Accuracy Improving Guidelines for Network-based Anomaly Detection Systems

By:
Ayesha Binte Ashfaq

A THESIS

Submitted to
The School of Electrical Engineering and Computer Science
National University of Sciences and Technology (SEECS, NUST)
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Thesis Supervisor:
Dr. Syed Ali Khayam

Committee Members:
Dr. Fauzan Mirza
Mr. Ali Sajjad
Mr. Qasim Rajput

Department of Computing (DoC)

January 2009
ABSTRACT

Since the seminal 1998/1999 DARPA evaluations of intrusion detection systems, network attacks have evolved considerably. In particular, after the CodeRed worm of 2001, the volume and sophistication of self-propagating malicious code threats have been increasing at an alarming rate. Many anomaly detectors have been proposed, especially in the past few years, to combat these new and emerging network attacks. Due to the rapidly evolving nature of network attacks, a considerable paradigm shift has taken place with focus now on Network-based Anomaly Detection Systems (NADSs) that can detect zero-day attacks. Contemporary NADS borrow concepts from a variety of theoretical fields (e.g., Information theory, stochastic and machine learning, signal processing, etc.) to model benign traffic. In this work, we survey and taxonomize recent NADSs with an aim to learn from their strengths and weaknesses. To this end, we propose a multidimensional taxonomy which allows a systematic classification of NADSs. At this time, it is also important to evaluate existing anomaly detectors to determine and learn from their strengths and shortcomings. Thus as part of this research work, we also evaluate the performance of eight prominent network-based anomaly detectors under malicious portscan attacks. These NADSs are evaluated on three criteria: accuracy (ROC curves), scalability (with respect to varying normal and attack traffic rates, and deployment points) and detection delay. These criteria are evaluated using two independently collected datasets with complementary strengths. We see that a few of the anomaly detectors provide high accuracy on one of the two datasets, but are unable to scale their accuracy across the datasets. Based on our experiments and the proposed taxonomy, we identify promising guidelines to improve the accuracy and scalability of existing and future anomaly detectors, which is one of the main contributions of this work. We show that the proposed guidelines provide considerable and consistent accuracy improvements for all evaluated NADSs.
To my mom and dad, for their love and support
ACKNOWLEDGEMENTS

First and foremost, I am immensely thankful to Almighty Allah for letting me pursue and fulfill my dreams. Nothing could have been possible without His blessings.

More than anything, I would like to thank my mom and dad who have endured me through this period of research that has been extremely demanding and challenging. Thanks to my mom for always being so motherly in everything she does. Especial thanks to my dad for not letting me worry about anything but work. This last one year of research could not have been possible without the unending support from my parents. They have always supported and encouraged me to do my best in all matters of life. To them I dedicate this thesis.

I am immensely grateful to my Grandfather for always being appreciative of my hard work, for his unending love and support and for encouraging me during the hardest of times.

I would also like to thank Haroon Bhai for all the care and love he showered on me during all these years. I would also like to thank Omer for being such a wonderful person that he is. Nothing would have been fun without you guys.

My heartfelt thanks to everyone at WISNET Lab, my committee members and all others who contributed in any way towards the successful completion of this thesis.

Finally, this thesis would not have been possible without the expert guidance of my esteemed advisor, Dr. Syed Ali Khayam, who has been a great source of inspiration for me during these years of research. Not only has he been readily available for me, as he so generously is for all of his students, but he always responded to any queries that I might have, read through my draft copies, listened to my moaning and complaining and supported me every step of the way.

Despite all the assistance provided by Dr. Ali Khayam and others, I alone remain responsible for any errors or omissions which may unwittingly remain.
## CONTENTS

### I INTRODUCTION

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-A</td>
<td>Motivation</td>
<td>1</td>
</tr>
<tr>
<td>I-B</td>
<td>Contribution</td>
<td>2</td>
</tr>
<tr>
<td>I-C</td>
<td>Thesis Organization</td>
<td>4</td>
</tr>
</tbody>
</table>

### II BACKGROUND AND RELATED WORK

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>II-A</td>
<td>IDS Detection Methods</td>
<td>7</td>
</tr>
<tr>
<td>II-B</td>
<td>Accuracy Criteria</td>
<td>9</td>
</tr>
<tr>
<td>II-C</td>
<td>Related Work</td>
<td>11</td>
</tr>
</tbody>
</table>

### III ADS EVALUATION FRAMEWORK

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>III-A</td>
<td>Anomaly Detection Algorithms</td>
<td>13</td>
</tr>
<tr>
<td>III-A1</td>
<td>Rate Limiting</td>
<td>14</td>
</tr>
<tr>
<td>III-A2</td>
<td>Threshold Random Walk (TRW) Algorithm</td>
<td>14</td>
</tr>
<tr>
<td>III-A3</td>
<td>TRW with Credit-based Rate Limiting (TRW-CB)</td>
<td>15</td>
</tr>
<tr>
<td>III-A4</td>
<td>Maximum Entropy Method</td>
<td>15</td>
</tr>
<tr>
<td>III-A5</td>
<td>Packet Header Anomaly Detection (PHAD)</td>
<td>15</td>
</tr>
<tr>
<td>III-A6</td>
<td>PCA-based Subspace Method</td>
<td>16</td>
</tr>
<tr>
<td>III-A7</td>
<td>Kalman Filter based Detection</td>
<td>16</td>
</tr>
<tr>
<td>III-A8</td>
<td>Next-Generation Intrusion Detection Expert System (NIDES)</td>
<td>17</td>
</tr>
<tr>
<td>III-B</td>
<td>EVALUATION DATASETS</td>
<td>17</td>
</tr>
<tr>
<td>III-B1</td>
<td>The LBNL Dataset</td>
<td>18</td>
</tr>
<tr>
<td>III-B2</td>
<td>Endpoint Dataset</td>
<td>20</td>
</tr>
</tbody>
</table>

### IV TAXONOMY OF NETWORK ANOMALY DETECTION SYSTEMS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV-A</td>
<td>Taxonomy based on Learning Model and Detection Principles</td>
<td>23</td>
</tr>
<tr>
<td>IV-B</td>
<td>Taxonomy based on Traffic Characteristics</td>
<td>26</td>
</tr>
</tbody>
</table>

### V PERFORMANCE EVALUATION

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>V-A</td>
<td>Accuracy and Scalability Comparison</td>
<td>31</td>
</tr>
</tbody>
</table>
VI LESSONS LEARNT

VI-A Objectives of the Study .................................................. 39
VI-B Why do Some Classes of the Taxonomy Perform Better than Others? .......... 40
VI-C Promising Guidelines to Improve the Accuracy of Existing and Future NADSs .... 42
    VI-C1 Multi-Window Classification (Guideline 4) ................................ 42
    VI-C2 Feature Space Extension (Guideline 5) ..................................... 44
    VI-C3 Traffic Splitting (Guideline 6) ............................................... 45
VI-D NADS Accuracy Improvements by Traffic Splitting and Multi-Window Classification 48
VI-E NADS Accuracy Improvements by Feature Space Extension and Multi-Window Classification ................................................................. 48
VI-F NADS Accuracy Improvements by Jointly Using the Proposed Guidelines ......... 51

VII CONCLUSION ............................................................... 55

References ................................................................. 57
LIST OF FIGURES

1. IDS Classification. .................................................. 8
2. ROC Curve. ............................................................. 9
3. ROC Curve Comparison. ............................................. 10
4. Taxonomy of the network anomaly detectors based on learning model and detection principles. 26
5. Taxonomy of the network anomaly detectors based on traffic scale and semantics. ............... 28
6. ROC analysis on the endpoint dataset; each ROC is averaged over 13 endpoints with 12 attacks per endpoint and 100 instances per attack. ......................................................... 32
7. Comparison of the Maximum Entropy, TRW and TRW-CB algorithms. .............................. 33
8. Detection and false alarm rates for the subspace method [25]. .............................................. 34
9. ROC curves for the lowest and highest rate attack in the endpoint dataset; results averaged over 12 endpoints with 100 instances of each attack. ......................................................... 36
10. ROC analysis on the LBNL dataset. .................................................................................. 37
11. An example of Multi-window classification: Maximum-Entropy Detector’s output on five LBNL time windows containing benign and malicious traffic. ..................................................... 43
12. PHAD Detector’s output for a single and multiple features analyzed for anomaly detection on LBNL dataset. .................................................................................. 45
13. An example of Traffic Splitting: Segregated and aggregate traffic rates during a TCP and a UDP attack. .................................................................................. 47
14. ROC-based accuracy comparison of TRW and TRW-CB with and without traffic splitting and multi-window classification. ......................................................... 49
15. ROC-based accuracy comparison of PHAD and Maximum Entropy with and without feature space extension and multi-window classification. ......................... 50
16. Block diagram of Network-based Anomaly Detection System that jointly employs the proposed guidelines. ......................................................... 52
17. ROC-based accuracy evaluation of original and improved NADS algorithms. .................... 53
# List of Tables

<table>
<thead>
<tr>
<th></th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Background Traffic Information for the LBNL Dataset</td>
<td>19</td>
</tr>
<tr>
<td>II</td>
<td>Background Traffic Information for Four Endpoints with High and Low Rates</td>
<td>20</td>
</tr>
<tr>
<td>III</td>
<td>Endpoint Attack Traffic for Two High- and Two Low-rate Worms</td>
<td>22</td>
</tr>
<tr>
<td>IV</td>
<td>Detection Delay of the Anomaly Detectors</td>
<td>38</td>
</tr>
<tr>
<td>V</td>
<td>Detection Delay of the Improved variants of Maximum Entropy and PHAD</td>
<td>51</td>
</tr>
</tbody>
</table>
CHAPTER 1

I. INTRODUCTION

With an increasing penetration of broadband Internet connectivity and an exponential growth in the worldwide IT infrastructure, individuals and organizations now rely heavily on the Internet for their communication and business needs. While such readily-available network connectivity facilitates operational efficiency and networking, systems connected to the Internet are inherently vulnerable to network attacks. These attacks have been growing in their number and sophistication over the last few years [1]. Malware, botnets, spam, phishing, and denial of service attacks have become continuous and imminent threats for today’s networks and hosts [1], [2]. Financial losses due to these attacks are in the orders of billions of dollars\(^1\). In addition to the short-term revenue losses for businesses and enterprises, network attacks also compromise information confidentiality/integrity and cause disruption of service, thus resulting in a long-term loss of credibility.

A. Motivation

Since the CodeRed worm of 2001, malware attacks have emerged as one of the most prevalent and potent threats to network and host security. Many network-based anomaly detection systems (NADSs) have been proposed in the past few years to detect novel network attacks [14]–[33]. Since malicious portscans are the vehicle used by malware and other automated tools to locate and compromise vulnerable hosts, some of these anomaly detectors are designed specifically for portscan detection [14]–[21], [29], while other detectors are more general-purpose and detect any anomalous traffic trend [22]–[28], [30]. Most of the network-based anomaly detectors, model and leverage deep-rooted statistical properties of

\(^1\)Economic losses to recover from the CodeRed worm alone are estimated at $2.6 billion [3].
benign traffic to detect anomalous behavior. A variety of theoretical frameworks—including stochastic, machine learning, information-theoretic and signal processing frameworks—have been used to develop robust models of normal behavior and/or to detect/flag deviations from that model. However, very little effort has been expended into the taxonomization and comparative evaluation of these recent NADSs for the portscan detection problem.

The main challenge of NADSs is to define a robust model of normal traffic behavior. In particular, an accurate model needs to cater for changes in normal traffic behavior over time. Such changes in normal traffic behavior lead to potentially low detection rates and high false alarm rates of NADSs. In view of the vast research literature on network anomaly detection, it is important to survey, taxonomize and evaluate existing NADSs. While a systematic taxonomy and comparative performance evaluation facilitates study of various aspects of contemporary anomaly detectors, more importantly they should reveal the strengths and shortcomings of existing NADSs and consequently lead to promising design guidelines that can be used to improve the accuracy of NADSs.

B. Contribution

In this research we aim to classify and evaluate prominent Network-based Anomaly Detection systems to learn from their strengths and to propose promising guidelines to improve the accuracy of current and future anomaly detectors.

Following are the research objectives set in accordance to the above stated problem statement:

- to taxonomize NADS inorder to develop better understanding of their strengths and weaknesses;
- to quantify and compare the accuracies of some of the prominent detectors under varying rates of attack and normal traffic and at different points of deployment;
• to identify promising traffic features and theoretical frameworks for portscan anomaly detection;
• to investigate the accuracy of contemporary anomaly detectors with respect to their detection delay;
• to identify a set of promising portscan detection guidelines that build on the strengths and avoid the weaknesses of the evaluated anomaly detectors; and finally
• to provide experimental results for the accuracy improvements achieved by the proposed guidelines.

In light of these objectives, we initially propose two multidimensional taxonomies of network-based anomaly detection systems with an aim to obtain insights into why some NADSs perform better than others. Our first taxonomy classifies NADSs based on their learning behavior and detection principles. The second taxonomy categorizes NADSs using their traffic scales and semantics. Moreover, we evaluate and compare eight prominent network-based anomaly detectors on two public portscan datasets.

In order to quantify and compare the accuracies and delay characteristics of prominent NADS proposed in the last few years, we also perform a comparative performance evaluation of a diverse set of anomaly detection systems. The anomaly detectors compared in this work were proposed in [14], [17], [18], [22], [25], [28], [30] and [31]. These NADSs are chosen because they employ very different traffic features and theoretical frameworks for anomaly detection. Moreover, most of these detectors are frequently used for performance benchmarking in the intrusion detection research literature [16], [19]–[21], [23], [24], [26], [27], and [29]. Some of these NADSs have been designed for and evaluated at endpoints while others have been tailored towards organization/ISP gateways. Similarly, some detectors are designed for portscan detection, while others are general-purpose NADSs. This diversity
allows us to determine how much, if any, performance improvement is provided by portscan NADSs over general-purpose ADSs.

For performance evaluation of the anomaly detectors, we use two independently-collected datasets with complementary strengths. The first dataset is an enterprise traffic dataset collected at the edge router of the Lawrence Berkeley National Lab (LBNL) [43]. Attack traffic in this dataset mostly comprises high-rate background traffic and low-rate outgoing scans. The second dataset comprises traffic data collected at network endpoints in home, university and office settings. Background traffic rates of these endpoints are relatively low as compared to the LBNL dataset, but the endpoint attack data contains relatively high-rate outgoing scan traffic.

We evaluate these NADSs on three criteria: accuracy, scalability, and detection delay. Accuracy is evaluated by comparing ROC (false alarms per day versus detection rate) characteristics of the NADSs. Scalability is evaluated with respect to different background and attack traffic rates. Since the two datasets used in this study are collected at different network entities and contain attacks with different characteristics, evaluation over these datasets allows us to compare the scalability of the proposed NADSs under varying traffic volumes. Detection delay is evaluated separately for high- and low-rate attacks.

Based on our findings, we propose a few promising guidelines to improve the accuracy and scalability of existing and future NADSs. Our results show that the proposed guidelines result in an average detection rate increase of 5% to 10%, while reducing the false alarm rates up to 50%.

C. Thesis Organization

The remainder of this document is structured as follows:
Chapter 2. provides the necessary background on anomaly detection systems, ROC analysis and outlines the existing research in the areas of IDS classification and evaluation.

Chapter 3. gives a brief description of the anomaly detection systems that have been evaluated as part of this study. Moreover it provides detailed statistics related to the datasets used for the performance evaluation. The background as well as attack traffic rates along with the formation of the combined datasets is explained in detail. Our findings show that some of the evaluated anomaly detectors provide reasonable accuracy with low detection delay. However, these detectors do not provide sustained accuracy on both the datasets.

In Chapter 4. we propose two multi-dimensional taxonomies. One of the taxonomy classifies NADSs based on learning behavior and detection principles while the second taxonomy classifies the detectors based on traffic semantics and scale. Our analysis shows that on endpoints, the best performing NADSs train on benign traffic profiles and are self learning systems, whereas router based NADSs that employ intelligent detection principles provide high accuracy. These NADSs generally operate on meso-scaled protocol level traffic features.

Chapter 5. provides the comparative performance evaluation results in the form of ROC curves on the two datasets. NADSs are also compared separately on low as well as high rate worms. Moreover, detection delay comparison has also been performed for the LBNL and the endpoint dataset, as well as for low and high rate worms. We evaluation shows that the detectors are unable to scale their accuracies for different points of network deployment.

Chapter 6. provides the lessons learnt from the classification and the comparative performance evaluation of prominent NADSs. In addition to this, promising portscan detection guidelines are proposed to improve the performance of existing and future NADSs. Experimental results are also provided in the form of ROC curves for the performance improvements realized by jointly using the proposed guidelines.
Chapter 7 summarizes the key conclusions of this work.
CHAPTER 2

II. BACKGROUND AND RELATED WORK

In order to combat the rapidly-evolving malicious attacks, network intrusion detection methods have also become increasingly sophisticated. Intrusion Detection Systems (IDS) offer techniques for modeling and recognizing normal and abusive system behavior. These have potential to mitigate or prevent malicious attacks, if updated signatures or novel attack recognition and response capabilities are in place. Such methodologies include: statistical models, immune system approaches, protocol verification, file checking, neural networks, whitelisting, expression matching, state transition analysis, dedicated languages, genetic algorithms and burglar alarms. But, by choosing an attack method suitably an attacker has the possibility of escaping the detection of an IDS.

A. IDS Detection Methods

In broad terms, the field of intrusion detection comprises two types of detection methods: misuse detection (also known as signature detection) and anomaly detection. Misuse detection, the predominant detection method employed in today’s anti-virus software, requires a signature of an attack to be known before the attack can be detected. While such signature-based detectors can provide 100% detection rates for known attacks, they have an inherent limitation of not being able to detect new or previously-unseen attacks; a 468% increase in previously-unseen attacks was reported over just a six month period in 2007 [1]. Moreover, development and dissemination of attack signatures require human intervention and therefore misuse detectors are finding it difficult to cope with rapidly-evolving network intrusions. On the other end of the intrusion detection spectrum are Network-based Anomaly Detection Systems (NADSs) which model the benign or normal traffic behavior of a network or host and detect significant deviations from this model to identify anomalies in network traffic.
Since NADSs rely on normal traffic behavior for attack detection, they can detect previously-unknown attacks. Consequently, significant research effort has been focused on development of NADSs in the past few years [4].

Anomaly detection systems can further be categorized into either host based systems or network based systems [5]:

- **Network-based ADS**: Network based systems detect anomalies by analyzing unusual network traffic patterns [4]
- **Host based ADS**: Host-based systems detect anomalies by monitoring an endpoint’s operating system (OS) behavior, for instance by tracking OS audit logs, processes, command-lines or keystrokes [6] - [10]

This Intrusion detection classification is shown in Figure. 1. These network-based IDSs can be either endpoint or perimeter based depending on the traffic analyzed for anomaly identification.

\(^2\)Interestingly, the promise and advantages of anomaly detectors over signature detectors were identified by the seminal DARPA-funded IDS evaluation studies much before the CodeRed worm [11], [55].
B. Accuracy Criteria

The accuracy of an intrusion detection system is generally evaluated on two competing criteria:

- **Detection rate**: What fraction of anomalies are correctly detected by the IDS.
- **False Alarm rate**: What fraction of the total anomalies detected by the IDS are in fact benign data.

To understand the tradeoff between these accuracy criteria, consider an IDS that classifies all the test data as anomalous. Such an IDS will achieve 100% detection rate, but at the cost of an unacceptable 100% false alarm rate. At the other end of this spectrum, consider an IDS that classifies all of the test data as normal. This IDS will have an attractive 0% false alarm rate, but is useless because it does not detect any anomalies. To evaluate the accuracy of an IDS, detection thresholds of the IDS are tuned and for each threshold value the detection rate is plotted against the false alarm rate. Each point on such a plot, referred to as an ROC curve [11], represents performance results for one configuration (or threshold value) whereas the curve represents the behavior for the complete set of configurations.
A receiver operating characteristics (ROC) curve is a technique for visualizing, organizing and selecting classifiers based on their performance [12]–[13]. A typical ROC curve is shown in Fig. 2. The ROC curve is the plot of TPR (true positive rate) vs. FPR (false positive rate) for different threshold values.

The diagonal line \( y = x \) represents the strategy of random guessing. For example, if a classifier randomly guesses the positive class half the time, it can be expected to get half the positives and half the negatives correct; this yields the point \( (0.5, 0.5) \) in ROC space.

Any classifier that appears in the lower right triangle performs worse than random guessing. This triangle is therefore usually empty in ROC graphs.

What is of interest are the curves in the upper left triangle. Higher the curve, better the performance of the classifier. This is shown in Figure 3.
C. Related Work

We focus on prior IDS/ADS classification and evaluation studies. Details of anomaly detectors evaluated in this work are deferred to subsequent sections.

Taxonomy and classification in any field assists systematic study of various aspects of the main area under study. Intrusion detection systems have been a key aspect of the field of network security since its inception. Numerous surveys have been performed to study the intrusion detection systems [34]–[38]. Along side the study of the ADSs efforts have also been vested in their categorization and classification [38] – [41]. These few taxonomies proposed in the past, for classifying the IDS on numerous fronts, have rendered valuable but fewer insights into the intricate details of the IDSes. The classification by Debar [39] was probably the first taxonomy and survey of intrusion detection systems. A two dimensional behavioral classification of IDSs was performed by Axelsson [40] in 2000 and a similar but more comprehensive classification was proposed by M. Almgren [41] in 2003. Recently a reasonably detailed and comprehensive taxonomy was proposed by Juan M. E-T [42] in 2004. Though these taxonomies, proposed to date, provide understanding of intrusion detection systems along various dimensions, but lack to correlate these dimensions for a better and deeper understanding of the detection aspects of the ADSs.

Performance evaluation of IDSs received significant attention from the industry and academia in the late 1990’s [48]–[62]. However, in the past few years, only four studies have performed comparative comparison of anomaly detectors [45]–[47]. Similarly, very few prior studies have performed ROC analysis of the evaluated IDSs. Still fewer studies have made their evaluation datasets available online.

DARPA-funded IDS evaluation studies by the MIT Lincoln Lab in 1998 and 1999 represent a shift in the IDS evaluation methodology [11], [55]. Datasets used in these studies were
made publicly available [56] and the ROC method used in these studies has since become the de facto standard for IDS accuracy evaluation. While some shortcomings of the DARPA evaluation have been highlighted [64], [65], in the absence of other benchmarks, the results and datasets of this study have been used extensively in subsequent works. In the present study’s context, the DARPA dataset is somewhat dated.

The four recent ADS evaluation studies focus on specific types of detectors and attacks [45]–[47]. The study by Wong et al. [45] is most relevant in the present context. Wong et al. [45] evaluated four variants of the rate limiting detector under portscan attacks at two different network points [14]–[21]. Two findings of this study are pertinent to the present work: 1) classical rate limiting is not an effective technique for portscan detection, and 2) rate limiting can operate on aggregate-level DNS traffic and hence can potentially scale to core-level deployments. Attack and background traffic data used in this study are not publicly available.

Ingham and Inoue [46] compared seven HTTP anomaly detection techniques under real-world attacks reported at public databases. These authors report the same evaluation difficulties that were faced by us: 1) Some anomaly detectors are not described completely; 2) Implementation source code is not available; and 3) labeled data used for algorithm evaluation are not publicly available. Consequently, the authors in [46] make their implementation and attack data publicly available “to encourage further experimentation”. We subscribe to the same viewpoint and therefore all data and implementation used in this project are available online [44]. Lazarevic et al. performed a comparative analysis of four data mining based anomaly detection techniques in [47]. The live network traffic data used by this study is not publicly available.
CHAPTER 3

III. ADS EVALUATION FRAMEWORK

In this section we would give details regarding the evaluated anomaly detection systems and the datasets used for the evaluation. Moreover, characteristic features of the two datasets will also be provided.

A. Anomaly Detection Algorithms

We will focus on network-based anomaly detectors and compare the anomaly detectors proposed in [14], [17], [18], [22], [25], [28], [30], and [31]. Most of these detectors are quite popular and used frequently for performance comparison and benchmarking in the ID research community. Improvements to these algorithms have also been proposed in [16], [19]–[21], [23], [24], [26], [27], [29], and [45].

Before briefly describing these detectors, we highlight that some of these detectors are designed specifically for portscan detection, while others are general-purpose network anomaly detectors. More generically, based on the proposed taxonomies, the algorithms evaluated in this study can be subdivided into numerous NADS classes shown in Fig. 4 and Fig. 5. Clearly, the evaluated ADSs are quite diverse in their traffic features as well as their detection frameworks. These ADSs range from very simple rule modelling systems like PHAD [22] to very complex and theoretically-inclined partially Self-Learning systems like the PCA-based subspace method [25] and the Sequential Hypothesis Testing technique [17]. This diversity is introduced to achieve the following objectives: a) to correlate the performance of the NADSs with the classes of the taxonomy in which they fall; b) to identify promising traffic features and theoretical frameworks for portscan anomaly detection; c) to investigate the accuracy,

3Some promising commercial ADSs are also available in the market now [32], [33]. We did not have access to these ADSs, and therefore these commercial products are not evaluated in this study.
and delays of these anomaly detectors under different attack and normal traffic scenarios and at different points of deployment in the network; and d) to identify a set of promising portscan detection guidelines that build on the strengths and avoid the weaknesses of the evaluated anomaly detectors.

We provide brief descriptions of the evaluated algorithms. Majorly we will focus on the algorithm adaptation and parameter tuning for the datasets under consideration. Readers are referred to [14], [17], [18], [22], [25], [28], [30], and [31] for details of the algorithms. For techniques operating on fixed-sized time windows, we use a window of 20 seconds. All other parameters not mentioned in this section are the same as those described in the algorithms’ respective papers.

1) Rate Limiting: Rate limiting [14], [15] detects anomalous connection behavior by relying on the premise that an infected host will try to connect to many different machines in a short period of time. Rate limiting detects portscans by putting new connections exceeding a certain threshold in a queue. An alarm is raised when the queue length, $\eta_q$, exceeds a threshold. ROCs for endpoints are generated by varying $\eta_q = \mu + k\sigma$, where $\mu$ and $\sigma$ represent the sample mean and sample standard deviation of the connection rates in the training set, and $k = 0, 1, 2, \ldots$ is a positive integer. Large values of $k$ will provide low false alarm and detection rates, while small values will render high false alarm and detection rates. In the LBNL dataset, connection rate variance in the background traffic is more than the variance in the attack traffic. Therefore, to obtain a range of detection and false alarm rates for the LBNL dataset, we use a threshold of $\eta_q = w\mu$, with a varying parameter $0 \leq w \leq 1$, and the queue is varied between 5 and 100 sessions.

2) Threshold Random Walk (TRW) Algorithm: The TRW algorithm [17] detects incoming portscans by noting that the probability of a connection attempt being a success should be much higher
for a benign host than for a scanner. To leverage this observation, TRW uses sequential hypothesis testing (i.e., a likelihood ratio test) to classify whether or not a remote host is a scanner. We plot ROCs for this algorithm by setting different values of false alarm and detection rates and computing the likelihood ratio thresholds, $\eta_0$ and $\eta_1$, using the method described in [17].

3) TRW with Credit-based Rate Limiting (TRW-CB): A hybrid solution to leverage the complementary strengths of Rate Limiting and TRW was proposed by Schechter et al. [18]. Reverse TRW is an anomaly detector that limits the rate at which new connections are initiated by applying the sequential hypothesis testing in a reverse chronological order. A credit increase/decrease algorithm is used to slow down hosts that are experiencing unsuccessful connections. We plot ROCs for this technique for varying $\eta_0$ and $\eta_1$ as in the TRW case.

4) Maximum Entropy Method: This detector estimates the benign traffic distribution using maximum entropy estimation [30]. Training traffic is divided into 2,348 packet classes and maximum entropy estimation is then used to develop a baseline benign distribution for each packet class. Packet class distributions observed in real-time windows are then compared with the baseline distribution using the Kullback-Leibler (K-L) divergence measure. An alarm is raised if a packet class’ K-L divergence exceeds a threshold, $\eta_k$, more than $h$ times in the last $W$ windows of $t$ seconds each. Thus the Maximum Entropy method incurs a detection delay of at least $h \times t$ seconds. ROCs are generated by varying $\eta_k$.

5) Packet Header Anomaly Detection (PHAD): PHAD learns the normal range of values for all 33 fields in the Ethernet, IP, TCP, UDP and ICMP headers [22]. A score is assigned to each packet header field in the testing phase and the fields’ scores are summed to obtain a packet’s aggregate anomaly score. We evaluate PHAD-C32 [22] using the following packet header fields: source IP, destination IP, source port, destination port, protocol type and TCP flags.
Normal intervals for the six fields are learned from 5 days of training data. In the test data, fields’ values not falling in the learned intervals are flagged as suspect. Then the top $n$ packet score values are termed as anomalous. The value of $n$ is varied over a range to obtain ROC curves.

6) **PCA-based Subspace Method:** The subspace method uses Principal Component Analysis (PCA) to separate a link’s traffic measurement space into useful subspaces for analysis, with each subspace representing either benign or anomalous traffic behavior [25]. The authors proposed to apply PCA for domain reduction of the Origin-Destination (OD) flows in three dimensions: number of bytes, packets, IP-level OD flows. The top $k$ eigenvectors represent normal subspaces. It has been shown that most of the variance in a link’s traffic is generally captured by 5 principal components [25]. A recent study showed that the detection rate of PCA varies with the level and method of aggregation [71]. It was also concluded in [71] that it may be impractical to run a PCA-based anomaly detector over data aggregated at the level of OD flows. We evaluate the subspace method using the number of TCP flows aggregated in 10 minutes intervals. To generate ROC results, we changed the number of normal subspace as $k = 1, 2, \ldots, 15$. Since the principal components capture maximum variance of the data, as we increase $k$, the dimension of the residual subspace reduces and fewer observations are available for detection. In other words, as more and more principal components are selected as normal subspaces, the detection and false alarm rates decrease proportionally. Since there is no clear detection threshold, we could not obtain the whole range of ROC values for the subspace method. Nevertheless, we evaluate and report the subspace method’s accuracy results for varying number of principal components.

7) **Kalman Filter based Detection:** The Kalman filter based detector of [28] first filters out the normal traffic from the aggregate traffic, and then examines the residue for anomalies.
In [28], the Kalman Filter operated on SNMP data to detect anomalies traversing multiple links. Since SNMP data was not available to us in either dataset, we model the traffic as a 2-D vector $X_t$. The first element of $X_t$ is the total number of sessions (in the endpoint dataset) or packets (in the LBNL dataset), while the second element is the total number of distinct remote ports observed in the traffic. We defined a threshold, $\eta_f$ on the residue value $r$ to obtain ROC curves. Thresholding of $r$ is identical to the rate limiting case. An alarm is raised, if $r < -\eta_f$ or $r > \eta_f$.

8) Next-Generation Intrusion Detection Expert System (NIDES): NIDES [31] is a statistical anomaly detector that detects anomalies by comparing a long-term traffic rate profile against a short-term, real-time profile. An anomaly is reported if the $Q$ distribution of the real-time profile deviates considerably from the long-term values. After specific intervals, new value of $Q$ are generated by monitoring the new rates and compared against a predefined threshold, $\eta_s$. If $\Pr(Q > q) < \eta_s$, an alarm is raised. We vary $\eta_s$ over a range of values for ROC evaluation.

B. EVALUATION DATASETS

We wanted to use real, labeled and public background and attack datasets to measure the accuracy of the evaluated anomaly detectors. Real and labeled data allow realistic and repeatable quantification of an anomaly detector’s accuracy, which is a main objective of this work. Moreover, as defined in Section I-B, another objective is to evaluate the accuracy or scalability of the anomaly detectors under different normal and attack traffic rates and at different points of deployment in the network. This evaluation objective is somewhat unique to this effort, with [45] being the only other study that provides some insight into host versus edge deployments.

Different network deployment points are responsible for handling traffic from varying number of nodes. For instance, an endpoint requires to cater for only its own traffic, while
an edge router needs to monitor and analyze traffic from a variety of hosts in its subnet. In general, as one moves away from the endpoints towards the network core, the number of nodes, and consequently the traffic volume, that a network entity is responsible for increase considerably. We argue that if an algorithm that is designed to detect high- or low-rate attacks at a particular point of deployment, say an edge router, scales to and provides high accuracy at other traffic rates and deployment points, say at endpoints, then such an algorithm is quite valuable because it provides an off-the-shelf deployment option for different network entities. (We show later in this study that some existing algorithms are able to achieve this objective.)

To test the anomaly detectors for scalability, we use two real traffic datasets that have been independently-collected at different deployment points. The first dataset is collected at the edge router of the Lawrence Berkeley National Laboratory (LBNL), while the second dataset is collected at network endpoints by our research lab\(^4\). In this section, we describe the data collection setups and the attack and background traffic characteristics of the LBNL and the endpoint datasets.

1) The LBNL Dataset: This dataset was obtained from two international network locations at the Lawrence Berkeley National Laboratory (LBNL) in USA. Traffic in this dataset comprises packet-level incoming, outgoing and internally-routed traffic streams at the LBNL edge routers. Traffic was anonymized using the `tcpmkpub` tool; refer to [66] for details of anonymization.

LBNL Background Traffic: LBNL data used in this study is collected during three distinct time periods. Some pertinent statistics of the background traffic are given in Table I. The average remote session rate (i.e., sessions from distinct non-LBNL hosts) is approximately 4 sessions per second. The total TCP and UDP background traffic rate in packets per second is

\(^4\)We also wanted to use a traffic dataset collected at a backbone ISP network; such datasets have been used in some prior studies [25]–[27]. However, we could not find a publicly available ISP traffic dataset.
shown in column 5 of the table. A large variance can be observed in the background traffic rate at different dates. This variance will have an impact on the performance of volumetric anomaly detectors that rely on detecting bursts of normal and malicious traffic.

The main applications observed in internal and external traffic are Web (HTTP), Email and Name Services. Some other applications like Windows Services, Network File Services and Backup were being used by internal hosts; details of each service, information of each service’s packets and other relevant description are provided in [67].

LBNL Attack Traffic: Attack traffic was isolated by identifying scans in the aggregate traffic traces. Scans were identified by flagging those hosts which unsuccessfully probed more than 20 hosts, out of which 16 hosts were probed in ascending or descending order [66]. Malicious traffic mostly comprises failed incoming TCP SYN requests; i.e., TCP portscans targeted towards LBNL hosts. However, there are also some outgoing TCP scans in the dataset. Most of the UDP traffic observed in the data (incoming and outgoing) comprises successful connections; i.e., host replies are received for the UDP flows. Table I [column 6] shows the attack rate observed in the LBNL dataset. Clearly, the attack rate is significantly lower than the background traffic rate. Thus these attacks can be considered low rate relative to the background traffic rate. (We show later that background and attack traffic at endpoints exhibit the opposite characteristics.)

Since most of the anomaly detectors used in this study operate on TCP, UDP and/or IP packet features, to maintain fairness we filtered the background data to retain only TCP and
UDP traffic. Moreover, since most of the scanners were located outside the LBNL network, to remove any bias we filter out internally-routed traffic. After filtering the datasets, we merged all the background traffic data at different days and ports. Synchronized malicious data chunks were then inserted in the merged background traffic.

2) Endpoint Dataset: Since no publicly-available endpoint traffic set was available, we spent up to 14 months in collecting our own dataset on a diverse set of 13 endpoints. Complexity and privacy were two main reservations of the participants of the endpoint data collection study. To address these reservations, we developed a custom tool for endpoint data collection. This tool was a multi-threaded MS Windows application developed using the Winpcap API [68]. (Implementation of the tool is available at [44].) To reduce the packet logging complexity at the endpoints, we only logged some very elementary session-level information of TCP and UDP packets. Here a session corresponds to a bidirectional communication between two IP addresses; communication between the same IP address on different ports is considered part of the same network session. To ensure user privacy, the source IP address (which was fixed/static for a given host) is not logged, and each session entry is indexed by a one-way hash of the destination IP with the hostname. Most of the detectors evaluated in this work can operate with this level of data granularity.

Statistics of the two highest rate and the two lowest rate endpoints are listed in Table II.\(^5\)

\(^5\)The mean session rates in Table II are computed using time-windows containing one or more new sessions. Therefore, dividing total sessions by the duration does not yield the session rate of column 5.
As can be intuitively argued, the traffic rates observed at the endpoints are much lower than those at the LBNL router. In the endpoint context, we observed that home computers generate significantly higher traffic volumes than office and university computers because: 1) they are generally shared between multiple users, and 2) they run peer-to-peer and multimedia applications. The large traffic volumes of home computers are also evident from their high mean number of sessions per second. For this study, we use 6 weeks of endpoint traffic data for training and testing. Results for longer time periods were qualitatively similar.

To generate attack traffic, we infected VMs on the endpoints by the following malware: Zotob.G, Forbot-FU, Sdbot-AFR, Dloader-NY, SoBig.E@mm, MyDoom.A@mm, Blaster, Rbot-AQJ, and RBOT.CCC; details of the malware can be found at [69]. These malware have diverse scanning rates and attack ports/applications. Table III shows statistics of the highest and lowest scan rate worms; Dloader-NY has the highest scan rate of 46.84 scans per second (sps), while MyDoom-A has the lowest scan rate of 0.14 sps, respectively. For completeness, we also simulated three additional worms that are somewhat different from the ones described above, namely Witty, CodeRedv2 and a fictitious TCP worm with a fixed and unusual source port. Witty and CodeRedv2 were simulated using the scan rates, pseudocode and parameters given in research and commercial literature [69], [70].

Endpoint Background Traffic: The users of these endpoints included home users, research students, and technical/administrative staff. Some endpoints, in particular home computers, were shared among multiple users. The endpoints used in this study were running different types of applications, including peer-to-peer file sharing software, online multimedia applications, network games, SQL/SAS clients etc.

Endpoint Attack Traffic: The attack traffic logged at the endpoints mostly comprises outgoing portscans. Note that this is the opposite of the LBNL dataset, in which most of the
attack traffic is inbound. Moreover, the attack traffic rates (Table III) in the endpoint case are generally much higher than the background traffic rates (Table II). This characteristic is also the opposite of what was observed in the LBNL dataset. This diversity in attack direction and rates provides us a sound basis for performance comparison of the anomaly detectors evaluated in this study [17], [18].

For each malware, attack traffic of 15 minutes duration was inserted in the background traffic of each endpoint at a random time instance. This operation was repeated to insert 100 non-overlapping attacks of each worm inside each endpoint’s background traffic.
CHAPTER 4

IV. TAXONOMY OF NETWORK ANOMALY DETECTION SYSTEMS

Accurate taxonomies of intrusion detection systems have been proposed in past literature [39]–[42]. In this section, we combine and extend prior taxonomies to obtain a coherent classification framework designed specifically for Network Anomaly Detection Systems (NADSs). Moreover, unlike prior studies, our prime objective for the taxonomy is to gain insight into the effectiveness of the underlying traffic characteristics and detection principles used by different NADSs. To this end, we classify and evaluate prominent NADSs falling under different classes of the proposed taxonomy with an aim to identify which classes of the taxonomy perform better than others and why.

A. Taxonomy based on Learning Model and Detection Principles

By definition, an anomaly detection system forms a model of benign behavior and uses this model as a baseline to find anomalies in network traffic. Therefore, we propose a behavioral taxonomy of NADSs on a three-dimensional plane where: 1) The first dimension pertains to the learning model of the NADSs; 2) The second dimension refers to the learning behavior or the process by which these baseline models are created; 3) The third dimension classifies the detection principles employed to flag anomalies in real-time traffic. We now look at these dimensions in some detail; interested readers are referred to [40] and [41] for more information.

- **Learning model** is the learnt distribution of normal traffic semantics against which perturbations in test data are identified and significant variations are flagged as anomalous. This dimension is subdivided into the following classes:
  - **Normal Behavior** refers to the benign profiles used by the NADSs for training. For example, Packet Header based Anomaly Detector (PHAD) [4] learns the benign or
typical intervals of packet header values during training. Any deviation in the test data that is not consistent with the learnt training intervals is flagged as anomalous.

– **No initial knowledge base** implies a change detection algorithm that does not require training on benign data. An example of this class is the Threshold Random Walk (TRW) detector [4] which computes a likelihood ratio by evaluating failed incoming TCP connections and raises an alarm when this ratio exceeds a programmed threshold.

• **Learning Behavior** refers to the process by which the model of benign behavior is obtained.
  
  – **Self Learning** NADSs learn by example. For instance, the Kalman filter based detector of [4] learns the initial network states from the benign data which is then used to predict the future state in the test data and identify anomalies.
  
  – **Pre-Programmed** NADSs include an initial notion of normality built into the detector. For example, the Deviation Score based Wavelet Analysis NADS [4] uses a preprogrammed deviation score of more than 2.0 to flag anomalies in test data.

• **Detection Principles** refer to the techniques employed by an NADS to identify anomalies.
  
  – **Descriptive Statistics** NADSs collect statistics of various parameters of the system and then employs a distance vector approach to identify the variations between the learnt profiles and the test data.
  
  – **Rule Modeling** NADSs learn and formulate a number of rules pertaining to the normal operation of the system. An alarm is raised if the traffic observed during actual operation matches poorly with the rule base.
  
  – **Simple Statistics** NADSs employ simple statistical parameters (e.g., failed connec-
tion attempts, unsuccessful logins etc.) for anomaly identification.

- **Threshold NADSs** apply programmed thresholds on the collected statistics for alarm generation.

Fig. 4 outlines the proposed taxonomy based on learning and detection principles. The height of the bars represent the detection principle employed by the NADSs for anomaly identification. In the figure, some prominent NADSs are classified using the proposed taxonomy. (Percentage values following the names of some anomaly detector are the accuracies of these detectors and detailed explanation of these values are deferred to subsequent sections.) Details of the NADSs classified in Fig. 4 can be found in the references of [4]. The green bars in Fig. 4 represent those detectors that fulfil all criteria of a true NADS as they are self-learning and rely completely on normal traffic behavior for anomaly detection.

Existing NADSs clearly form four distinct clusters in this taxonomy. The first cluster comprises true anomaly detectors that develop a model of normal traffic behavior and automatically adapt to (or self learn) changes in normal behavior; e.g., PHAD [4] trains on the benign profiles and learns the intervals for the allowed values of the packet header fields. The second cluster contains NADSs that are self learning but do not require an initial knowledge base for training; e.g., the Sketch based Anomaly Detector [4] that forms a sketch of the entire network to learn the communication patterns between hosts and then the learnt behavior is used as a baseline to identify anomalies in the next time window. The third cluster of the NADSs include detectors that are preprogrammed and require no initial data for training; e.g., the TRW detector [4] which is designed to detect failed incoming TCP connections. The fourth cluster lies at the intersection of preprogrammed and self learning detectors; e.g., the Subspace Method [4] which employs a statistical measure to automatically determine the threshold values for alarm generation but does not train on benign data. Despite forming
clusters, note that existing detectors span all classes of the detection principles and learning behavior.

B. Taxonomy based on Traffic Characteristics

In addition to a behavioral taxonomy, to derive useful lessons from an NADS classification we need to categorize existing NADSs with respect to their traffic characteristics. Network traffic can be analyzed at different levels of aggregation and detail. For instance, traffic can be examined as flows between connected hosts or as individual packets, bytes or bits exchanged between the hosts. This traffic may be analyzed for anomalies at endpoints, enterprise routers,
ISPs or Internet backbones. Semantic details like protocol header information or active network services may also be analyzed for anomalies. To study the impact of traffic aggregation, volume and semantics on network anomaly detector’s accuracy, in this section we classify the NADSs discussed in the last section with respect to their underlying traffic characteristics and network deployment points. Using the taxonomy of [42], we now propose another three-dimensional NADS classification framework based on traffic scale and semantics.

- **Traffic Semantics:** In view of the different level of traffic features used by different anomaly detectors, the first dimension of the present taxonomy subdivides NADSs based on the following analysis models:
  
  - **Flow Analysis** is the study of various attributes of traffic flows. These can either be the number of packets that are transferred between hosts, the number of connections initiated, etc.
  
  - **Protocol Analysis** is the study of network protocol information. Different NADSs analyze different types (MAC, network, transport or application layer) of protocol information in network packets.
  
  - **Services** can be accessed remotely or locally. In either case, the traffic generated can be anomalous if a host inside the network has been compromised. Thus the traffic originating within the network as a result of these services also need to be scrutinized.

- **Traffic Scale:** Traffic semantics can be viewed along varying scales. In network traffic analysis, distributions of traffic features along varying aggregation scales are completely different. Thus the scale of traffic analysis needs to be defined before deductions about classification accuracy are made.
  
  - **Micro-level Analysis** pertains to the study of the distribution of low level feature. It
is the analysis either at the level of individual bytes or packets or in a time window of < 1 sec.

- **Meso-level Analysis** is the analysis of connection initiations or on a scale with time windows in seconds to mins, or traffic destined to a specific host etc.

- **Macro-level Analysis** is the study of the distributions of high level features. For example, the traffic across a network or analysis of traffic in large time bins.

- **Deployment Point**: NADSs are generally designed for a particular point of deployment in the network. For instance, some network-based anomaly detection systems have been
proposed for ISP traffic (e.g. Kalman Filter [4]), while others are designed for router (e.g. TRW based detector [4]) and endpoint (e.g. Rate Limiting [4]) deployments. Recently, a few NADSs have also been proposed for the Internet backbone transit networks (e.g. Subspace method [4]). Since the traffic volume and characteristics change considerably at each of these deployment points, we further classify existing ADSs in accordance with their deployment scenarios:

– **Backbone Traffic Analysis** can be used to identify anomalies before the traffic reaches the stub networks. However, the volume of data to be analyzed at this deployment point is large and therefore anomaly detection is difficult.

– **ISP Traffic Analysis** can again attempt to throttle malicious traffic before it reaches the customers’ stub networks. NADSs for ISP gateways have the same limitation as the backbone routers.

– **Enterprise Gateway Traffic Analysis** is for NADSs that analyze the complete network traffic destined to or originating from the hosts within the network. These detection systems run on the network routers that connect the local network to the ISP network.

– **Endpoint Traffic Analysis** can be used to detect and contain malicious traffic at the endhosts. Since the traffic rates are much lower than the traffic analyzed by NADSs running on routers, these detectors can provide potentially higher accuracy than Backbone, ISP and Gateway NADSs.

Fig. 5 shows the classification of existing NADSs with respect to the present taxonomy. It can be seen that the true anomaly detectors fall in four of the twelve categories of the taxonomy and are typically designed for deployment on network gateways. Moreover, majority of the network based anomaly detection systems use flow- or protocol-level meso-
scale traffic semantic information for detection. We note that detection methods like NIDES, Kalman Filter and Rate Limiting use micro-scaled flow level information. TRW and TRW-CB use the TCP connection information and thus constitute protocol-level meso-scaled analysis category. Similarly, PHAD makes use of the protocol header information for identifying anomalies. Subspace method forms a model of the OD-level flows, and therefore falls in the macro-scaled flow-level category.

Many of the detectors classified in this taxonomy span across multiple classes. In order to represent such dependence across classes, two bars with the same number has been shown in Fig. 5. For instance, DIDS [4] uses protocol- as well as service-level information for threshold based anomaly detection. Similarly, PHAD and Maximum Entropy have the capacity to operate at varying levels of traffic aggregation; i.e., these NADSs work on individual packets as well as packets destined to specific hosts and in varying time windows. Moreover, the Deviation Score based Wavelet Analysis works on bytes as well as flows in specific windows.
CHAPTER 5

V. PERFORMANCE EVALUATION

In this section, we evaluate the accuracy, scalability and delay of the anomaly detectors described in the last section on the endpoint and router datasets.

A. Accuracy and Scalability Comparison

In this section, we present ROC analysis on the endpoint dataset. The following section explains the scalability experiments in which ROC analysis is performed on the LBNL dataset and the results are compared with the endpoint experiments.

1) Averaged ROCs for the Endpoint Dataset: Fig. 6 provides the averaged ROC analysis of the anomaly detection schemes under consideration. Clearly, the Maximum Entropy detector provides the highest accuracy by achieving near 100% detection rate at a very low false alarm rate of approximately 5 alarms/day. The Maximum Entropy detector is followed closely by the credit-based TRW approach. TRW-CB achieves nearly 90% detection rate at a reasonable false alarm rate of approximately 5 alarms/day. The original TRW algorithm, however, provides very low detection rates for the endpoint dataset. Results of these three schemes are shown more clearly in Fig. 7(a). Based on these results, the Maximum Entropy algorithm provides the best accuracy on endpoints, while TRW provides the best detection on LBNL dataset.

The Kalman Filter approach is also quite accurate as it provides up to 85% detection rates at a reasonably low false alarm cost. Rate Limiting, although designed to detect outgoing scanning attacks, provides very poor performance. This result substantiates the results of [45] where very high false positive rates for high detection rates were reported for classical rate limiting. Hence, we also deduce that rate limiting is ineffective for portscan detection at endpoints.
PHAD does not perform well on the endpoint data set. The detection is accompanied with very high false alarm rates. NIDES achieve reasonable detection rates at very low false alarm rates, but is unable to substantially improve its detection rates afterwards. PHAD relies on previously seen values in the training dataset for anomaly detection. Therefore, if a scanner attacks a commonly-used port/IP then PHAD is unable to detect it. On similar grounds, if the malicious traffic is not bursty enough as compared to background traffic then NIDES will not detect it, irrespective of how much the detection threshold is tuned.

Due to the thresholding difficulties for the subspace method explained in Section II-C, in Fig. 8 we report results for this technique for varying values of selected principal components. The highest detection rate of 22% is observed at $k = 2$ principal components. This already low detection rate decreases further at $k = 5$ and drops to 0% at $k = 15$. False alarm rates
show the opposite trend. Thus the subspace method fails to give acceptable accuracy on the endpoint dataset.

The ROC results for the endpoint dataset are somewhat surprising because two of the top three detectors are general-purpose anomaly detectors (Maximum Entropy and Kalman Filter), but still outperform other detectors designed specifically for portscan detection, such as the TRW and the Rate Limiting detectors. We, however, note that this analysis is not entirely fair to the TRW algorithm because TRW was designed to detect incoming portscans, whereas our endpoint attack traffic contains mostly outgoing scan packets. The credit-based
variant of TRW achieves high accuracy because it leverages outgoing scans for portscan detection. Thus TRW-CB combines the complementary strengths of rate limiting and TRW to provide a practical and accurate portscan detector for endpoints. This result agrees with earlier results in [45].
2) ROCs for Low- and High-Rate Endpoint Attacks:: To evaluate the scalability of the ADSs under high- and low-rate attack scenarios, Fig. 9 plots the ROCs for the highest rate (Dloader-NY) and lowest rate (MyDoom-A) attacks in the endpoint dataset. It can be observed that for the high-rate attack [Fig. 9(a)] Maximum Entropy, TRW, TRW-CB and Kalman Filter techniques provide excellent accuracy by achieving 100% or near-100% detection rates with few false alarms. NIDES’ performance also improves as it achieves approximately 90% detection rate at very low false alarm rates. This is because the high-rate attack packets form bursts of malicious traffic that NIDES is tuned to detect. Rate Limiting and PHAD do not perform well even under high attack rate scenarios.

Fig. 9(b) shows that the accuracies of all detectors except PHAD and Maximum Entropy degrade under a low-rate attack scenario. Maximum Entropy achieves 100% detection rate with false alarm rate of 4-5 alarms/day. TRW-CB recovers quickly and achieves a near-100% detection rate for a daily false alarm rate around 10 alarms/day. NIDES, however, shows the biggest degradation in accuracy as its detection rate drops by approximately 90%. This is because low-rate attack traffic when mixed with normal traffic does not result in long attack bursts. TRW’s accuracy is also affected significantly as its detection rate drops by about 35% as compared to the high-rate attack. PHAD does not rely on traffic rate for detection, and hence its accuracy is only dependent on the header values observed during training.

3) Averaged ROCs for the LBNL Dataset:: Fig. 10 shows the ROCs for the LBNL dataset. Comparison with Fig. 7 (a) and (b) reveals that the Maximum Entropy detector is unable to maintain its high accuracy on the LBNL dataset; i.e., the Maximum Entropy algorithm cannot scale to different points of network deployment. TRW’s performance improves significantly as it provides a 100% detection rate at a negligible false alarm cost. TRW-CB, on the other hand, achieves a detection rate of approximately 70%. Thus contrary to the endpoint dataset,
Fig. 9. ROC curves for the lowest and highest rate attack in the endpoint dataset; results averaged over 12 endpoints with 100 instances of each attack.

the original TRW algorithm easily outperforms the TRW-CB algorithm on LBNL traces. As explained in Section III-B1, the LBNL attack traffic mostly comprises failed incoming TCP connection requests. TRW’s forward sequential hypothesis based portscan detection
algorithm is designed to detect such failed incoming connections, and therefore it provides high detection rates. Thus on an edge router, TRW represents a viable deployment option. Kalman Filter detector’s accuracy drops as it is unable to achieve a detection rate above 60%. PHAD provides very high detection rates, albeit at an unacceptable false alarm rate. Other detectors’ results are similar to the endpoint case. (Results for the subspace method were similar to those reported earlier and are skipped for brevity.) It can be observed from Fig. 10 that all algorithms except TRW fail to achieve 100% detection rates on the LBNL dataset. This is because these algorithms inherently rely on the high burstiness and volumes of attack traffic. In the LBNL dataset, the attack traffic rate is much lower than the background traffic rate. Consequently, the attack traffic is distributed across multiple time windows, with each window containing very few attack packets. Such low density of attack traffic in the evaluated time-windows remains undetected regardless of how much the detection thresholds
Table IV provides the detection delay for each anomaly detector. On the endpoint dataset, delay is reported for the highest and the lowest rate attacks, while on the LBNL dataset this delay is computed for the first attack that is detected by an anomaly detector. A delay value of $\infty$ is listed if an attack is not detected altogether. It can be observed that detection delay is reasonable (less than 1 second) for all the anomaly detectors except the Maximum Entropy detector which incurs very high detection delays. High delays are observed for the Maximum Entropy detector because it waits for perturbations in multiple time windows before raising an alarm. Among other viable alternatives, TRW-CB provides the lowest detection delays for all three experiments. Detection delay for the TRW is also reasonably low.
CHAPTER 6

VI. LESSONS LEARNT

This section provides an outline of the objectives of the study and the deductions pertaining to these objectives. Moreover, promising portscan detection guidelines are proposed based on the taxonomy and the NADS evaluation along with some experimental results, in the form of ROC curves, for the accuracy improvements realized due to the proposed guidelines.

A. Objectives of the Study

In this study, we taxonomized NADSs proposed in the last few years. Moreover, we also evaluated eight prominent network-based anomaly detectors using two portscan traffic datasets having complementary characteristics. These detectors were evaluated on accuracy, scalability and delay criteria. Based on the results of this research study, we now rephrase and summarize our deductions pertaining to the main objectives of this study:

- Which class of the taxonomy perform better than others? On the endpoint dataset, our evaluation shows that the protocol level – micro and meso scaled green class of true anomaly detectors provides the best accuracy. However, on the router based LBNL dataset, we observe that the protocol level – meso scaled red class of preprogrammed NADSs outperform the green class. Thus the protocol level – micro and meso scaled Maximum Entropy detector is unable to maintain its high accuracy on the LBNL dataset.

- Which algorithms provide the best accuracy under varying rates of attack and normal traffic and at different points of deployment? Under the varying attack and background traffic rates observed in the two datasets, a general-purpose Maximum Entropy Detector [30] and variants of the Threshold Random Walk (TRW) algorithm [17], [18] provided the best overall performance under most evaluation criteria. In this context, TRW is
suitable for deployment at routers, while TRW-CB and Maximum Entropy are suitable for deployment at endpoints.

- What are the promising traffic features and theoretical frameworks for portscan anomaly detection? The Maximum Entropy and TRW detectors use statistical distributions of failed connections, ports and IP addresses. Furthermore, based on the results of the Maximum Entropy detector on endpoints, a histogram-based detection approach, in which baseline frequency profiles of a set of features is compared with real-time feature frequencies, appears very promising.

- What detection delays are incurred by the anomaly detectors? If an attack is detected, detection delay is less than 1 second for all anomaly detectors, except the Maximum Entropy Estimation method which incurs very large delays.

- What are promising portscan detection guidelines that build on the strengths and avoid the weaknesses of the evaluated anomaly detectors? From the high detection rates of the Maximum Entropy and PHAD detectors, it appears that using a higher dimensional feature space facilitates detection, without compromising complexity. On the other hand, relying on specific traffic features (e.g., rate, connection failures, etc.) can degrade accuracy as the attack and background traffic characteristics change. In summary, a number of statistical features used in an intelligent histogram-based classification framework appear promising for portscan anomaly detection.

B. Why do Some Classes of the Taxonomy Perform Better than Others?

Based on the accuracy results, protocol level – micro and meso scaled green class provides the best accuracy on endpoints, while the protocol level – meso scaled red class provides the best detection on the LBNL dataset. Both the classes are based on micro or meso scaled protocol level information. Since high-rate malicious packets affect an increase in the frequency
of certain packet headers (e.g., the ports being attacked in case of a worm, or the IP being attacked in case of a DoS attack), attacks can be detected most accurately if the protocol level information is analyzed at different aggregation levels.

At the endpoints, the volume of traffic is much lesser than that at the router. Moreover, the endpoint behavior changes with time as different applications are used by the endhost. Thus NADSs that are used for deployment at endpoints need to train on benign data to cater for the changing user behaviors. That is why the green class of self learning NADSs that trained on benign profiles provide the best accuracy on the endpoint dataset.

A router-based NADS analyzes large quantities of data and consequently the attack traffic gets averaged out in the normal traffic. Thus training on benign profiles would not render higher detection rates, as can be seen by the results of NADSs on LBNL dataset. However, intelligent statistical measures (e.g., the likelihood ratio test used by TRW) improve the NADS’ accuracy. Thus to provide better accuracy on enterprise/network level traffic, selection of the right detection method/principle is more important to achieve higher detection rates than training on benign data.

In light of the accuracy evaluation results, Maximum Entropy provides best detection and false alarm rates on individual basis because of the following inbuilt characteristics:

- It segregates traffic into multiple packet classes;
- Analyzes a high dimensional feature space;
- Generates an alarm when anomalies span across multiple time windows.

PHAD detector operates on similar principles and thus also provides high detection rates. In all datasets we observe that traffic rates keep changing. While all NADSs apply fixed thresholds to classify anomalies in real-time traffic, an accurate NADS should vary its classification thresholds with respect to the changing patterns in benign traffic.
C. Promising Guidelines to Improve the Accuracy of Existing and Future NADSs

Based on above discussion, we propose the following guidelines to improve the accuracy of NADSs:

**Guideline 1:** To provide high detection rates, endpoint based NADSs should be self learning systems which train on benign traffic profiles.

**Guideline 2:** To provide high detection rates, router based NADSs should employ intelligent detection principles for identifying anomalies in network traffic.

**Guideline 3:** NADSs should operate on meso-scaled protocol-level packet features.

**Guideline 4:** To reduce the false alarm rates, NADSs should raise an alarm only when they encounter anomalies spanning across multiple time windows.

**Guideline 5:** To improve the detection rates, NADSs should simultaneously consider multiple packet header fields, e.g. TCP SYN, Ports, Protocol etc.

**Guideline 6:** To improve detection rates, NADSs should segregate traffic into multiple packet classes before anomaly detection.

**Guideline 7:** Adaptive thresholding should be introduced to allow the NADSs to dynamically adjust their detection thresholds in accordance with the changing normal traffic characteristics.

Guidelines 4–7 aim at improving the accuracy of the anomaly detection system as well as reduce human intervention in their operation. Following is a detailed description of these guidelines and the accuracy improvements achieved:

1) *Multi-Window Classification (Guideline 4):* We have seen in the comparative evaluation study of the NADSs, that most NADSs suffer from high false alarm rates. The problem mainly
Fig. 11. An example of Multi-window classification: Maximum-Entropy Detector’s output on five LBNL time windows containing benign and malicious traffic.

stems from the fact that most NADSs raise an alarm as soon as the first anomalous time window is identified. We observed that, due to an inherent burstiness present in attack traffic, anomalies tend to sustain across multiple time windows. An example of this behavior is shown in Fig. 11. In Fig. 11(a), even if a false alarm is generated for a benign traffic window, the false alarm does not span multiple windows. On the other hand, anomalous activity tends
to occur in bursts, and therefore multiple successive windows are flagged as anomalous. This difference between NADS classification on malicious and benign traffic can be leveraged to reduce an NADS’ false alarm rate. Specifically, an NADS can reduce its false alarms if it raises an alarm only after sufficient number of anomalous time windows have been observed in a given time period. We call this simple existing technique *Multi-Window Classification*.

For accurate multi-window classification, we consider a fixed number of $w$ most recent classifications by an NADS. In the $w$ classifications, a majority vote is taken to classify the current time window as benign or anomalous. It should be highlighted that multi-window classification will introduce detection delays in the NADSs. However, as already shown, detection delays of most existing NADSs are extremely low and hence these NADSs can tolerate slightly longer detection delays to achieve higher accuracies.

2) *Feature Space Extension (Guideline 5):* We have seen in the comparative performance evaluation that Maximum Entropy and PHAD are the highest accuracy detectors. Both these detectors employ a rich feature space for detection. Thus greater the number of packet fields analyzed for anomaly detection, higher the probability of finding the anomaly. Thus, if within one time window, instead of analyzing a few packet header fields, the maximum available fields are analyzed, its highly probable that the NADS finds an anomaly that perturbs any of the observed packet feature.

Figure 12 shows the distribution of the packet score calculated for each packet based on the packet header fields analyzed. In Figure 12(a) PHAD detector computes the packet score based on a single anomalous packet header field. Figure 12(b) shows the packet score distribution, for PHAD, when multiple packet header fields are simultaneously used for packet score calculation. In Figure 12(a), since packet score does not exceed the specified threshold value, PHAD detector fails to detect the anomalies which are otherwise detected
if diverse packet features are analyzed as shown in Figure 12(b). Thus, using a rich feature space assists the detection of anomalies that perturb any network traffic feature resulting in high detection rates for the NADSs.

3) Traffic Splitting (Guideline 6): Traffic Splitting is also aimed at improving an NADS’s detection rate.
Our preliminary investigation revealed that much of the malicious traffic is not detected because of an averaging-out effect introduced by relatively large volumes of benign background traffic. More specifically, if the attack traffic rate is comparable to or less than the background traffic rate then background traffic acts like noise during anomaly detection and allows malicious traffic to bypass an NADS.

As an example, note that aggregate traffic routed towards/from a network is a composite of multiple traffic types, such as TCP, UDP, ARP, ICMP traffic etc. Now consider the Witty worm which was a very high-rate UDP-based worm. Since TCP comprises of almost 80% of the traffic seen on the Internet, if an NADS analyzes aggregate network traffic then the attack traffic is overwhelmed by the majority TCP traffic. Such traffic averaging degrades the detection rate of an NADS. Note that in this example traffic other than UDP acts as noise and, depending upon whether the volume of background traffic is substantial, would either delay or prevent the detection of the anomaly. To counter this problem, we propose to perform anomaly detection separately on different types of network traffic. Hence, we use traffic semantics to segregate a single traffic stream into multiple traffic substreams before anomaly detection is performed. Such traffic splitting will inherently allow the background traffic to be segregated from the attack traffic, thereby facilitating the anomaly detection phase. After traffic splitting, separate instances of an NADS operate on different substreams in parallel. The outputs of these NADS instances are combined to detect anomalies. Based on the example given above, traffic splitting should, in addition to improving detection rates, reduce detection delays.

As a proof-of-concept, in Fig. 13 shows a comparison of aggregate traffic with segregated TCP and UDP traffics. Both of these anomalous windows were observed and analyzed in the Threshold Random Walk Credit Based (TRW-CB) algorithm under RBOT.CCC’s and
Witty’s malicious traffic. TRW-CB calculates the likelihood ratio for detecting anomalies. It is clear from the Fig. 13 that when aggregate traffic is analyzed without segregation, the output of the likelihood ratio test does not cross the fixed TRW-CB threshold and the malicious traffic remains undetected for both examples. However, when traffic splitting is employed and TCP and UDP traffic is analyzed separately, the threshold is exceeded many
times in the 200 second windows shown in Fig. 13(a) and (b). Hence traffic splitting removes the noisy background traffic from malicious traffic and subsequently increases the detection rate of an NADS.

D. NADS Accuracy Improvements by Traffic Splitting and Multi-Window Classification

We now jointly apply the two accuracy improvement techniques of traffic splitting and multi-window classification on the TRW and the TRW-CB detectors to observe the accuracy improvements that can be achieved. TRW and TRW-CB are evaluated to present the worst case scenario as both these detectors already incorporate some notion of traffic splitting. For traffic splitting, we segregate traffic into four classes: 1) UDP, 2) TCP, 3) broadcast, and 4) all other traffic types. For multi-window classification, we use a window size of $w = 5$; i.e., a majority vote of the last 5 NADS classifications is used to flag the current window as normal or anomalous.

It can be seen in Fig. 14(a) and (b) that the proposed techniques provide consistent accuracy improvements for TRW and TRW-CB on both datasets. On the LBNL dataset, while the detection rates are similar to the original algorithms, substantial reductions are observed in the false alarm rates. On the endpoint dataset, the proposed techniques effect significant improvements in both detection as well as false alarms rates as shown in Fig. 14(a).

E. NADS Accuracy Improvements by Feature Space Extension and Multi-Window Classification

We evaluate two NADSs, namely the Maximum Entropy and PHAD detectors. These detectors already follow Guideline 5 because they operate on a high dimensional feature space. Furthermore, the Maximum Entropy detector already has the notion of time extension (Guideline 4) is built into it. More specifically, the Maximum Entropy detector segregates traffic into multiple protocol classes before the anomaly detection step and then raises an
alarm only after an anomaly is observed in multiple time windows. Such a strategy results in very low false alarm rates, but compromises the detection rate. To make this algorithm compliant with Guideline 5, instead of waiting for multiple windows before making a decision, we modified the Maximum Entropy method to raise an alarm if the divergence of a given number of protocol classes in a time-window exceed a threshold.
Similarly, a time extension is introduced into PHAD to reduce its false alarm rate. Specifically, to make PHAD complaint with Guideline 4, an alarm is raised only when an anomaly is observed in multiple time windows/packets.

Figs. 15 (a) and (b) show a comparative analysis of the original and the Improved variants of PHAD and Maximum Entropy. It can be seen that evaluating PHAD across multiple
time windows using a high dimensional feature space clearly improves the accuracy of the detector. Similarly, evaluating Maximum Entropy across its feature space instead of its original time domain design considerably improves the accuracy of the detector.

Evaluating NADSs in space and across multiple time windows might have an impact on the detection delay of the detectors. So we also perform delay comparison so as to observe the extent of the delay that guideline 4 can incur. Table V provides the detection delay for Maximum Entropy detector and PHAD. It can be observed that the detection delay for the Improved variant of Maximum Entropy detector is dramatically lower than the original algorithm. This is because the Improved variant of the Maximum Entropy detector does not wait for multiple anomalous windows before raising an alarm. For PHAD, the detection delay remains unaltered because the Improved variants simultaneously operates in space and time.

In the following section, we highlight the accuracy improvements that can be realized by jointly applying all these guidelines.

### F. NADS Accuracy Improvements by Jointly Using the Proposed Guidelines

We now argue that these described guidelines are complementary to each other and can be applied simultaneously to achieve higher accuracy while limiting the need for human intervention during NADS operation.

Fig. 16 shows a block diagram of an NADS that jointly employs the proposed guidelines. The first thing we note is that none of the guidelines require modifications to the NADS.

<table>
<thead>
<tr>
<th></th>
<th>ST-Max Entropy</th>
<th>ST-PHAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>MyDoom (msec)</td>
<td>157</td>
<td>900</td>
</tr>
<tr>
<td>Dloader-NY (msec)</td>
<td>100</td>
<td>990</td>
</tr>
<tr>
<td>LBNL (msec)</td>
<td>333</td>
<td>330</td>
</tr>
</tbody>
</table>
The Traffic Splitter is working as a pre-NADS phase which is segregating a single stream of traffic into multiple packet classes; these classes can be formed on any basis. The second phase, as shown in Fig 16 is the feature space extension. Once traffic is segregated into multiple packet classes, each packet class is further segregated into multiple packet features to be examined for an anomaly. Each packet feature class is sent to a separate NADS instance which uses the threshold provided by the Adaptive Thresholding Module to classify traffic in the observed window as benign or anomalous. Outputs from multiple instances of the NADS, each analyzing a unique packet feature class, are combined into a single result and handed over to the Multi-Window Classifier. The Multi-Window Classifier acting as a post-NADS phase takes the majority vote of prior classification results to decide whether or not an alarm should be raised.

Fig. 17 shows the accuracy of five prominent NADSs along with the jointly-improved
versions of these detectors after application of the proposed guidelines; all parameters are the same as described in previous sections. Due to a lack of threshold tuning capability, adaptive thresholding only results in a single point on the ROC plane. Fig. 17(a) shows...
the accuracy comparison for the endpoint dataset. Maximum Entropy improves slightly on its already accurate performance. Jointly-improved Kalman Filter detector provides better accuracy than the original algorithm with a detection rates of approximately 96%. TRW-CB detector maintains similar accuracy as before, but with the additional advantage of ADS automation. PHAD and TRW detectors show dramatic improvements in accuracies as they achieve detection rate improvements of approximately 45% and 70%, respectively, without compromising their low false alarms rates. Note that although there was a slight degradation in the jointly-improved TRW’s accuracy on the LBNL dataset, on the endpoint dataset the proposed techniques provide remarkable accuracy improvements for TRW. Thus the proposed guidelines, in addition to the benefits enumerated above, allow an NADS to scale to different points of deployment in the network.

From Fig. 17(b), marked and mostly consistent improvements in all the NADSs’ accuracies can be observed on the LBNL dataset. The Maximum-Entropy detector achieves a remarkable 100% detection rate at a reasonable false rate. Kalman Filter based detector’s accuracy also improves drastically as its detection rate increases from 54% to 80% with few false alarms. A similar accuracy improvement trend is observed for the PHAD detector. No improvement in detection/false alarm rate is observed in the TRW-CB detector; however, it continues to provide the same detection rates without any human intervention and at an acceptable false alarm rate. The TRW detector is the only exception on the LBNL dataset as it incurs somewhat higher false alarms after using the proposed guidelines.
CHAPTER 7

VII. CONCLUSION

The main aim of this research work was to develop a better understanding of the basic building blocks of Network-based Anomaly Detection Systems and to identify the features that distinguish one NADS from the other. We initially propose two multidimensional taxonomies of Network-based Anomaly Detection Systems with an aim to obtain insights into why some NADSs perform better than others. Our first taxonomy classifies NADSs based on their learning behavior and detection principles. The second taxonomy categorizes NADSs using traffic scales and semantics.

Moreover, we evaluated and compared eight prominent network based anomaly detectors on two independently collected public portscan datasets. These NADSs employed different traffic features and diverse theoretical frameworks for anomaly detection and have been used frequently for performance benchmarking in Intrusion Detection research literature.

NADSs were evaluated on three criteria: accuracy, scalability, and detection delay. Accuracy was evaluated by comparing ROC (false alarms per day versus detection rate) characteristics of the NADSs. Scalability was evaluated with respect to different background and attack traffic rates. Since the two datasets used in this study were collected at different network entities and contained attacks with different characteristics, evaluation over these datasets allowed us to compare the scalability of the proposed NADSs under varying traffic volumes. Detection delay was evaluated separately for high- and low-rate attacks.

Based on our findings, we proposed a few promising portscan detection guidelines to improve the accuracy and scalability of existing and future NADSs. Our experimental results showed that the proposed guidelines resulted in an average detection rate increase of 5% to 10%, while reducing the false alarm rates up to 50%. Thus, the proposed guidelines, while
reducing human intervention, provided drastic and consistent improvements in NADS’s accuracies.
REFERENCES


Maryland, 17.2 October 1988. NIST.


