Network Anomaly Detection Method in combination with Intelligent Sampling

Ankur Romesh Batta, Vani Joginipelli, Phani Shashank Kudaravalli
George Mason University
abatta@gmu.edu, vjoginip@gmu.edu, pkudarav@gmu.edu

Abstract

This paper presents research in the area of improving Network Anomaly Detection effectiveness by using Network Anomaly Detection Methods in combination with Intelligent Sampling. The paper researches the impact of Intelligent Sampling on Anomaly Detection. The Intelligent Flow sampling can both reduce the data for processing and improve the effectiveness of anomaly detection. The research proposes the use of the Principal Component Analysis (PCA) Anomaly Detection Method in combination with Selective Sampling and Smart Sampling Flow sampling techniques. Principal Component Analysis (PCA) is used in a variety of domains to reduce the number of dimensions in input sets without losing the "information" contained in them. PCA is particularly helpful in anomaly detection since it can reduce the data dimensionality into a smaller set of independent variables. At its core, PCA produces a set of principal components, which are orthonormal Eigen value/eigenvector pairs. In other words, it projects a new set of axes which best suit the data. These set of axes represent the normal connection data. Anomaly/Outlier detection occurs by mapping live network data onto these 'normal' axes and calculating the distance from the axes. If the distance is greater than a certain threshold, then the connection is classified as an attack. The team evaluated the different types of existing Sampling techniques (Packet Sampling and Flow Sampling) and the different existing Anomaly Detection Methods and came to the conclusion that Principal Component Analysis (PCA) Anomaly Detection Method in combination with Flow Sampling is more effective than other combinations of Sampling and Anomaly Detection Methods. The team also researched the type of Flow Sampling (Selective Sampling and Smart Sampling) that is suitable for detecting particular types of anomalies.

1. Introduction

Network anomaly detection techniques rely on the analysis of network traffic and the characterization of the dynamic statistical properties of traffic normality in order to accurately and timely detect network anomalies. Anomaly detection is based on the concept that perturbations of normal behavior suggest the presence of anomalies and/or attacks/faults.

Today traffic measurement, anomaly detection and analysis have become more complicated with the continuously increasing network traffic. It has become difficult to store and process all network traffic flow information with limited number of resources and as a result sampling becomes an essential component of scalable Internet monitoring.

Sampling is the process of making partial observations of a system of interest, and drawing conclusions about the full behavior of the system from these limited observations. It is important to minimize information loss while reducing the volume of collected data in order to make the corresponding processes realizable. The way in which the partial information is transformed into knowledge of the system is of high research and practical importance for developing efficient and effective anomaly detection techniques. In this article we have researched on using intelligent sampling techniques for anomaly detection.

Intelligent sampling is required to obtain a reliable estimate of detailed information from only a subset of flow records. It exploits [2] the fact that for specific-purpose applications such as anomaly detection, a large fraction of information is contained in a small fraction of flows. Hence, by using sampling techniques that opportunistically/intelligently and preferentially sample traffic data, we can achieve "magnification" of the appearance of anomalies within the sampled data set and therefore improve their detection. Choosing the right sampling techniques and effectiveness of the

chosen technique on the anomaly detection is important.

There are various Anomaly Detection Methods that can be used in combination with different types of Sampling. This paper suggests the Anomaly Detection Method and the Sampling Type combination that it finds is more effective than other combinations.

2. Research problem

Network performance [2] and traffic monitoring has become essential for managing a network efficiently and ensuring reliable operation. However, due to the large number of flows on a high-capacity link and the corresponding difficulty in storing and processing all flow information with a limited amount of resources, sampling has attracted a great deal of attention as a way to collect statistical information about flows. Several studies have been devoted to analyzing and evaluating the tradeoffs between sampling accuracy and efficiency, which essentially refer to the issues of minimizing information loss while reducing the volume of collected data in order to make the corresponding processes realizable.

Anomaly detection is based on the concept that perturbations of normal behavior suggest the presence of anomalies and/or attacks. Network anomaly detection techniques rely on the analysis of network traffic and the characterization of the dynamic statistical properties of traffic normality in order to accurately and quickly detect network anomalies. In this article, we have researched the impact of sampling on anomaly detection.

This paper researches the two different types of Sampling techniques, Flow-based Sampling and Packet-based sampling to find out which is more effective and at the same time less resource intensive. It also researches the two types of Flow Sampling techniques- Selective Sampling and Smart Sampling and lays out which technique should be used for detecting specific anomalies.

This paper also researches the various Anomaly Detection Methods and suggests the Principal Component Analysis (PCA) Anomaly Detection Method that it finds is more effective in Anomaly Detection than other methods.

The paper explores the usage of Anomaly Detection Methods in combination with sampling techniques in order to increase Anomaly Detection effectiveness and decrease the resource overhead required to implement Anomaly Detection Solutions. The paper suggests first using Selective Flow Sampling and Selective Smart Sampling and then using Principal Component Analysis (PCA) Method on the sampled data, to detect anomalies. This improves the accuracy of anomaly detection and at the same time this approach is not resource intensive. This paper also researches which Flow Sampling technique (Selective Sampling or Smart Sampling) should be used for particular types of anomalies.

3. Related research works

In this section we present two different anomaly detection techniques that represent most commonly used anomaly detection strategies. Firstly, we present a Sequential Non-Parametric Change-Point Detection Method and then Entropy-based Algorithm for Anomaly Detection.

3.1 Sequential Non-Parametric Change-Point Detection Method [10]

The objective of Change-Point Detection (CPD) is to determine if the observed time series is statistically homogeneous and, if not, to find the point in time when the change happens. The attack detection algorithm that is described below belongs to the sequential category of Change Point Detection in which tests are done online with the data presented sequentially and the decisions are made on-the-fly.

Since non-parametric methods are not model-specific, they are more suitable for analyzing systems like the Internet which is dynamic and complex. The nonparametric CUSUM (Cumulative Sum) method is applied for the detection of attacks. The main idea of the non-parametric CUSUM algorithm is that the mean value of a random sequence $\{X_n\}$ is negative during normal operation and becomes positive when a change occurs.

Thus, we can consider $\{X_n\}$ as a stationary random process which under the normal conditions, the mean of $\{X_n\}$, $E(X_n)=c$. A parameter α is chosen to be an upper bound of c, i.e., $\alpha > c$, and another random process $\{Z_n\}$ is defined so that $Z_n = X_n - \alpha$, which has a negative mean during normal operation. The purpose of introducing α is to offset the possible positive mean in $\{X_n\}$ caused by small network anomalies so that the test statistic Y_n, which is described below, will be reset to zero frequently and will not accumulate with time.

When an attack takes place, Z_n will suddenly increase and become a large positive number. Suppose, during an attack, the increase in the mean

of Z_n, can be lower-bounded by h. The change detection is based on the observation of h>>c.

More specifically, let

$$y_n = (y_{n-1} + Z_n)^+$$

$$y_0 = 0.$$

Where x+ is equal to x if x > 0 and 0 otherwise. The decision function can be described as follows:

$$d_N(y_n) = 0, \text{ if } y_n < N,$$
$$d_N(y_n) = 1, \text{ if } y_n > N$$

Where $d_N(y_n)$ is the decision at time n: '0' stands for normal operation and '1' for attack (a change occurs), while N represents the attack threshold. This anomaly detection technique has been used with different type of metrics, like the SYN/FIN packet ratio or the percentage of new source IP addresses in a time in order to detect Denial of Service attacks.

3.2 Entropy-based Algorithm for Anomaly Detection [2]

An entropy-based anomaly detection method identifies network anomalies by examining some characteristic traffic feature distributions, and represents a wide class of commonly used anomaly detection strategies. This method is independent of network topology and traffic characteristics, and can be applied to monitor every type of network.

The entropy H(X) of a data set X = {x1, x2, xn} is defined as,

$$H(X) = -\sum_{i=1}^{N} p_i \log_2(p_i), \quad (1)$$

Where N is the number of elements contained in data set X, pi is the probability P[X = xi].

Entropy measures the randomness of a data set. High entropy values signify a more dispersed probability distribution, while low entropy values denote concentration of a distribution. Entropy values, as defined in Eq. 1, range between 0 and log2N. In order to have a metric independent of the number of distinct values of the data set, we normalize the entropy by dividing H(X) with the maximum entropy value log2N. The normalized entropy is given by the following while its values range in (0, 1).

$$H_n(X) = -\frac{\sum_{i=1}^{N} p_i \log_2(p_i)}{\log_2 N} \quad (2)$$

Entropy has been extensively used for anomaly detection purposes. Some common traffic feature distributions that are valuable in network anomaly detection are:

- The source IP address (srcIP)
- The destination IP address (dstIP)
- The source port (srcPort)
- The destination port (dstPort)
- The flow size (flow-size)

For example, an anomaly such as an infected host that tries to infect other hosts in the Internet (worm propagation) results in decrease of the entropy of the source IP addresses. The infected machine produces a large number of flows, causing the same source IP address to dominate in the flow distribution of source IP addresses. On the other hand, during a port scanning activity, the entropy of the destination port increases due to the scan of random destination ports. Based on these alterations, the network operator can identify the presence of an anomaly using predefined thresholds on the changes in the corresponding entropy values.

4. Solutions/Analysis

4.1 Network Traffic Anomalies

In this article we focus on three well-known malicious anomalies that could be characterized as network attacks — distributed denial of service (DDoS), worm propagation, and port scan — and two other common anomalies that are caused by legitimate network usage: flash crowd and alpha flow.

DDoS attack: A DDoS attack is characterized by an explicit attempt to prevent the legitimate use of a service. In general, DoS attacks exploit known vulnerabilities of a communication protocol in order to disable the victim's ability to service requests. One frequent type of DDoS attack is SYN flooding, where malicious sources send a large number of TCP SYN packets to the victim's service, thus making the target machine unable to handle all these requests. Other types of DDoS attacks are UDP and ICMP flooding attacks where a very high number of UDP or ICMP packets are sent toward the victim's network from multiple sources.

Worm propagation: The term worm defines a malicious self-replicating program that tries to infect other machines by exploiting a specific vulnerability. During the propagation phase, the infected machine

sends a small number of packets per target to a large number of machines on the Internet.

Port scan activity: Port scan activity includes traffic caused by a single machine that sends probe packets to a wide range of ports toward a specific host to check which services are available.

Flash crowd: A flash crowd event consists of a large legitimate demand for a specific service (i.e., many clients simultaneously downloading the new release of a Linux distribution or a security patch from an HTTP/FTP server). This type of event results in the increase of both inbound (requests) and outbound traffic (responses) from the HTTP/FTP server.

Alpha flows: Alpha flows compose a network anomaly in which traffic increases from just a few high-volume connections between two hosts. Alpha traffic is usually caused by large file transmissions over high-bandwidth links or network experiments between different domains.

4.2 Intelligent Sampling

We now discuss the impact of sampling on anomaly detection. Anomaly detection is based on the concept that perturbations of normal behavior suggest the presence of anomalies and/or attacks. Network anomaly detection techniques rely on the analysis of network traffic and the characterization of the dynamic statistical properties of traffic normality in order to accurately and quickly detect network anomalies.

Based on the observation that for specific-purpose applications such as anomaly detection, a large fraction of information is contained in a small fraction of flows, by using intelligent sampling techniques, we achieve "magnification" of the appearance of anomalies within the sampled data set by preferentially selecting the appropriate data. This way we achieve further improvement of the detection effectiveness and in some cases reveal anomalies that would have been invisible in the unsampled case.

Contrary to common practice and belief, where sampling is considered an inherently lossy process that negatively impacts the fidelity of the sampled stream with reference to most network management and monitoring processes, we argue that sampling not only does not harm the anomaly detection process, but in several cases facilitates and improves its effectiveness. Therefore, a whole new class of sampling schemes, opportunistic sampling/intelligent sampling, may emerge, which aim at turning the inherent in principle drawback of information loss in sampling to a significant beneficial feature for anomaly detection.

Sampling techniques can be divided in two major categories: ***packet-based and flow-based sampling***.

Packet-based sampling

In packet-based sampling packets are selected using a deterministic or nondeterministic method. After observing raw packets, it extracts the signature of the offending packets. Then it specifically blocks only the offending traffic, leaving the legitimate traffic untouched. Often, large-scale attack tools initialize packet headers or content with certain, nonrandom data. For example, the TCP window size or sequence number, which is advertised in a TCP-SYN packet, could be fixed. Packet-based anomaly detection observes raw packets as they traverse the network links.

Observation of network traffic can be done in several ways. One is to configure a spanning port. A router or switch then makes a copy of every packet that is sent/ received on one or more of its interface ports, and sends this copy out on the span port. Another method is the use of network taps. Those are passive devices, which allow the fully transparent observation of packets on a network link. The advantage of network taps is that they can be used even when no network device is available to provide traffic via spanning ports.

Advantages

- Lots of detailed information- Precise timing information, information in packet headers

Disadvantages

- Overhead
    - Hard to keep up with high-speed links
    - Often requires a separate monitoring device

Flow-based sampling [2]

In flow-based sampling packets are first classified into flows. A flow is defined as a set of packets that have in common the following packet header fields: source IP address, source port, destination IP address, destination port, and protocol. In this case sampling is performed in flows, which results in the selection of all packets that make up a particular flow.

Application of packet sampling on network traffic measurements has been extensively studied in the literature, mainly for traffic analysis, planning, and management purposes. Researchers have proposed schemes that follow an adaptive packet sampling approach in order to achieve more accurate measurements of network traffic.

Recent research results demonstrated that flow sampling *improves estimation accuracy of flow statistics*. This fact makes flow sampling more suitable for anomaly detection purposes. In the following we describe two well-known preferential flow-based sampling techniques. The first, referred to as ***selective sampling***, targets small flows (in terms of number of packets), while the second, referred to as ***smart sampling***, selects large flows. Both present an opportunistic character in their operation, as they aim to exploit the fact that, with reference to the occurrence of anomalies, a large fraction of information is contained within a small fraction of flows. Therefore, anomalies usually indicated by the occurrence of outliers can be more easily revealed within an appropriately selected data set such as the one that may result from intelligent sampling.

It has been demonstrated that small flows are usually the source of many network attacks (e.g., DDoS, port scans, worm propagation); therefore, they should be preferentially selected in order to achieve high anomaly detection effectiveness. Selective sampling follows this paradigm, and the selection of an individual flow is based on the following expression:

$$p(x)=\begin{cases} c & x \le z \\ \dfrac{z}{n \cdot x} & x > z \end{cases}, \qquad (1)$$

Where x is the flow size in packets, 0 < c £ 1, n ³ 1 and z is a threshold (measured in packets). As we can observe from expression, flows that are smaller than z are sampled with a constant probability c, while flows that are larger in size than z are sampled with probability inversely proportional to their size. With the appropriate value for parameter c a significant proportion of small flows can be selected without decreasing anomaly detection effectiveness. The selection of large flows can be further reduced by increasing the value of parameter n.

On the other hand, smart sampling is a type of flow based sampling that focuses on the selection of large flows. More specifically, in smart sampling a flow of size x is selected with probability p(x) according to the following expression:

$$p(x)=\begin{cases} x/z & x < z \\ 1 & x \ge z \end{cases}, \qquad (2)$$

Where x is the flow size in bytes and z is a threshold. In our study we consider x as the flow size in packets. As we can observe from Eq. 2, flows that are larger in size than z are sampled with probability 1, while flows that are smaller than z are sampled with probability proportional to their size. This sampling scheme is suitable for detecting anomalies that are caused by large flows, like flash crowd events and alpha flows.

Advantages

- Accurate Anomaly Detection

Disadvantages

- Resource Intensive

4.2.1 Intelligent Sampling Method Proposed

We choose to use the flow-based sampling in our solution. We choose to use both the Selective Sampling and Smart Sampling flow-based sampling techniques depending on the type of anomaly to be detected. The anomaly detection effectiveness for a particular type of anomaly can be increased by selecting the appropriate flow sampling technique. The table below shows the type of flow-based sampling technique to be used for the five types of network anomalies mentioned in the section “Network Traffic Anomalies”-

Anomaly	**Description**	**Flow Sampling Type**

Distributed denial of service (DDoS) attack	An attack on a specific service, making the resource unavailable to its users	Selective Sampling
Worm propagation	A self-replicating program that tries to infect other machines by exploiting a specific vulnerability	Selective Sampling
Port scan	A self-replicating program that tries to infect other machines by exploiting a specific vulnerability	Selective Sampling
Flash crowd	A large demand for a specific service (i.e., many clients downloading a specific file from an HTTP/FTP server)	Smart Sampling
Alpha flows	A small number of flows that have a very large quantity of packets (data transferred between two specific hosts)	Smart Sampling

Table 1: Flow Sampling Type suitable for different types of Anomalies

Thus network anomaly detection effectiveness can be improved and anomaly classification can be done using opportunistic flow sampling. For specific-purpose applications such as anomaly detection, a large fraction of information is contained in a small fraction of flows. Therefore, observing the network traffic characteristics of various classes of anomalies, we can select the appropriate sampling method to preferentially sample the traffic data in order to enhance anomaly detection effectiveness. Even with small rates of anomalous traffic, intelligent sampling techniques significantly improve anomaly detection effectiveness and in several cases reveal anomalies that would otherwise be untraceable.

We suggest the Principal Component Analysis (PCA) method in combination with opportunistic flow-based sampling for the effective detection of network anomalies. The PCA method is described in section "Network Anomaly Detection Method".

4.2.2 Flow-based Sampling Implementation Issues and Challenges [2]

The application of flow-based sampling poses several challenging implementation issues. More specifically, flow sampling needs to make a decision on whether or not to sample a flow record that has already been collected and stored in memory at the end of a time-moving window. Flow sampling can be implemented using a hash table with a five-tuple object identifying a flow (i.e., source and destination IP addresses, source and destination ports, and protocol) as the key, and a value for the counter of packets belonging to the specific flow. A new hash table is created periodically, and the flows are sampled at the end of the time window, replacing the previous hash table. For each arriving packet the hash table is traversed, increasing the counter of the corresponding flow if it already exists, or creating a new entry in case of a new flow within the time window.

Such a procedure can be implemented with the use of a network processor card, which can be deployed either within enhanced future routers or at specific network measurement points. Modern network processor cards are able to conduct passive monitoring at speeds starting from 1 Gb/s and possibly up to 10 Gb/s. Every entry in the hash table requires 8 bytes for both the source and destination IP addresses 4 bytes for both the source and destination ports, 1 byte for the protocol, and 3 bytes for the counter, resulting in 16 bytes per entry. In a 1 Gb/s link, there are about 100,000 packets/s; thus, in the worst case scenario where every packet belongs to a different flow, we would need $16 \times 100{,}000 = 1{,}600{,}000$ bytes (less than 1.6 Mbytes of memory). In the case where a time window of 10 s is utilized, approximately 16 Mbytes of memory would be required.

4.3 Network Anomaly Detection Method-Principal Component Analysis (PCA) [6]

Principal Component Analysis is used in a variety of domains to reduce the number of dimensions in input sets without losing the "information" contained in them. PCA is particularly helpful in anomaly detection since it can reduce the data dimensionality into a smaller set of independent variables.

At its core, PCA produces a set of principal components, which are orthonormal Eigen value/eigenvector pairs. In other words, it projects a new set of axes which best suit the data. In our implementation, these set of axes represent the normal connection data. Anomaly/Outlier detection occurs by mapping live network data onto these 'normal' axes and calculating the distance from the axes. If the distance is greater than a certain threshold, then the connection is classified as an attack. This section introduces PCA and describes how it is used in anomaly detection.

4.3.1 PCA Methodology

Anomaly detection systems typically require more data than is available at the packet level. Using preprocessing and feature extraction methods, the data available for anomaly detection is high dimensional in nature. The computational cost of processing massive amounts of data in real time is immense. Therefore, applying Principal Component Analysis as a data reduction tool while retaining the important properties of the data is useful. PCA works to explain the variance-covariance structure of a set of variables through a new set of orthonormal projection values which are linear combinations of the original variables. Principal components are particular linear combinations of p random variables X1, X2, ... , Xp. These variables have three important properties:

1. X1, X2, ..., Xp are uncorrelated,
2. X1, X2, ... , Xp are sorted in descending order, and
3. $X_{total} = \sum_{i=0}^{p} X_i$, the total variance is equal to the sum of the individual variances.

These variables are found from Eigen analysis of the covariance or correlation matrix of the original variables Xo1, Xo2, ... , Xop.

Let the original data, in this case the training data, X be an n x p data matrix of n observations with each observation composed of p fields (or dimensions) X1, X2,...,Xp.

Let R be a p x p correlation matrix of X1, X2,...,Xp. If (λ1, e1), (λ2, e2), ..., (λp, ep) are the p Eigen value/eigenvector pairs of the correlation matrix R, then the ith principal component is

$$\begin{aligned} y_i &= \mathbf{e}_i'(\mathbf{x} - \overline{\mathbf{x}}) \\ &= e_{i1}(x_1 - \overline{x}_1) + e_{i2}(x_2 - \overline{x}_2) \\ &\quad + \ldots + e_{ip}(x_p - \overline{x}_p), i = 1, 2, \ldots, p \end{aligned}$$

Where

λ1 ≥ λ2 ≥ ... ≥ λp ≥ 0,

ei' = ei1, ei2,...,eip is the ith eigenvector,

x = (x1, x2, ..., xp) is the observed data along the variables X1, X2,...,Xp,

$\overline{\mathbf{x}} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_p)$ is the sample mean vector of the observation data.

The principal components derived from the covariance matrix are usually different from the principal components generated from the correlation matrix. When some values are much larger than others, then their corresponding Eigen values have larger weights.

4.3.2 Distance Calculation

Calculating distance from a point is a fundamental operation in anomaly detection techniques. Methods include nearest-neighbor, kth nearest neighbor, Local Outlier Factor, etc. In general, the distance metric used is Euclidean distance. This is the primary calculation in the nearest neighbor approach. Let x = (x1, x2, ... , xp) and y = (y1, y2, ... , yp) be two p-dimensional observations. The Euclidean distance is defined as:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})'(\mathbf{x} - \mathbf{y})} \qquad (1)$$

In Equation 1, each feature carries the same weight in calculating the Euclidean distance. However, when features have a varied weight distribution or are measured on different scales, then the Euclidean distance is no longer adequate. The distance metric needs to be modified to reflect the distribution and importance of each field in the data. One of these metrics is known as the Mahalanobis distance

$$d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})'\mathbf{S}^{-1}(\mathbf{x} - \mathbf{y}) \qquad (2)$$

Where $\mathbf{S}^{-1}$ is the sample covariance matrix. In our work, we replaced $\mathbf{S}^{-1}$ with the correlation matrix, $\mathbf{R}^{-1}$, since many fields in the training set were measured on different scales and ranges. Using the correlation matrix more effectively represents the relationships between the data fields.

4.3.3 Applying PCA to Anomaly/Outlier Detection

In applying PCA, there are two main issues: how to interpret the set of principal components, and how to calculate the notion of distance.
First, each Eigen value of a principal component corresponds to the relative amount of variation it encompasses. The larger the Eigen value, the more significant its corresponding projected eigenvector. Therefore, the principal components are sorted from most to least significant. If a new data item is projected along the upper set of the significant principal components, it is likely that the data item can be classified without projecting along all the principal components.
Secondly, eigenvectors of the principal components represent axes which best suit a data sample. If the data sample is the training set of normal network connections, then those axes are considered normal. Points which lie at a far distance from these axes would exhibit abnormal behavior. Using a threshold value (t), any network connection with Mahalanobis distance greater than the threshold is considered an outlier, and hence in our case an attack.
Consider the sample principal components, y1, y2, ... , yp of an observation x where:

$$y_i = \mathbf{e}_i'(\mathbf{x} - \overline{\mathbf{x}}), i = 1, 2, ..., p$$

The sum of squares of the partial principal component scores is equal to the principal component score:

$$\sum_{i=1}^{p} \frac{y_i^2}{\lambda_i} = \frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} + ... + \frac{y_p^2}{\lambda_p} \qquad (3)$$

Equates to the Mahalanobis distance of the observation X from the mean of the normal sample data set.

4.3.4 PCA Framework

All anomaly detections require an offline training or learning phase whether those methods are outlier detection, statistical models, or association rule mining. Many times, the mechanisms applied in the online and offline phases are tightly coupled. Principal component analysis, however, clearly separates the offline and online detection phases. This property is an advantage for hardware implementation. Figure 1 outlines the steps involved in PCA.

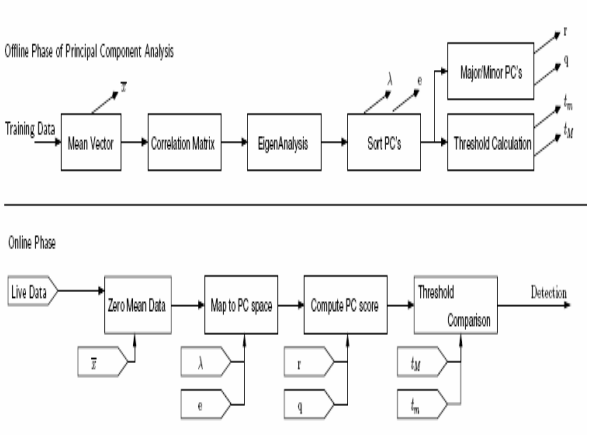

Figure 1: PCA for Network Intrusion Detection

In the offline phase, labeled training data is taken as input and a mean vector of the whole sample is computed. Ideally these data sets are a snapshot of activity in a real network environment.

Secondly, a correlation matrix is computed from the training data. A correlation matrix normalizes all the data by calculating the standard deviation.

Next, Eigen analysis is performed on the correlation matrix to extract independent orthonormal Eigen value/eigenvector pairs. These pairs make up the set of principal components used in online analysis.

Lastly, the sets of principal components are sorted by Eigen value in descending order. The Eigen value is a relative measure of the variance of its corresponding eigenvectors. Using PCA to extract the most significant principal components is what makes it a dimensionality reducing method because only a subset of the most important principal components are needed to classify any new data.

To increase the detection rate of PCA, we use a modified version of PCA. In addition to using the most significant principal components (q) to find

intrusions, it is helpful to look for intrusions along a number of least significant components (r) as well. The most significant principal components are part of the major principal component score (MajC) and the least significant components belong to calculating a minor principal component score (MinC). MajC is used to detect extreme deviations with large values on the original features. These observations follow the correlation structure of the sample data.
However, some attacks may not follow the same correlation model. MinC is used to detect those attacks. As a result, two thresholds are needed to detect attacks. If the principal components are sorted in descending order, then q is a subset of the highest values and r is a subset of the smallest components. The MajC threshold is denoted tM while theMinC threshold is referred to as tm. An observation x is an attack if:

$$\sum_{i=1}^{q} \frac{y_i^2}{\lambda_i} > t_M \text{ or } \sum_{i=p-r+1}^{p} \frac{y_i^2}{\lambda_i} > t_m \qquad (4)$$

The online portion takes q major principal components and r minor principal components and maps online data into the Eigen space of those principal components. There are two parallel pipelines, one for calculating the major component variability score (MajC) and one for the minor (MinC). The simulations show that adding the MinC pipeline increases the detection ability and decreases the false alarm rate of using PCA for anomaly detection.

For hardware design, the most computationally expensive portion of PCA is performing eigenvector calculations and sorting. The process of calculating eigenvectors is sequential and difficult to parallelize. Fortunately, this task is part of the offline phase.

We are primarily concerned with accelerating online intrusion detection using PCA. For this segment, the most important bottleneck is computing the PC score. This can be overcome by using hardware parallelism and extensive pipelining in the implementation.

5. Summary

In this article the problem of improving network anomaly detection effectiveness through the application of intelligent flow sampling is researched. We concluded that Flow Sampling more accurately detects Network Anomalies than Packet Sampling. We also concluded that for effective detection of different types of anomalies, different types of Flow Sampling (Selective Sampling and Smart Sampling) techniques should be used. We found that Selective Sampling is more suited for DDOS, Worm propagation and port scan anomalies while Smart Sampling is more suited for Flash crowd and Alpha flows anomalies.

The key motivation and principle of the approach presented in the paper is the exploitation of the fact that for specific-purpose applications such as anomaly detection, a large fraction of information is contained in a small fraction of flows. Hence, by using sampling techniques that opportunistically/intelligently and preferentially sample traffic data, we can achieve "magnification" of the appearance of anomalies within the sampled data set and therefore improve their detection. By observing the network traffic characteristics of various classes of anomalies, we can select the appropriate sampling method to preferentially sample the traffic data in order to enhance anomaly detection effectiveness. Even with small rates of anomalous traffic, intelligent sampling techniques significantly improve anomaly detection effectiveness and in several cases reveal anomalies that would otherwise be untraceable.

We also researched various Anomaly Detection Methods including Sequential Non-Parametric Change-Point Detection Method, Entropy-based Algorithm for Anomaly Detection and Principal Component Analysis (PCA) based Method. We found the PCA method is more accurate and effective in detecting network anomalies than the other methods. PCA reduces [6] the data dimensionality into a smaller set of independent variables. At its core, PCA produces a set of principal components, which are orthonormal Eigen value/eigenvector pairs. In other words, it projects a new set of axes which best suit the data. These set of axes represent the normal connection data. Anomaly/Outlier detection occurs by mapping live network data onto these 'normal' axes and calculating the distance from the axes. If the distance is greater than a certain threshold, then the connection is classified as an attack.

We concluded that PCA Method of Anomaly Detection along with Flow Sampling can prove to be a powerful combination for detecting Network Anomalies.

6. Future works

Future research in this area can focus on doing a practical implementation of Flow-based sampling along with PCA Anomaly Detection Method.

Also, it can focus on further reducing the amount of sampled data using the principle of two-stage sampling, where in the first stage sampling is performed at the flow level, and in the second stage packets are sampled from the already selected flows. Utilizing intelligent sampling at the first stage, the final output would be a significantly reduced data set containing a great part of the anomalous traffic.

7. References

[1] Thomas Chen, Zhi Fu, Liwen He and Tim Strayer, "Recent Developments in Network Intrusion Detection"

[2] Georgios Androulidakis, Vassilis Chatzigiannakis, and Symeon Papavassiliou, "Network Anomaly Detection and Classification via Opportunistic Sampling", National Technical University of Athens, Feb 2009

[3] G. Androulidakis and S. Papavassiliou, "Improving network anomaly detection via selective flow-based sampling", Network Management and Optimal Design Laboratory (NETMODE), School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Iroon Polytexneiou 9, Zografou 15780, Athens, Greece

[4] Jianning Mai, ChenNee Chuah, Ashwin Sridharan, Tao Ye and Hui Zang, "Is Sampled Data Sufficient for Anomaly Detection?"

[5] Esphion, Network Disaster protection, "Packet vs flow-based anomaly detection", A White Paper.

[6] Abhishek Das, Sanchit Misra, Sumeet Joshi, Joseph Zambreno, Gokhan Memik and Alok Choudhary , "An Efficient FPGA Implementation of Principle Component Analysis based Network Intrusion Detection System"

[7] Georgios Androulidakis, Vasilis Chatzigiannakis, Symeon Papavassiliou, Mary Grammatikou and Vasilis Maglaris, "Understanding and Evaluating the impact of Sampling on Anomaly Detection Techniques", Network Management & Optimal Design Lab (NETMODE), School of Electrical and Computer Engineering, National Technical University of Athens, Zografou, Athens, Greece

[8] Peng Ning, Sushil Jajodia, "Intrusion Detection Techniques"

[9] Wei Wang and Roberto Battiti, "Identifying Intrusions in Computer Networks based on Principal Component Analysis", Department of Computer Science and Telecommunications, Via Sommarive, 14 — I-38050 Trento—Italy, December 7, 2005

[10] Androulidakis G., Papavassiliou S., "Intelligent Flow-Based Sampling for Effective Network Anomaly Detection", Global Telecommunications Conference, 2007

[11] Nick Duffield, Patrick Haffner, Balachander Krishnamurthy, Haakon Ringberg, "Rule-Based Anomaly Detection on IP Flows", AT&T Labs—Research, Florham Park, NJ 07932, Dept. of Comp. Science, Princeton University, Princeton, NJ 08544

[12] Daniela Brauckhoff, Bernhard Tellenbach, Arno Wagner, Anukool Lakhina, Martin May and ETH Zurich, "Impact of Packet Sampling on Anomaly Detection Metrics", Boston University

[13] Petar Cisar and Sanja Maravic Cisar, "A Flow-based Algorithm for Statistical Anomaly Detection"

[14] Yan Gao, Zhichun Li and Yan Chen, "A DoS Resilient Flow-level Intrusion Detection Approach for High-speed Networks", Department of EECS, Northwestern University

[15] V. Chatzigiannakis, G. Androulidakis, K. Pelechrinis, S. Papavassiliou and V. Maglaris, "Data fusion algorithms for network anomaly detection: classification and evaluation", Network Management & Optimal Design Laboratory (NETMODE), School of Electrical & Computer Engineering, National Technical University of Athens (NTUA)

[16] Ling Huang, XuanLong Nguyen, Minos Garofalakis, Michael I. Jordan, Anthony Joseph, and Nina Taft, "In-Network PCA and Anomaly Detection"