Anomaly Based Intrusion Detection
Using Incremental Approach: A Survey

M H Bhuyan¹, D K Bhattacharyya¹ and J K Kalita²

¹ Department of Computer Science & Engineering
Tezpur University
Napaam, Tezpur, Assam, India.
{mhb,dkb}@tezu.ernet.in
² Department of Computer Science
University of Colorado at Colorado Springs
CO 80933-7150, USA
jkalita@uccs.edu

Abstract. As the communication industry has connected distant corners of the globe using advances in network technology, intruders or attackers have also increased attacks on networking infrastructure commensurately. System administrators can attempt to prevent such attacks using intrusion detection tools and systems. There are many commercially available signature-based Intrusion Detection Systems (IDSs). However, most IDSs lack the capability to detect novel or previously unknown attacks. A special type of IDSs, called Anomaly Detection Systems, develop models based on normal system or network behavior, with the goal of detecting both known and unknown attacks. Anomaly detection systems face many problems including high rate of false alarm, ability to work in online mode, and scalability. This paper presents a survey of incremental approaches for detecting anomaly in normal system or network traffic. The technological trends, open problems, and challenges over anomaly detection using incremental approach are also discussed.

1 Introduction

The Internet connects users and providers of information and media services from distant corners of the world. Due to widespread availability of advanced networking technologies, the threat from spammers, intruders or attackers, and criminal enterprises is also increasing. Intrusion Detection Systems (IDSs) and firewall technologies can prevent some of these threats. One study [1] estimates the number of intrusion attempts over the entire Internet to be in the order of 25B per day and increasing. McHugh [2] claims that attacks are becoming more sophisticated while they get more automated, and thus the skills needed to launch them are being reduced.

There are two types of IDSs: signature-based and anomaly-based. Signature-based IDSs exploit signatures of known attacks. Such systems require frequent updates of signatures for known attacks and cannot detect unknown attacks or anomalies for which signatures are not stored in the database. In contrast, anomaly based IDSs are extremely effective in finding and preventing known as well as unknown or zero day attacks [3]. However, an anomaly detection system has many shortcomings such as high rate of false alarm, and the failure to scale up to gigabit speeds.

An incremental approach updates normal profiles dynamically based on changes in network traffic without fresh training using all data for attack detection. Based on existing profiles, it can take the decision to raise an attack if abrupt changes happen in network traffic. Thus, it is useful to detect anomalies from normal system or network traffic from existing signatures or profiles, using an incremental approach. That is, an incremental approach updates profiles or signatures dynamically incorporating new
profiles as it encounters them. It does not need to load the whole database each time into the memory and learn fresh from the beginning. In the last two decades, a good number of anomaly-based intrusion detection approaches [3–5] have been developed, but a lot of them are general in nature, and thus quite simple. Due to the lack of papers that discuss various facets of incremental anomaly detection, we present a survey in a structured manner along with current research challenges and issues in this field.

This paper gives a short survey of anomaly detection using incremental approaches. In Section 2, we introduce the basic idea of anomaly detection, types of anomalies, and various aspects of anomaly detection. A good number of incremental techniques and validity measures for anomaly detection are discussed in Section 3. Section 4 is dedicated to evaluation datasets whereas Section 5 presents performance evaluation mechanisms. Section 6 discusses research issues and challenges. Section 7 has our concluding remarks.

2 Anomaly Detection and Its Types

Anomaly detection refers to the important problem of finding nonconforming patterns or behavior in live traffic data. These non-conforming patterns are often known as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, peculiarities or contaminants in different application domains. In contrast, noise consists of non-interesting patterns that hinder traffic data analysis.

2.1 Preface to Anomaly Detection

The central premise of anomaly detection is that intrusive activity is a subset of anomalous activity [3]. When there is an intruder who has no idea of the legitimate user’s activity patterns, the probability that the intruder’s activity is detected as anomalous should be high. Kumar and Stafford [4] suggest four possibilities in such a situation, each with a non-zero probability.

- **Intrusive but not anomalous**: An IDS may fail to detect this type of activity since the activity is not anomalous. But, if the IDS detects such an activity, it may report it as a false negative because it falsely reports the absence of an intrusion when there is one.

- **Not intrusive but anomalous**: If the activity is not intrusive, but it is anomalous, an IDS may report it as intrusive. These are called false positives because an intrusion detection system falsely reports intrusions.

- **Not intrusive and not anomalous**: These are true negatives; the activity is not intrusive and should not be reported as intrusive.

- **Intrusive and anomalous**: These are true positives; the activity is intrusive and much be reported as such.

2.2 Anomaly detection

Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior. In practice, it is very difficult to precisely detect anomalies in network traffic or normal data. A generic architecture of an incremental anomaly based network intrusion detection system (ANIDS) is shown in figure 1.

2.3 Aspects of Anomaly Detection

Anomaly detection is usually flexible and adaptive enough to detect novel attacks. However, there are still many problems in anomaly detection approaches. Some issues in anomaly detection are discussed below.
The purpose of anomaly detection is to find anomalies in various application domains. An anomalous MRI image may indicate presence of malignant tumors [6]. Anomalies in credit card transaction data may indicate credit card or identity theft [7]. Anomalous readings from a spacecraft sensor can signify a fault in some component of the spacecraft [8]. Detection of anomalies or outliers were discussed as early as in the 9th century [9]. Anomalies are classified into three categories.

- **Point anomalies**: An individual data instance can be defined as anomalous with respect to the rest of the data.
- **Contextual anomalies**: A data instance may be anomalous in a specific context with respect to a condition [10]. Two sets of attributes are normally used to detect this type of anomalies: (a) contextual attributes and (b) behavioral attributes. Contextual attributes determine the context or neighborhood of an instance. Behavioral attributes are responsible for the non-contextual characteristics of an instance.
- **Collective anomalies**: Sometimes, it is not a single data point that is anomalous, but it is a collection of related data instances that is anomalous with respect to the entire dataset.

It is worth mentioning that unlike point anomalies, detection of collective anomalies requires identifying the individual but related anomalous points in a dataset. To detect contextual anomalies, one needs to identify the appropriate contextual or behavioral attributes in the data.

- **Data Labels**: Labeling a data instance as normal or anomalous based on its behavior is often prohibitively expensive. A human expert often does labeling manually and hence it requires substantial effort to obtain the labeled training dataset. Typically, getting a labeled set of anomalous data instances covering all possible types of anomalous behavior is more difficult than getting labeled data for normal behavior.

- **Reporting of anomalies**: An important aspect for any anomaly detection technique is the way it reports anomalies. Typically, the outputs produced by anomaly detection techniques are one of the following two types.
  - **Scores**: An anomaly score is assigned to each instance in the test data depending on the degree to which that instance is considered an anomaly. Usually, the output of such a technique is a ranked list of anomalies. An analyst may choose to either analyze the top few anomalies or use a threshold to select anomalies for further investigation.
Labels: A label (normal or anomalous) is assigned to each test instance. Scoring based anomaly detection techniques allow the analyst to use a domain specific threshold to select the most relevant anomalies. Generally, an IDS uses clustering techniques for grouping activities. It merely classifies the data, without any interpretation of the data. A classical labelling strategy is to measure the cardinality of the clusters, and label some percentage of the smallest clusters as malicious. This approach does, however, have some limitations, and does not detect massive attacks [11] properly, e.g. Denial-of-Service attacks. Dunn's index [12] or C – index [13] is well suited clustering quality evaluation indexes in labelling clusters to detect attacks. A hallmark of good clustering quality is compact clusters distant from each other. Dunn's index \( D \) is defined as the ratio between the minimal intra-cluster distances \( d_{\text{min}} \) to maximal inter-cluster distance \( d_{\text{max}} \), i.e., \( d_{\text{min}}/d_{\text{max}} \). \( D \) is limited to the interval \([0, 1]\) and should be maximized for good clustering. On the other hand, \( C – \text{index} \) \( C \) is defined in terms \( S \), i.e., the sum of distances over all pairs of objects in the same cluster. Let \( n \) be the number of such object pairs, and \( S_{\text{min}} \) be the sum of the \( n \) smallest distances if all pairs of objects are considered. Likewise, let \( S_{\text{max}} \) be the sum of the \( n \) largest distances out of all pairs. Then \( C \) is computed as \( (S – S_{\text{min}})/(S – S_{\text{max}}) \). \( C \) is limited to the interval \([0, 1]\) and should be minimized for good clustering. \( D \) measures only two distances whereas \( C \) requires clusters of equal cardinality to produce proper quality evaluations.

3 Existing Approaches to Incremental Anomaly Detection

Recently, several papers have focused on modeling and analyzing anomaly based IDSs. Only a few of these present incremental approaches. Based on an exhaustive survey of published incremental anomaly detection approaches, we conclude that most approaches have high rate of false alarm, are non-scalable, and are not fit for deployment in high-speed networks. Anomaly detection techniques can be of three types: unsupervised, semi-supervised and supervised. Some published papers in each of these three categories are discussed in the following in brief.

3.1 Unsupervised Approaches

These detection approaches do not require training data, and thus are most widely applicable. These techniques make the implicit assumption that normal instances are far more frequent than anomalies in the test data. If this assumption is not true, such techniques suffer from high false alarm. Most existing unsupervised anomaly detection approaches are clustering based. Clustering is a technique to group similar objects. It deals with finding structure in a collection of unlabeled data. Representing the data by fewer clusters necessarily loses certain finer details, but achieves simplification. In anomaly detection, clustering plays a vital role in analyzing the data by identifying various groups as either belonging to normal or to anomalous categories. There are many different clustering based anomaly detection approaches in the literature. Most commonly used clustering techniques are: partitioning-based (e.g., Zhong et al. [14]), hierarchical (e.g., Hsu et al. [15], Burbeck et al. [16], Kalle et al. [17]), density-based (e.g., Ren et al. [18]), and grid-based techniques. These techniques are discussed in the following in brief.

(a) Hsu et al. [15]: Adaptive resonance theory network (ART) is a popular unsupervised neural network approach. Type I adaptive resonance theory network (ART1) deals with binary numerical data, whereas type II adaptive resonance theory network (ART2) deals with general numerical data. Several information systems collect mixed type attributes, which include numeric attributes and categorical attributes. However, both ART1 and ART2 do not deal with mixed type data. If the categorical attributes are transferred to binary data format, the binary data do not reflect reality with the same
fidelity. It ultimately influences clustering quality. The authors present a modified adaptive resonance theory network (M-ART) and a conceptual hierarchy tree to solve the problem of clustering with mixed data. They show that the M-ART algorithm can process mixed type data well and has a positive effect on clustering.

(b) **Zhong et al. [14]:** This paper presents an incremental clustering algorithm for intrusion detection using clonal selection based on a partitioning approach. It partitions the dataset into initial clusters by comparing the distance from data to cluster centroid with the size of cluster radius, and analyzes the clustering data with mixed attributes by using an improved definition of distance measure. The objective function optimizes clustering results by applying a clonal selection algorithm [19], and then labels clusters as normal or anomalous as appropriate. The authors try to find better cluster centroids to improve the partitioning quality which is evaluated by the objective function. If the value of objective function is small, the sum of the distances from data to the cluster centers is also small and the objects in the same cluster are more close to each other. The method attempts to optimize cluster results from one iteration to the next using the clonal selection algorithm [19]. The authors establish this incremental technique in terms of high detection rate and low false positive rate.

### 3.2 Semi-supervised Approaches

Here, the training data instances belong to the normal class only. Data instances are not labeled for the attack class. There are many approaches used to build the model for the class corresponding to normal behavior. This model is used to identify the anomalies in the test data. Some of the detection methods are discussed in the following.

(a) **Burbeck et al. [16]:** ADWICE (Anomaly Detection With fast Incremental Clustering) uses the first phase of the BIRCH clustering framework [20] to implement fast, scalable and adaptive anomaly detection. It extends the original clustering algorithm and applies the resulting detection mechanism for analysis of data from IP networks. The performance is demonstrated on the KDD99 intrusion dataset as well as on data from a test network at a telecom company. Their experiments show good detection quality (95%) and acceptable false positives rate (2.8 %) considering the online, real-time characteristics of the algorithm. The number of alarms is further reduced by application of the aggregation techniques implemented in the Safeguard architecture.

(b) **Rasoulifard et al. [21]:** It is important to increase the detection rate for known intrusions and also to detect unknown intrusions at the same time. It is also important to incrementally learn new unknown intrusions. Most current intrusion detection systems employ either misuse detection or anomaly detection. In order to employ these techniques effectively, Rasoulifard et al. propose an incremental hybrid intrusion detection system. This framework combines incremental misuse detection and incremental anomaly detection. The framework can learn new classes of intrusion that do not exist in data used for training. The framework has low computational complexity, and so it is suitable for real-time or on-line learning. The authors use the KDD98 intrusion dataset to establish this method.

(c) **Kalle et al. [17]:** Anomaly detection is very expensive in real-time. First, to deal with massive data volumes, one needs to have efficient data structures and indexing mechanisms. Second, the dynamic nature of today’s information networks makes the characterization of normal requests and services difficult. What is considered normal during some time interval may be classified as abnormal in a new context, and vice versa. These factors make many proposed data mining techniques less suitable for real-time intrusion detection. Kalle et al. look at the shortcomings of ADWICE and propose a new grid index that improves detection performance while preserving efficiency in search. Moreover, they

---

1 Safeguard: The safeguard project, (online) http://www.safeguardproject.info/
propose two mechanisms for adaptive evolution of the normality model: incremental extension with new elements of normal behavior, and a new feature that enables forgetting of outdated elements of normal behavior. It evaluates the technique for network-based intrusion detection using the KDD99 intrusion dataset as well as on data from a telecom IP test network. The experiments yield good detection quality and act as proof-of-concept for adaptation of normality.

3.3 Supervised Approaches

In this approach, a predictive model is developed based on a training dataset (i.e., data instances labeled as normal or attack class). Any unseen data instance is compared against the model to determine which class it belongs to. There are two major issues that arise in supervised anomaly detection. First, the anomalous instances are far fewer in number compared to normal instances in the training data. Issues that arise due to imbalanced class distributions have been addressed in the data mining and machine learning literature [22]. Second, obtaining accurate and representative labels, especially for the anomaly class is usually challenging. A number of techniques have been proposed that inject artificial anomalies in a normal dataset to obtain a labeled training dataset [23]. Other than these two issues, the supervised anomaly detection problem is similar to building predictive models. Some supervised anomaly detection methods are discussed below.

(a) **Yu et al. [24]**: The authors propose an incremental learning method by cascading a service classifier (SC) over an incremental tree inducer (ITI) for supervised anomaly detection. A service classifier ensures that the ITI method is trained with instances with only one service value (e.g., ftp, smtp, telnet etc.); the ITI method is trained with instances incrementally without the service attribute. The cascading method has three phases: (i) training, (ii) testing and (iii) incremental learning. During training phase, the service classifier method is first applied to partition the training dataset into disjoint clusters according to different services. Then, the ITI method is trained with the instances in each cluster. The service classifier method ensures that each training instance is associated with only one cluster. In the testing phase, it finds the cluster to which the test instances belong. Then, the ITI method is tested with the instances. In the incremental learning phase, the service classifier and ITI cascading method are not re-trained; the authors use incremental learning to train the existing ITI binary tree. Nearest Neighbor combination rules embedded within K-Means+ITI and SOM+ITI cascading methods are used in experiments. The authors also compare the performance of SC+ITI with K-Means, SOM, ITI, K-Means+ITI and SOM+ITI methods in terms of detection rate and false positive rate (FPR) over the KDD’99 dataset. Results show that the ITI method has better performance than K-Means, SOM, K-Means+ITI and SOM+ITI methods in terms of overall detection rate. However, the SC+ITI cascading method outperforms the ITI method in terms of detection rate and FPR and obtains better detection rate compared to other methods. Like the ITI method, SC+ITI also provide additional options for handling missing values and incremental learning.

(b) **Lascov et al. [25]**: The authors focus on the design and analysis of efficient incremental SVM learning, with the aim of providing a fast, numerically stable and robust implementation. A detailed analysis of convergence and algorithmic complexity of incremental SVM learning is carried out in this paper. Based on this analysis, a new design for storage and for numerical operations is proposed. The design speeds up the training of an incremental SVM by a factor of 5 to 20. The performance of the new algorithm is demonstrated in two scenarios: learning with limited resources and active learning. The authors discuss various applications of the algorithm, such as in drug discovery, online monitoring of industrial devices and surveillance of network traffic.

(c) **Ren et al. [18]**: The authors propose a new anomaly detection algorithm that can update the normal profile of system usage dynamically. The features used to model a system’s usage pattern are derived from program behavior. A new program behavior is inserted into old profiles by density-based
incremental clustering when system usage pattern changes. It is much more efficient compared to traditional updating by re-clustering. The authors test their model using the 1998 DARPA BSM audit data, and report that the normal profiles generated by their algorithm are less sensitive to noise data objects than profiles generated by the ADWICE algorithm. The method improves the quality of clusters and lowers the false alarm rate.

3.4 Discussion

Based on this short and selective survey of incremental techniques for anomaly detection, we make the following observations.

- Most incremental anomaly detection techniques have been benchmarked by using KDD99 intrusion dataset; hence, they cannot claim to be truly up-to-date. The KDD99 intrusion datasets have been already established to be biased considering the presence of normal and attack data in unrealistic proportions.
- Performances of these techniques are not great in terms of false positive rates.
- We report in tabular format how the various detection approaches compare against one another in regards to detection methods, and salient features for anomaly detection. Comparison will be handy for researchers working in this field (Refer to Table 1 and Table 2).

<table>
<thead>
<tr>
<th>Detection approaches</th>
<th>Methods</th>
<th>Proximity Measures</th>
<th>Clustering (C/)</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsupervised</td>
<td>M-ART [15]</td>
<td>LCP (least common points)</td>
<td>C</td>
<td>Mixed</td>
</tr>
<tr>
<td></td>
<td>Clonal Selection [14]</td>
<td>Manhattan distance</td>
<td>C</td>
<td>Numeric</td>
</tr>
<tr>
<td>Semi-supervised</td>
<td>ADWICE TRD [16]</td>
<td>Euclidean distance</td>
<td>C</td>
<td>Numeric</td>
</tr>
<tr>
<td></td>
<td>Hybrid IDS [21]</td>
<td>Average distance</td>
<td>H</td>
<td>Numeric</td>
</tr>
<tr>
<td></td>
<td>ADWICE GRD [17]</td>
<td>Euclidean distance</td>
<td>C</td>
<td>Numeric</td>
</tr>
<tr>
<td></td>
<td>SVM [25]</td>
<td>Nearest neighbor</td>
<td>S</td>
<td>Numeric</td>
</tr>
<tr>
<td>Supervised</td>
<td>SVM-based [18]</td>
<td>Nearest neighbor</td>
<td>C</td>
<td>Numeric</td>
</tr>
</tbody>
</table>

3.5 Validity measures

The performance of any ANIDS is highly dependent upon (i) its individual configuration, (ii) the network it is monitoring and (iii) its position within that network. Simply benchmarking an ANIDS once in a certain environment does not provide a definitive method for assessing it in a given situation. A number of metrics are used to assess the suitability of a given ANIDS for a particular situation or environment. Some of these are discussed next.

- Ability to identify attacks: The main performance requirement of an ANIDS is to detect intrusions. In particular, many vendors and researchers appear to consider any attempt to place malicious traffic on the network as an intrusion. In reality a more useful system will log malicious traffic and only inform the operator if the traffic poses a serious threat to the security of the target host.
Table 2: Comparing incremental techniques for anomaly detection

<table>
<thead>
<tr>
<th>Reference</th>
<th>Real (R)/Non-</th>
<th>Features</th>
<th>Dataset Used</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burbeck et al. [16]</td>
<td>R/N</td>
<td>CF (Cluster feature) tree</td>
<td>KDDCup99</td>
<td>Detection Rate (%)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>False Positive Rate (%)</td>
</tr>
<tr>
<td>Hu et al. [15]</td>
<td>N</td>
<td>Distance-based tree</td>
<td>UCI</td>
<td>-</td>
</tr>
<tr>
<td>Burbeck et al. [17]</td>
<td>R/N</td>
<td>CF tree based index</td>
<td>KDDCup99</td>
<td>-</td>
</tr>
<tr>
<td>Yu et al. [24]</td>
<td>N</td>
<td>Cluster based service value</td>
<td>KDDCup99</td>
<td>92.63</td>
</tr>
<tr>
<td>Zhong et al. [14]</td>
<td>N</td>
<td>Distance</td>
<td>KDDCup99</td>
<td>97.2</td>
</tr>
<tr>
<td>Rasoulifard et al. [21]</td>
<td>N</td>
<td>Weighted majority voting</td>
<td>DARPA98</td>
<td>-</td>
</tr>
<tr>
<td>Lössov et al. [25]</td>
<td>R</td>
<td>Objective function</td>
<td>KDDCup99</td>
<td>-</td>
</tr>
<tr>
<td>Ren et al. [18]</td>
<td>N</td>
<td>Distance</td>
<td>DARPA98/BSM</td>
<td>-</td>
</tr>
</tbody>
</table>

- **Known vulnerabilities and attacks**: All ANIDSs should be capable of detecting known vulnerabilities. However, research indicates that many commercial IDSs fail to detect recently discovered attacks. On the other hand, if a vulnerability or attack becomes known, all systems should be patched; otherwise, workarounds should be applied so that the need for an ANIDS to detect these events is obviated.

- **Unknown attacks**: Detecting unknown attacks must be the most important feature of any ANIDS. Only the ability of an ANIDS to detect attacks that are not yet known justifies expenses of its implementation and deployment. New vulnerabilities are being discovered every day and being able to detect known attacks is no longer enough.

- **Stability, Reliability and Security**: An ANIDS should be able to continue operating consistently in all circumstances. The application and the operating system should ideally be capable of running for months, even years without segmentation faults or memory leakage. An important requirement imposed on an ANIDS is the ability to consistently report identical events in the same manner. The system should also be able to withstand attempts to compromise it. The ability of an attacker to identify an ANIDS on a network can be extremely valuable to the attacker. The attacker may then attempt to disable the ANIDS using DoS or DDoS techniques. The ANIDS system should be able to withstand all of these types of attack.

- **Identifying target and source**: An alert raised after detecting an anomaly should also identify the source of the threat and the exact identity of the target system. Additional information from whois or DNS lookup on an IP address should also be obtained, if necessary.

- **Outcome of attack**: Another useful feature of an ANIDS should be to determine the outcome of an attack (success or failure). In most cases, an alert simply indicates that an attempt to intrude has been made. It is then the responsibility of the analyst to search for correlated activities to determine the outcome of the attack. If an ANIDS were to present the analyst with a list of other alerts generated by the target host, and a summary of other (non-alert) traffic, the evaluation of the outcome can be greatly accelerated.

- **Legality of data collected**: The legality of the data collected by an ANIDS is of extreme importance if any legal activity may be pursued against the attacker. A disturbingly large number of systems do not collect the actual network packets; instead they simply record their own interpretation of events. A more robust system must capture and store network traffic in addition to simply raising the alert.

- **Signature updates**: An ANIDS should have the ability to detect new types of intrusions and effectively update signatures dynamically.
Table 3: Normal and attack traffic information for KDDCup99 dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Dos</th>
<th>Probe</th>
<th>nZr</th>
<th>rZl</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>10% KDD</td>
<td>391458</td>
<td>4107</td>
<td>52</td>
<td>1126</td>
<td>97277</td>
</tr>
<tr>
<td>Corrected KDD</td>
<td>229853</td>
<td>4107</td>
<td>52</td>
<td>1126</td>
<td>97277</td>
</tr>
<tr>
<td>Whole KDD</td>
<td>229853</td>
<td>4107</td>
<td>52</td>
<td>1126</td>
<td>97277</td>
</tr>
</tbody>
</table>

4 Evaluation Datasets

Researchers use various datasets for testing and evaluating different anomaly detection methods. It is very difficult to evaluate an anomaly detection system based on live network traffic or any raw network traffic. That is why some benchmark datasets are used for evaluating an anomaly detection system. Some of these are discussed in brief.

Lincoln Lab Datasets: In 1998, MIT's Lincoln Lab performed an evaluation of anomaly-based intrusion detection systems [26]. To perform this evaluation, the Lab generated the DARPA training and testing datasets. Both datasets contain attacks and background traffic. In 1999, the KDDCup competition used a subset of the preprocessed DARPA training and test data supplied by Solvo and Lee [27], the principal researchers for the DARPA evaluation. The raw training data was about four gigabytes of compressed binary tcpdump data from seven weeks of network traffic. This was processed into about five million connection records. The dataset is known as the KDD99 dataset and a summary of it is reported in Table 3. There are four main types of attacks that are identified: denial of service, remote-to-local, user-to-root, and surveillance/probing. Background traffic is simulated and the attacks are all known. The training set, consisting of seven weeks of labeled data, is available to the developers of intrusion detection systems. The testing set also consists of simulated background traffic and known attacks, including some attacks that are not present in the training set.

LBNL Datasets: This dataset can be obtained from Lawrence Berkeley National Laboratory (LBNL) in the USA. Traffic in this dataset is comprised of packet-level incoming, outgoing, and internally routed traffic streams at the LBNL edge routers. Traffic was anonymized using the tcpmkpub tool [28]. The main applications observed in internal and external traffic are Web (i.e., HTTP), Email, and Name Services. It identifies attack traffic by isolating the corresponding scans in aggregate traffic traces. The outgoing TCP scans in the dataset follow LBNL hosts for resetting the TCP connection. Clearly, the attack rate is significantly lower than the background traffic rate (see Table 4 for detailed statistics).

Table 4: Background and Attack Traffic Information for the LBNL Datasets

<table>
<thead>
<tr>
<th>Date</th>
<th>Duration (min)</th>
<th>LBNL Hosts</th>
<th>Remote Hosts</th>
<th>Background Traffic rate (packet/sec)</th>
<th>Attack Traffic rate (packet/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/04/2004</td>
<td>10 min</td>
<td>5,767</td>
<td>3,342</td>
<td>8.47</td>
<td>0.41</td>
</tr>
<tr>
<td>12/15/2004</td>
<td>60 min</td>
<td>5,767</td>
<td>10,478</td>
<td>3.5</td>
<td>0.061</td>
</tr>
<tr>
<td>12/16/2004</td>
<td>60 min</td>
<td>5,767</td>
<td>7,138</td>
<td>243.83</td>
<td>72</td>
</tr>
</tbody>
</table>

2 (online) http://www.tcpdump.org/
Background Traffic Information for Four Endpoints with High and Low Rates

<table>
<thead>
<tr>
<th>Endpoint ID</th>
<th>Endpoint Type</th>
<th>Duration (months)</th>
<th>Total Sessions</th>
<th>Mean Session Rate (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Home</td>
<td>1</td>
<td>6.73399</td>
<td>1.92</td>
</tr>
<tr>
<td>4</td>
<td>Home</td>
<td>2</td>
<td>4.24,585</td>
<td>3.26</td>
</tr>
<tr>
<td>6</td>
<td>University</td>
<td>7</td>
<td>60,939</td>
<td>3.19</td>
</tr>
<tr>
<td>10</td>
<td>University</td>
<td>13</td>
<td>1.52,048</td>
<td>0.21</td>
</tr>
</tbody>
</table>

**End-point Datasets:** The traffic rates observed at the end-points are much lower than those at the LBNL routers. The large traffic volumes of home computers are also evident from their high mean numbers of sessions per second. To generate attack traffic, the analysts infected VMs on the end-points with different malwares: Zotob.G, Forbot-FU, Sdbot-AFR, Dloader-NY, So-Big.E@mm, MyDoom.A@mm, Blaster, Rbot-AQJ, and RBOT.CCC. Details of the malwares can be found at 3. The attack traffic logged at the end-points is mostly comprised of outgoing port scans. Moreover, the attack traffic rates at the end-points are generally much higher than the background traffic rates of LBNL datasets. For each malware, attack traffic of 15 minutes duration was inserted in the background traffic of each end-point at a random time instance. The background and attack traffic statistics of the end-point datasets are given in Table 5 and Table 6.

**Network Trace Datasets:** Network traces [29] captured from live networks is often used for testing intrusion detection systems. The main advantage of using network traces is that the results do not demonstrate any bias due to artifacts from simulated data that are not representative of actual data. However, when network traces are available, they are often limited. For example, the traces might only contain packet header data, or might be summarized even further into flow-level information.

### Table 6: Endpoint Attack Traffic for Two High and Two Low-rate Worms

<table>
<thead>
<tr>
<th>Malware</th>
<th>Release Date</th>
<th>Avg. Scan rate (sp/s)</th>
<th>Port(s) Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dloader-NY</td>
<td>Jul 2005</td>
<td>46.84 sps</td>
<td>TCP 1,35,139</td>
</tr>
<tr>
<td>Forbot-FU</td>
<td>Sept 2005</td>
<td>52.33 sps</td>
<td>TCP 445</td>
</tr>
<tr>
<td>Rbot-AQJ</td>
<td>Oct 2005</td>
<td>0.68 sps</td>
<td>TCP 1,39,709</td>
</tr>
<tr>
<td>MyDoom-A</td>
<td>Jan 2006</td>
<td>0.14 sps</td>
<td>TCP 3127,3198</td>
</tr>
</tbody>
</table>

### 4.1 Discussion

Based on our survey on the existing incremental anomaly detection techniques in the context of preprocessed or real-life datasets, we observe that:

- Most datasets are useful for offline anomaly detection;
- Some existing datasets are not labeled (LBNL, Network trace) and do not have any attack statistics (for e.g., LBNL, Network trace);
- There is both packet and flow level information in the network trace dataset;

Here, we report a comparison (Table 7) of the existing datasets in terms of different parameters, such that attack information, traffic information in terms of normal or attack etc.

---

3 Symantec security response (online) http://securityresponse.symantec.com/avcenter
Table 7: Comparison among intrusion datasets

<table>
<thead>
<tr>
<th>Name of the dataset</th>
<th>Categories of attacks</th>
<th>Benchmark</th>
<th>Traffic type (normal)</th>
<th>Real/Non-real time</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDDCup99</td>
<td>DOS, Probe, U2R, R2L</td>
<td>B</td>
<td>N/A</td>
<td>N</td>
</tr>
<tr>
<td>LBINL</td>
<td></td>
<td>B</td>
<td>N/A</td>
<td>N</td>
</tr>
<tr>
<td>Real-Point</td>
<td>Zotob-L, Furbot-FU,</td>
<td>N</td>
<td>N/A</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>Sbot-AFR, Diloder-NY,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>So-Big.E@mm, MyDoom.A@mm, Blaster, Rbot-AQI, and Rbot-CCC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Traces</td>
<td></td>
<td>N</td>
<td>N/A</td>
<td>X</td>
</tr>
</tbody>
</table>

5 Evaluation Criteria and Analysis

Benchmark intrusion datasets play an important role while evaluating an attack detection system. But there is only one well known and commonly available benchmark dataset (i.e., KDDCup99) for performance analysis of IDSs. Researchers analyze their detection methods based on the live network traffic (i.e., network trace); but they cannot claim that the detection methods work in all situations. Some of the evaluation criteria are discussed below in brief.

5.1 Metrics

The intrusion detection community commonly uses four metrics. The first two are detection rate and false positive rate and, conversely, true and false negative rates [30]. Two other metrics: effectiveness and efficiency are defined by Staniford et al. [31]. Effectiveness is defined as the ratio of detected scans (i.e., true positives) to all scans (true positives plus false negatives). Similarly, efficiency is defined as the ratio of the number of identified scans (i.e., true positives) to all cases flagged as scan (true positives plus false positives) [31], and is the same as the detection rate defined previously.

5.2 ROC Analysis

Relative Operating Characteristic (ROC) curve is often used to evaluate the performance of a particular detector. This approach is used by Lincoln Lab for evaluation of anomaly-based detection systems and discussed in detail by McHugh [32]. An ROC curve has false positive rate on its x-axis and true positive rate on its y-axis, thus moving from (0, 0) at the origin to (1, 1). The detection system must return a likelihood score between 0 and 1, when it detects an intrusion in order for the ROC curve to provide meaningful results. The ROC curve can be used to determine how well the overall system performs, the most appropriate threshold values given acceptable true and false positive rates, and to compare different detection systems.

5.3 Complexity and Delay Comparison

The training and classification time taken by anomaly detectors as well as their training and run-time memory requirements can be computed using the hprof tool [33]. Contrary to common intuition, complexity does not translate directly into accuracy of an anomaly detector. A delay value of $\infty$ is listed if an attack is not detected altogether. The detection delay is reasonable (less than 1 second) for all the anomaly detectors we surveyed.
6 Research Issues and Challenges

Based on our survey of published papers on incremental anomaly detectors, we observe that most techniques have been validated using the KDD99 intrusion datasets in an offline mode. However, the effectiveness of an ANIDS based on incremental approach can only be judged in a real-time environment. Following are some of the research issues we have identified in this area.

- Most existing IDSs have been found inadequate with new networking paradigms currently used for both wired and wireless communication. Thus, adaptation to new network paradigms needs to be explored.
- Most existing methods are dependent on multiple input parameters. Improper estimation of these parameters leads to high false alarm rates.
- The clustering techniques that are used by anomaly detectors need to be faster and scalable over high dimensional and voluminous mixed type data.
- Lack of labeled datasets for training or validation is a crucial issue that needs to be addressed. The KDD99 datasets are out-of-date. New valid datasets need to be created and made available to researchers and practitioners. Developing a reasonably exhaustive dataset for training or validation for use in supervised or semi-supervised anomaly detection approaches is a challenging task.
- Estimation of unbiased anomaly scores for periodic, random or bursty attack scenarios is another challenging issue.
- Lack of standard labeling strategies is a major bottleneck in the accurate recognition of normal as well as attack clusters.
- Development of pre or post-processing mechanisms for false alarm minimization is necessary.
- Handling of changing traffic pattern remains a formidable problem to address.

7 Conclusion

In this paper, we have examined the state of modern incremental anomaly detection approaches in the context of computer network traffic. The discussion follows two well-known criteria for categorizing of anomaly detection: detection strategy and data source. Most anomaly detection approaches have been evaluated using Lincoln Lab (i.e. KDDCup99 intrusion dataset), Network Traces and LBNL datasets. Experiments have demonstrated that for different types of attacks, some anomaly detection approaches are more successful than others. Therefore, ample scope exists for working toward solutions that maintain high detection rate while lowering false alarm rate. Incremental learning approaches that combine data mining, neural network and threshold-based analysis for the anomaly detection have shown great promise in this area.

Acknowledgements This work is a part of a research project funded by the Department of Information Technology, Govt. of India.

References


4 Bro (online) http://www.bro-ids.org/
5 SNORT (online) http://www.snort.org/


