low-rate port scanning, that do not result in notable traffic volume changes.

The anomaly detection scheme by Guet al. [10] uses a single composite feature distribution to characterize traffic and computes a parametric model of the distribution using training data. Observed network traffic is, then, compared to the constructed model to identify anomalies. The authors assume that the training data-set does not contain any anomalies. The proposed anomaly detection scheme uses Principal Component Analysis (PCA) to identify an orthogonal basis along which the measurement data exhibit the highest variance. The principal components with high variance model the normal behavior of a network, whereas the remaining components of small variance are used to identify and classify anomalies. The proposed scheme aims at finding anomalies in large backbone networks and, consequently, aggregates traffic into origin-destination (OD) flows between network ingress and egress points. But it is hard to select the right number of principal components to achieve: 1) a low false-positive rate and 2) a subspace of PCA components that is anomaly-free.

2.2 PROPOSED SYSTEM

Our system contains three different phases. One is histogram detector that will observe the network traffic and alert the system upon anomaly detection. Second phase consists of histogram cloning which assures the anomaly detection and finds the suspicious flows from network traffic. Finally third phase is to apply association rule mining algorithm i.e. FP Growth to find the frequent item sets.

Process Summary:
1] Form network between computers or laptops.
2] Histogram detector will observe network for certain interval.
3] On anomaly detection form clones of histogram and find suspicious flows in network.
4] Apply FP Growth algorithm to this suspicious flows.
5] Find frequent item sets from the set of suspicious flows.

We build a histogram-based detector for our evaluation that uses the Kullback–Leibler (KL) distance to detect anomalies. Each histogram detector monitors a flow feature distribution, like the distribution of source ports or destination IP addresses. We assume n histogram-based detectors that correspond to n different traffic features and have each m histogram bins.
As an alternative to arbitrary binning, we introduce histogram cloning. With histogram cloning, different clones provide alternative ways to group feature values into a desired number of bins/groups creating effectively additional views along which an anomaly may be visible. The cloning mechanism is coupled with a simple voting scheme that controls the sensitivity of the detector and eventually affects a tradeoff between false positives and negatives.

Assume a time interval with an anomaly. Pre filtering selects all flows that match the union of the meta-data Vj provided by n detectors, i.e., all flows that match where are filtered. Pre filtering usually removes a large part of the normal traffic. This is desirable for two reasons. First, it generates a substantially smaller dataset that results in faster processing in the following steps. Second, it improves the accuracy of association rule mining by removing flows that could result in false-positive item-sets.

Here, we apply the first step of association rule mining, i.e., we find frequent item-sets to extract anomalous flows from a large set of flows observed during a time interval. The standard algorithm for discovering frequent item-sets is the Apriori algorithm.

III. OBJECTIVE

Identifying an anomaly from the network traffic during a time interval and find the flows associated with the event(s) that triggered an observed anomaly.

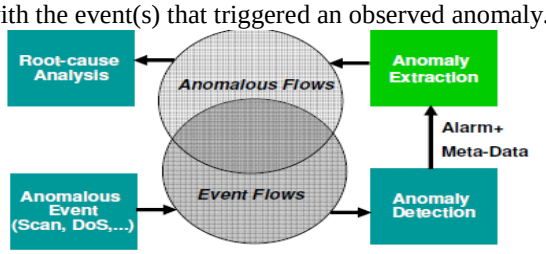

FIG.1 ANOMALY EXTRACTION PATH

In this project, we are observing the network traffic for time interval t and identifying the anomaly using histogram detector. Upon detection of anomaly, we build the clones of histogram detector and find suspicious flows that causes anomaly in the network. We then filter this data to eliminate large fraction of normal flows. A summary report of frequent item-sets in the set of suspicious flows is generated by applying association rule mining.

IV. LITERATURE SURVEY

F, Silveira and Diot [1] introduced a tool called URCA that searches for anomalous flows by iteratively eliminating subsets of normal flows. URCA also classifies the type of a detected anomaly. Nevertheless, it requires to repeatedly evaluating an anomaly detector on different flow subsets, which can be costly. Compared to this work, we simply computing frequent item-sets on pre filtered flows is sufficient to identify anomalous flows.

Do Witcher [2] is a scalable system for worm detection and containment in backbone networks. Part of the system automatically constructs a flow-filter mask from the intersection of suspicious attributes (meta-data) provided by different detectors leverage suspicious attributes from an anomaly detector and study the anomaly extraction problem in more depth. We highlight that using the intersection can miss anomalous flows and find that the union of the meta-data combined with association rule mining gives better results.

Dewaele et al. [3] use sketches to create multiple random projections of a traffic trace, then model the marginal's of the sub traces using Gamma laws and identify deviations in the parameters of the models as anomalies. In addition, their method finds possible anomalous source or destination IP addresses by taking the intersection of the addresses hashing into anomalous sub traces.
Compared to this work, we introduce and validate techniques to address the more challenging problem of finding anomalous flows rather than IP addresses.

Lakhina et al. [4] use SNMP data to detect network-wide volume anomalies and to pinpoint the origin-destination (OD) flow along which an anomaly existed. In contrast, our approach takes as input a large number of flow records, e.g., standard 5-tuple flows, and extracts anomalous flows. An OD flow may include millions of both normal and anomalous 5-tuple flows and, therefore, can form the input to our methodology.
Li et al. [5], use sketches to randomly aggregate flows as an alternative to OD aggregation. The authors show that random aggregation can detect more anomalies than OD aggregation in the PCA subspace anomaly detection method. In addition, the authors discuss how their method can be used for anomaly extraction. However, the work and evaluation focus primarily on anomaly detection.

Lee and Stolfo [6] show how association rules can be used to extract interesting intrusion patterns from system calls and tcp dump logs.
Vaarandi [7] introduces a tool called LogHound that provides an optimized implementation of Apriori and demonstrates how LogHound can be used to summarize traffic flow records.
Yoshida et al. [8] also use frequent item-set mining to identify interesting events in traces from the MAWI traffic archive.

Li and Deng [9] outline a variant of the Eclat frequent item-set mining algorithm] that operates in a sliding window fashion and evaluate it using traffic flow traces from a Chinese university.
Chandola and Kumar [10] describe heuristics for finding a minimal set of frequent item-sets that summarizes a large set of flows.

Mahoney and Chan [11] use association rule mining to find rare events that are suspected to represent anomalies in packet payload data. They evaluate their method on the 1999 DARPA/Lincoln Laboratory traces. Their approach targets edge networks where mining rare events is possible. In massive backbone data, however, this approach is less promising. Another application of rule mining in edge networks is eXpose, which learns fine-grained communication rules by exploiting the temporal correlation between flows within very short time windows.

Compared to these studies, association rule mining can be combined with anomaly detection to effectively extract anomalous flows. Hierarchical heavy-hitter detection methods [10], [7] group traffic into hierarchical clusters of high resource consumption and focus primarily on optimizing computational performance for summarizing normal traffic. For example, they have been used to identify clusters of Web servers in hosting farms. Hierarchical heavy-hitter detection is similar to frequent item-set mining in that both approaches find different forms of multidimensional heavy hitters. Compared to these studies, intelligently combining multidimensional heavy-hitters with anomaly detection enables us to extract anomalous flows. In addition, frequent item-set mining scales to higher dimensions much better than existing hierarchical heavy-hitter detection methods. Finally, substantial work has focused on dimensionality reduction for anomaly detection in backbone network. These papers investigate techniques and appropriate metrics for detecting traffic anomalies, but do not focus on the anomaly extraction problem which we are addressing in this project.

V. MATHEMATICAL MODEL

1] U is main set of users (ATM Holders) like u1, u2, u3….
U = {u1, u2, u3…….}
2] A is main set of Administrators like a1, a2, a3….
A = {a1, a2, a3…….}
3] C is the main set of histogram clones like c1, c2, c3....
C = {c1, c2, c3.....}
4] Identify the processes as P.
P = {Set of processes}
P = {P1, P2, P3……}

If (anomaly is detected in the network) then
P1 = {e1, e2, e3, e4}
Where
{e1=i|i is to build c number of clones}
{e2=j|j is to find anomalous bins from histogram}

{e3=k|k is to filter suspicious data}
{e4=l|l is to find frequent item sets from given suspicious data}
Else
P1 = {e1, e2}
Where
{e1=i|i is to observe network traffic during time interval t}
{e2=j|j is to check whether anomaly detects or not}

VI. PROJECT SETUP

Operating Environment:-

a) S/W Specification

Operating System : Windows 7.
Development End : JAVA [JDK 1.6]
IDE : Eclipse Helios
Tool : JCreator

b) H/W Specification

Processor : PIV– 500 MHz to 3.0 GHz.
RAM : 1GB.
Disk : 20 GB.
Monitor : Any Color Display.
Key Board : Standard Windows Keyboard

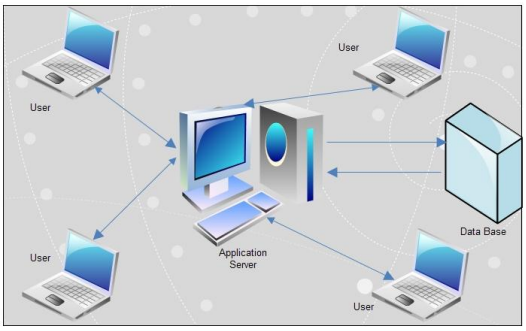

Fig.2 System Architecture

VII. DEVELOPMENT METHODOLOGY

7.1 MODULE INFORMATION

Module1:
Design the Graphical User Interface (GUI) for our system with client and server.

Module2:
Build histogram detector to observe the traffic on the network and detect anomalies.

Module3:
Find suspicious flows from the network traffic that causes anomaly in the network.

Module4:
Implement Apriori and FP Growth algorithm to find frequent item sets
Approach Overview (3 steps)

***7.1.1 Detection**:*
Use a number of histogram-based detectors:

1. Identify affected bins and create set V of corresponding feature values
2. Use histogram cloning to reduce collisions and false positives

***7.1.2 Filtering**:*
Filter flows that match union of meta-data provided by N detectors

1. Filtered flows are called „suspicious“ flows

***7.1.3 Mining**:*
Use association rules to extract and summarize anomalous flows from the set of suspicious flows

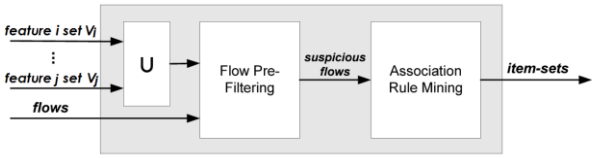

Fig.3 Anomaly Extraction Steps

7.2 Association Rule Mining

Given a a number of itemsets, find frequent subsets which are common to at least a minimum number s of the itemsets. An itemset is a flow (7-tuple): {srcIP, dstIP, srcPort, dstPort, proto, #packets, #bytes}
Key intuition: anomalies trigger a large number of flows with one or more common feature values, e.g., src IP addr, dst port, #packets. Use modified Apriori algorithm to find frequent subset

VIII. CONCLUSION

We are implementing FP Growth algorithm and Apriori algorithm to find out frequent item sets. We will compare the results of Apriori algorithm and FP Growth algorithm and show how FP Growth algorithm achieves better results in reducing the time and space complexity and provides better optimization results as compared to Apriori Algorithm. Implementation by FP Growth Algorithm will be the extension to our work.

The proposed methodology is very useful for finding the root cause of detected anomalies, which helps in anomaly mitigation, network forensics and anomaly modeling.

REFERENCES

[1] F. Silveira and C. Diot, “*URCA: Pulling out anomalies by their root causes*,” in Proc. IEEE INFOCOM, Mar. 2010, pp. 1-9.
[2] S. Ranjan, S. Shah, A. Nucci, M. M. Munafò, R. L. Cruz, and S.M Muthukrishnan, “*Dowitcher: Effective worm detection and containment in the Internet core*,” in Proc. IEEE INFOCOM, 2007, pp.2541-2545.
[3] G. Dewaele, K. Fukuda, P. Borgnat, P. Abry, and K. Cho, “*Extracting hidden anomalies using sketch and non Gaussian multi resolution statistical detection procedures*,”

in Proc. LSAD, 2007, pp. 145-152. [10]

[4] A. Lakhina, M. Crovella, and C. Diot,"Diagnosing network-wide traffic anomalies," in Proc. ACM SIGCOMM, 2004, pp. 219-230.

[5] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, "Detection and identification of network anomalies using sketch subspaces," in Proc. 6th ACM SIGCOMM IMC, 2006, pp. 147-152.

[6] W. Lee and S. J. Stolfo, "*Data mining approaches for intrusion detection*," in Proc. 7th USENIX Security Symp., 1998, vol. 7, p. 6.

[7] R. Vaarandi, "*Mining event logs with SLCT and LogHound*," in Proc.IEEE NOMS, Apr. 2008, pp. 1071-1074.

[8] K. Yoshida, Y. Shomura, and Y. Watanabe "*Visualizing network status*," in Proc. Int. Conf. Mach. Learning Cybern., Aug. 2007, vol.4, pp. 2094-2099.

[9] X. Li and Z.-H. Deng, "*Mining frequent patterns from network flows for monitoring network,*" Expert Syst. Appl. vol. 37, no. 12, pp.8850-8860, 2010.

[10] V. Chandola and V. Kumar, "*Summarization—Compressing data into an informative representation*," Knowl. Inf. Syst., vol. 12, pp. 355-378, 2007.

[11] M. V.Mahoney and P. K. Chan, "*Learning rules for anomaly detection of hostile network traffic*," in Proc. 3rd IEEE ICDM, 2003, pp.601-604.