REAL-TIME ANALYSIS OF AGGREGATE NETWORK TRAFFIC FOR ANOMALY DETECTION

A Dissertation

by

SEONG SOO KIM

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2005

Major Subject: Computer Engineering

REAL-TIME ANALYSIS OF AGGREGATE NETWORK TRAFFIC FOR ANOMALY DETECTION

A Dissertation

by

SEONG SOO KIM

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

A. L. Narasimha Reddy
(Chair of Committee)

Riccardo Bettati
(Member)

Don R. Halverson
(Member)

Michael S. Pilant
(Member)

Chanan Singh
(Head of Department)

May 2005

Major Subject: Computer Engineering

ABSTRACT

Real-time Analysis of Aggregate Network Traffic
for Anomaly Detection. (May 2005)

Seong Soo Kim, B.S., Yonsei University;
M.S., Yonsei University
Chair of Advisory Committee: Dr. A. L. Narasimha Reddy

The frequent and large-scale network attacks have led to an increased need for developing techniques for analyzing network traffic. If efficient analysis tools were available, it could become possible to detect the attacks, anomalies and to appropriately take action to contain the attacks before they have had time to propagate across the network.

In this dissertation, we suggest a technique for traffic anomaly detection based on analyzing the correlation of destination IP addresses and distribution of image-based signal in postmortem and real-time, by passively monitoring packet headers of traffic. This address correlation data are transformed using discrete wavelet transform for effective detection of anomalies through statistical analysis. Results from trace-driven evaluation suggest that the proposed approach could provide an effective means of detecting anomalies close to the source. We present a multidimensional indicator using the correlation of port numbers as a means of detecting anomalies.

We also present a network measurement approach that can simultaneously detect, identify and visualize attacks and anomalous traffic in real-time. We propose to represent samples of network packet header data as frames or images. With such a formulation, a series of samples can be seen as a sequence of frames or video. This enables techniques from image processing and video compression such as DCT to be applied to the packet header data to reveal interesting properties of traffic. We show that “scene change analysis” can reveal sudden changes in traffic behavior or anomalies. We show that “motion prediction” techniques can be employed to understand the patterns of some of the attacks. We show that it may be feasible to represent multiple pieces of data as different colors of an image enabling a uniform treatment of multidimensional packet header data.

Measurement-based techniques for analyzing network traffic treat traffic volume and traffic header data as signals or images in order to make the analysis feasible. In this dissertation, we propose an approach based on the classical Neyman-Pearson Test employed in signal detection theory to evaluate these different strategies. We use both of analytical models and trace-driven experiments for comparing the performance of different strategies. Our evaluations on real traces reveal differences in the effectiveness of different traffic header data as potential signals for traffic analysis in terms of their detection rates and false alarm rates. Our results show that address distributions and number of flows are better signals than traffic volume for anomaly detection. Our results also show that sometimes statistical techniques can be more effective than the NP-test when the attack patterns change over time.

DEDICATION

This dissertation is gratefully dedicated to

My Wife, Eun Kyung Jun

My Children, Minjeong and Taewoo

and My Parents, Jung Nam Kim and Geum Ja Yun

I could not have completed my study without their love, encouragement, and patience.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Dr. Reddy for his advice and support throughout this research. I am greatly indebted to my advisor for the invaluable guidance and encouragement which he has provided me through my study. He is always open to new ideas and I really appreciate the freedom he gave me while working on my research. This research could not have been possible without his brillant ideas. I owe much for his unending help to do the research and for his finacial support during my graduate study.

I would like to express my sincere appreciation to Dr. Halverson who was willing to provide me the valuable statistical advice for this study. I thank the committee members, Dr. Bettati and Dr. Pilant for serving as committee members and for helping me through my final steps. Special thanks are given to Dr. Vannucci for her guidance in the field of Wavelet Analysis.

Most of all, I am grateful to my wife, Eun Kyung Jun, for her encouragement and patience over the past three and half years. I also thank my children, Minjeong and Taewoo, for giving me smiles and the will of life.

This research was supported in part by the National Science Foundation, Texas Higher Education Board, Texas Information Technology and Telecommunications Taskforce and Intel Corp.

TABLE OF CONTENTS

Page

ABSTRACT iii

DEDICATION v

ACKNOWLEDGMENTS vi

TABLE OF CONTENTS vii

LIST OF FIGURES xii

LIST OF TABLES xvi

CHAPTER

I INTRODUCTION 1

- A. Motivation 1
- B. Previous Research 5
- C. Contributions and Outline 8

II AGGREGATED TRAFFIC SYSTEM 10

- A. Traffic Analysis at the Source 10
- B. General Mechanism of the Detector 13
- C. Approach 14
  - 1. Simulated Attacks 16
- D. Data Structure for Computing Signal 18

III ADDRESS CORRELATION SIGNAL-BASED ANOMALY DETCTION 24

- A. Discrete Wavelet Transform (DWT) 24
  - 1. Data Transform 24
  - 2. Discrete Wavelet Transform 25

CHAPTER Page

        3. Timescale Selection 27
    B. Statistical Analysis 27
        1. Detection Mechanism 27
        2. Selective Reconstruction in DWT.. 29
        3. Thresholds Setting Through Statistical Analysis 30
            a. Statistical Consideration of Threshold: Wide-sense Stationary 32
    C. Detection in Postmortem Analysis 34
        1. Detection of Anomalies Using the Real Attack Traces 34
        2. Detection of Anomalies Using the Simulated Attack Traces 36
        3. Effect of DWT.. 38
    D. Detection in Real-time Analysis 39
        1. Individual Reconstruction in DWT.. 39
            a. Thresholds Setting Through Statistical Analysis 39
        2. Detection of Anomalies Using the Real Attack Traces 40
        3. Detection of Anomalies Using the Simulated Attack Traces 42
    E. Multidimensional Indicators.. 44
        1. Analysis of Network Traffic by Port Numbers 44
        2. Number of Flows 45
        3. Comprehensive Traffic Analysis.. 46
        4. Attack Volume 48
    F. Summary 50

IV TRAFFIC IMAGE-BASED ANOMALY DETECTION.................................. 51

    A. Network Traffic as Image 51
        1. Network Traffic.. 51
        2. Visual Representaion 52
    B. Requirements for Representing Network Traffic as Images 55

CHAPTER Page

    1. Sampling Rates ... 55
    2. Discrete Cosine Transform (DCT) ... 57
        a. Validity of Intra-frame DCT ... 58
        b. Inter-frame Differential Coding ... 59
        c. Effect of Sampling Rates on DCT Coefficients ... 61
C. Visual Modeling Network Traffic as Images ... 62
    1. Visual Patterns in Normal Network Traffic ... 62
    2. Visual Patterns in Semi-random Targeted Attacks ... 64
    3. Visual Patterns in Random Targeted Attacks ... 65
    4. Visual Patterns in Complicated Attacks ... 66
    5. Flow-based Visual Representation ... 68
    6. Multidimensional Visualization ... 69
        a. Visual Patterns in Port Number Domain ... 71
D. Anomaly Detection Using Scene Change Analysis ... 73
    1. Threshold Setting Through Statistical Analysis ... 73
        a. Real-time Analysis ... 73
        b. Postmortem Analysis ... 74
        c. Thresholds Setting ... 77
    2. Anomaly Detection ... 77
    3. Attack Estimation Using Motion Prediction ... 78
    4. Processing and Memory Complexity ... 80
E. Identification ... 82
    1. Identification of Attackers and Victims in Byte-segment Level of IP Address ... 82
    2. Identification of Attackers' and Victims' Entire IP Address ... 84
F. Comparison with IDS ... 86
    1. Intrusion Detection System and Snort ... 86

CHAPTER Page

        2. Overall Results of Snort and NetViewer 86
        3. Comparison of Snort and NetViewer 88
    G. Summary 90

V EVALUATION OF DIFFERENT PACKET HEADER DATA AS SIGNALS FOR ANOMALY DETECTION 92

    A. Traffic Signals 93
        1. Scalar Signals 94
            a. Byte Counting 94
            b. Packet Counting 94
            c. Flow Counting 95
        2. Vector Signals 96
            a. Protocol Composition 96
            b. Image-based Signals 97
    B. Evaluation Methodology 98
        1. Type I and II Errors 98
        2. Likelihood Ratio 99
        3. Statistical Analysis Based on 3σ 99
    C. NP Test 100
        1. PDF of H_0 and H_1 100
        2. Bayes' Likelihood Ratio Test 104
            a. Effect of Threshold (η) in NP Test 106
        3. Expected True and False Positive Rates 108
        4. Application for Traffic Signals in NP Test 109
    D. Evaluation of Signals 112
        1. Byte Counting and Packet Counting 112
        2. Flow Counting 114
        3. Protocol Composition 115

CHAPTER Page

E. Evaluation of Image-based Signals 116
  1. Packet Distribution in Address Domain........ 116
  2. Flow Distribution in Address Domain 117
  3. Packet Distribution in Port Domain 119
  4. Multidimensional Signal 120
F. Analytical Results 121
  1. Sensitivity of Signals to Thresholds........ 121
  2. Real-time and Postmortem Analysis 122
  3. General Discussion of Results 123
  4. Consideration Using Other Traces 124
  5. Effect of Exponentially Weighted Moving Average 126
  6. Sensitivity of Signals to Attack Volume 128
G. Tool Development 130
  1. Frame of NetViewer 131
  2. Future Version of NetViewer 132
H. Summary 133

VI CONCLUSIONS AND FUTURE WORK 134

REFERENCES 138

APPENDIX 146

VITA 150

LIST OF FIGURES

FIGURE Page

1 The various filtering points 11

2 The block diagram of our detector 13

3 The autocorrelation function of the correlation coefficient signal (Fig. 5(a)) over 2 days 18

4 The data structure for computing weighted correlation 20

5 Comparison among signals of autocorrelation coefficient, full 32-bit correlation and our data structure 22

6 The cross-covariance functions show that the random variables of three kinds of signals are to be correlated 23

7 A multilevel two-band wavelet decomposition and reconstruction 25

8 The timing diagram in detection mechanism 28

9 The distribution of the ambient traces without attacks 31

10 The autocorrelation function of wavelet transformed signal in the sample paths. 33

11 IP address-based detection results using KREONet2 real attack traces in postmortem 35

12 IP address-based detection results using USC real attack traces in postmortem 36

13 IP address-based detection results using simulated attack traces in postmortem 37

14 Address-based detection results using real attack traces in real-time 41

FIGURE Page

15 Address-based detection results using simulated attack traces in real-time........ 42

16 Port numbers-based detection results in postmortem.. 44

17 Cross-correlation function of IP address (X), port numbers (Y) and number of flows (Z) 46

18 The multidimensional detection results using IP address and port numbers 47

19 The visualization of network traffic signal in IP address.................................. 54

20 The relationship between MSE and sampling rates 56

21 The relationship between sampling rates and nature of the traffic 57

22 The relationship between intra-frame MSE and DCT coefficients..................... 60

23 The relationship between DCT coefficients and sampling rates in normalized intra-frame MSE... 61

24 Visual measurement of normal network traffic.. 63

25 Visual measurement of semi-random typed attack 64

26 Visual measurement of (horizontal) random styled attack................................. 65

27 Visual measurement of (vertical) random styled attack..................................... 65

28 Visual measurement of complicated network attack.. 67

29 Visual measurement of the *flow-based* network traffic 68

30 Multidimensional visualization of semi-random attack..................................... 70

FIGURE Page

31 *Port-based* visual measurement 72

32 The results from trace-driven evaluation for detecting attacks in real-time 74

33 The trace-driven evaluation results from address-based image signal in destination address for detecting attacks 75

34 Illustrative procedures show potential attack estimation using motion prediction.. 79

35 The detection report for the Fig. 25 of anomaly identification 83

36 The flowchart in concatenation of identification 85

37 Snort report during 24 hours, April 28 through 29, 2004 87

38 NetViewer detection for April 28 through 29, 2004 88

39 The identification report on Apr 29, 2004, 05:01:00 by NetViewer................... 89

40 The relation of byte count and paket count of flows.. 95

41 Illustration of PDF and true/false positive rates... 102

42 The relationship between thresholds (η) and measurements criteria 107

43 Normality of address-based signals in real-time mode 109

44 Trace-driven detection results in scalar signals.. 112

45 Detection results based on protocol composition... 114

46 The relationship between measurements and thresholds 122

FIGURE Page

47 The relationship between performance and forgetting factor 126

48 The relationship between performance and attack volume 127

49 The running example of NetViewer 130

50 Detection results based on protocol composition 147

LIST OF TABLES

TABLE Page

1 Address Persistence in Successive Periods 12

2 The Descriptions of Five Attacks in KREONet2 Traces 15

3 The Nine Kinds of Simulated Attacks 16

4 The Statistical Measures in the Sample Paths 33

5 The Detectionability of the IP Correlation Signal and the DWT Signal 38

6 The Relation between Latencies and Confidence Levels in Nine Kinds of Attacks in Real-time Mode 43

7 The Detectionability of the Various Mixture Ratios of Auckland-IV Traces in Postmortem Mode 49

8 The Detection Latency of the Various Mixture Ratios of Auckland-IV Traces in Real-time Mode 49

9 Results of Scalar Signals 113

10 Results of Protocol Composition Signals 115

11 Results of Address-based Signals 116

12 Results of Flow-based Signals 118

13 Results of Port-based Signals 119

14 Results of Multidimensional Signals 120

TABLE Page

15 Port-based Measurement Results of Campus (TAMU) Traces 124

16 Results of Multidimensional Signals of Campus (TAMU) Traces 124

17 Address-based Measurement Results of ISP (USC) Traces 125

18 Results of Multidimensional Signals of ISP (USC) Traces 125

19 The Detection Latency of the Various Mixture Ratios of Simulated Attacks in Real-time Mode 128

20 Results of Protocol Composition Signals 148

CHAPTER I

INTRODUCTION

Malicious network traffic, such as DoS (denial of service) floods, worms and other malicious codes, has become a common threat for communication on the Internet. Currently, there are many well-known automated self-propagating codes that can be classified as DoS, DDoS (distributed denial of service), and DRDoS (distributed reflection denial of service) attacks. Compound attacks consisting of more than one strategy, self-modifying worms, and encryption-based worms are likely to increase this threat further in the future [1]. A number of recent studies have pointed to the need for fast detection of such worms for any effective mechanisms for thwarting such traffic [2]. The frequent and increasing attacks on computer network infrastructure have led to an increased need for developing techniques for analyzing and monitoring network traffic. Network traffic monitoring and analysis tools are being employed to counter this threat. If efficient analysis and visual tools were available, it could become possible to detect the spread of such malicious traffic in real-time, and to appropriately take action to contain the attacks before they have had much time to propagate across the network. In this dissertation, we study the possibilities of traffic-analysis based visual mechanism for attack detection and identification.

A. Motivation

The motivation for this work came from a need to reduce the likelihood that an attacker may hijack the AD (administrative domain) machines to stage an attack on a third party. By report of Incident Response Team in 1998, 97% of the attacked sites never knew about the attacks [3]. An AD like a campus may want to prevent such use of

---

The journal model is *IEEE/ACM Transactions on Networking*.

its machines for preventing attacks, limiting misuse of its machines and possibly limiting the liability from such attacks. With this aim, we focus on analyzing network traffic at the edge of an AD. In particular, we study the utility of observing packet header data of traffic at the network edge, such as source/destination addresses, port numbers, in order to detect attacks/anomalies originating from the AD.

Detecting anomalies/attacks close to the source allows us to limit the potential damage close to the attacking machines. Traffic monitoring close to the source may enable the network administrator quicker identification of potential anomalies and may allow better control of AD's resources. Attack propagation could be slowed through early detection of attacks.

Traditionally, Intrusion detection system (IDS) tools that rely upon operating system logs, process behaviors and firewall logs have been employed to monitor the network traffic. IDS tools monitor network and host traffic to filter the packets that belong to attacks with known behavior patterns. However, the frequent appearance of novel attacks compromises such analysis making true detection of unknown malicious traffic difficult. Measurement-based IDS tools and network traffic analysis have recently started attracting attention as a potential complementary approach [4], [5], [6], [7]. In this dissertation, we focus our attention on measurement-based approaches to anomaly detection.

As a methodological approach towards this aim of studying and classifying traffic on the network based on usage and protocols, a number of tools such as FlowScan [7], Cisco's FlowAnalyzer, and AutoFocus [6], are used as traffic analyzers. Some of these tools provide real-time reporting capability, but much of the analysis is done off-line. These tools have been effectively utilized for traffic engineering and postmortem anomaly detection. However, rigorous real-time analysis is needed for detecting and identifying the anomalies so that mitigation action can be taken as promptly as possible.

Some of these tools are based on the volume of traffic such as byte counts and packet counts. When links are congested, it is possible to always observe a fully utilized link without gaining further information about possible changes in network traffic. For example, traffic volume in terms of byte counts or packet counts could be employed to detect flood attacks which consume significant amount of network bandwidth. However, sophisticated low-rate attacks [8] and replacement attacks, which do not give rise to noticeable variance in traffic volume, can be undetected when network measurement tools analyze the traffic based on traffic volume signals. Similarly, when links are not sufficiently provisioned, normal traffic volumes may reach the capacity of the links most of the time. In such cases, attack traffic may not induce significant overshoot in traffic volume (merely replacing existing normal traffic) and hence may make traffic volume signal ineffective in detecting attacks.

Most current monitoring/policing tools that employ flow level analysis. Link speeds are increasing and the traffic analysis tools need to scale with the link speeds. The tools that collect and process flow data may not scale to high-speed links as they focus on individual flow behavior.

Our approach tries to look at aggregate packet header data in order to improve scalability. Our work here brings techniques from image processing and video analysis to visualization and real-time analysis of traffic patterns.

Our approach to detecting anomalies envisions two kinds of detection mechanisms: postmortem and real-time modes. A postmortem analysis may exploit many hours of traffic data as a single data set, employing more rigorous, resource-demanding techniques for analyzing traffic. Such an analysis may be useful for traffic engineering purposes, analysis of resource usage, understanding peak demand etc. Real-time analysis would concentrate on analyzing a small window of traffic data with a view to provide a quick and possibly dirty warning of impending/ongoing traffic anomalies. Real-time analysis may rely on less sophisticated analysis because of the resource demands and imminence of attacks.

Previous work has shown that a postmortem analysis of traffic volume (in bytes) can reveal changes in traffic patterns [4], [9], [10]. In this dissertation, we also study the effectiveness of such analysis in real-time analysis of traffic data. Real-time analysis may enable us to provide means of online detection of anomalies while they are in progress. Real-time analysis may employ smaller amounts of data in order to keep such analysis simple and efficient. At the same time, the data cannot be so small that meaningful statistical conclusions cannot be drawn. Data smoothing techniques can be employed to overcome such difficulties. However, real-time analysis may also require that any indications of attacks or anomalies be provided with short latencies. This tension between robustness and latency of anomaly detection makes real-time analysis more challenging.

Our approach passively monitors packet headers of network traffic at regular intervals, and generates signals/images of this packet header data. These signals/images are analyzed to find whether any abnormalities are observed in the traffic. Recent studies have shown that the traffic can have strong patterns of behavior over several timescales [4], and our work will show the possibility of analysis of WSS (wide-sense stationary) property in network traffic [5]. Recent work in [11] has shown that Gaussian approximation could work well for aggregated traffic. Self-propagating and automated malicious codes perturb normal network traffic patterns in general. By observing the traffic and correlating it to the previous normal states of traffic, it may be possible to see whether the current traffic is behaving in a similar/correlated manner. The network traffic could look different because of flash crowds, changing access patterns, infrastructure problems such as router failures, and DoS attacks. In case of anomalous traffic such as flash crowds and DoS attacks, the usage pattern of network may be changed and peculiarities could be represented in the generated signals/images. When anomalies are detected, further analysis can characterize the anomalies by their nature into several categories (random attack, targeted attack, multi-source attack, portscan attack etc.) and help in mitigating the attacks. Our approach relies on analyzing packet header data in order to provide indications of possible abnormalities in the traffic.

B. Previous Research

When there are insufficient resources (like bandwidth) under a large network load, we need to know which flow is receiving more than its fair share of service for detecting DoS attacks staged by few flows. The RED (random early detection) can be interpreted as a statistical method of using random selection to act selectively on high rate flows without keeping per-flow state information even though it is not effective on the non-responsive traffic. [12]. SACRED, LRU (least recent used)-RED, and LRU-FQ employ partial state to enhance the success of detecting large flows, which find a non-responsive UDP flow by monitoring flows above certain policy driven rate [13], [14], [15]. SRED (stabilized RED) shows that the misbehaving UDP sources can be identified by comparing with a zombie list [16].

Many approaches have been studied to detect, prevent and mitigate the malicious activities. For example, rule-based approaches, such as IDS (intrusion detection system), try to match the established rules to the potential DoS attack from external incoming traffic near the victims. To cope with novel attacks, IDS is required to be updated with the latest rules [17], [18]. In contrast, some approaches proactively seek a method that suppresses the overflowing of traffic at the source [19]. Controls based on rate limits [20] have been adopted for reducing the monopolistic consumption of available bandwidth to diminish the effects of attacks, either at the source or destination [19], [21], [22]. The most apparent symptoms of bandwidth attack may be sensed through monitoring bit rates [23] and packet counts of the traffic flow. Bandwidth accounting mechanisms have been suggested to identify and contain attacks [24], [25], [26], [27], [28]. Work in [27] making entries similar to the LRU cache by sampling, all subsequent packets belong to the entries are monitored. To reduce the false alarm error due to high sampling probability, it exploits parallel multistage filters combined with early removal, which require a large memory and may be not effective for aperiodic burst traffic. Packeteer [29] and others offer commercial products that can account traffic volume along multiple dimensions and allow policy-based rate control of bandwidth. Pushback mechanisms have been proposed to contain the detected attacks closer to the source [22], [26], [30]. Traceback techniques have been proposed to trace the source of DDoS attacks even when the source addresses may be spoofed by the attacker [31].

Recently statistical analysis of aggregate traffic data has been studied [4], [9], [10]. The work in [4], [9] have studied traffic volume as a signal for wavelet analysis and this earlier work have considerably motivated our current study here. The work in [10] has shown the application of wavelets to network traffic data. Our study builds on this earlier work and extends the statistical analysis of traffic data further. Fourier transforms have been applied to network traffic to study its periodicity [32].

Various forms of signatures have been traditionally utilized for representing the contents or identities. In digital information retrieval, the signature that is constructed by taking several hash values is applied for indexes representing words [33]. Traffic analysis signatures have been proposed for detecting anomalies. For example, disproportion of bi-directional flows can be used as a signature of anomalistic traffic [34]. The changing ratios (i.e., the rate of decrease) between the flow numbers of neighboring specific bit-prefix aggregate flows can be used for detecting peculiarities [35].

A number of popular monitoring tools such as FlowScan, Cisco's FlowAnalyzer, and AutoFocus [6], are used as traffic analyzers. FlowSan is open source software to gather and analyze network flow data taken from NetFlow records of Cisco routers [7]. In the FlowScan, cflowd writes raw flow files that wait to be post-processed by flowscan for providing against heavy-traffic flows or flood-based DoS attacks. However, excessive backlog of flow files may make real-time analysis difficult. Using FlowScan, characterization of anomalous network traffic behavior can be described at the flow level [36].

Recently traffic volume metrics, such as byte counts and packet rates, have been analyzed using wavelets to detect anomalies in network traffic [4]. While earlier work analyzed traffic as a time series of a single variable, our work here also tries to analyze distributions over different domains of packet header data, particularly the address space and port number space [37]. Our work also brings the tools from image processing and video analysis to traffic analysis.

Sketch-based techniques are shown to perform close to that of per-flow methods for network traffic analysis [38]. The technique employs multiple hash functions on address to generate data for traffic analysis. To detect significant changes, it implements a time series forecasting model on top of such summaries, which may be not useful for movable attacks because it will miss estimations that do not appear again after they experience significant change. A bloom filter [39] is a method for representing a set of n elements in m ($\leq n$) hash values to support membership queries. For each element, the bits at positions from independent hash functions are set to '1', where a particular bit might be set to '1' multiple times. Given a query for b, we check the bits at positions from hash values. If any of them is '0', then certainly b is not in the set. Otherwise we conjecture that b is in the set although there is a certain false positive probability, which is a clear trade-off between m and the false positive rate. Recent work in [40] has similarly employed 3 hash functions and LRU-like caching for extracting traffic attack patterns. While hashing techniques are general and powerful, (a) it is harder to identify the source or destination of attacks without additional work due to one-way functionality (b) randomization makes it harder to infer general trends or styles of attack as they happen. Our approach, though not as general, can be considered to employ four specific hash functions on the address space, while still allowing visualization of traffic patterns. The visualization part of the work in [40] has some similarities in visualization of network traffic (with significant differences in data representation and anomaly detection).

Much of the work reported here draws from the large body of work in wavelet analysis, image processing and video analysis. Various forms of approaches have been traditionally utilized for detecting scene changes in image processing. There are methods based on DC coefficients of the each transformed block in the image [41], color histogram differences [42], characteristic patterns in the standard deviation of pixel intensities for detection of fades [43], and color histogram of DC coefficients [44]. The existing methods have mainly been targeting the object in the center of camera focus, yet, the network image processing is necessary to consider the entire space due to uncertainty of attacks.

C. Contributions and Outline

The rest of the dissertation is organized as follows. In chapter II, we discuss our approach and generation of traffic signal. In chapter III, we describe the wavelet transform of the address correlation signal and the experimental results of the anomaly detection mechanism, in both postmortem and real-time. In chapter IV, we introduce the image-based modeling of network traffic and its statistical analysis. In chapter V, we introduce the various traffic signals that have been proposed for analysis and anomaly detection and comprehensively evaluate the effectiveness of these various signals. Our analysis is based on a number of real world traces. Our analysis employs both statistical measures and measures based on classical NP (Neyman-Pearson) Test employed in signal detection theory. Chapter VI concludes our dissertation and provides directions for future work.

In this dissertation, we will report on our measurements conducted on real traces of traffic at four major networks. This dissertation will make the following significant contributions: (a) studying the feasibility of correlation of destination IP addresses through signal processing such as wavelet analysis, (b) employing packet header data as images for traffic visualization, (c) employing image processing and compression techniques for efficiently storing and processing such traffic data, (d) evaluates the effectiveness of such measures in detecting and identifying the attacks in real-time with very small latencies, (e) provides a comprehensive evaluation of effectiveness of a number of different signals derived from network traffic headers, (f) provides an approach based on classical NP Test for evaluating the effectiveness of network traffic signals and (g) shows that statistical techniques can be as effective or sometimes more effective than the NP Test because of changing attack patterns.

CHAPTER II

AGGREGATED TRAFFIC ANALYSIS

This chapter presents our approach and methodology, and discusses the use of data structure for computing traffic signals/images of packet header data.

A. Traffic Analysis at the Source

We focus on analyzing the traffic at a network's edge. Traffic monitoring at a source network enables a detector to detect attacks early, to control hijacking of AD (administrative domain, e.g., campus) machines, and to limit the squandering of resources.

There are two kinds of filtering based on traffic controlling point. Ingress filtering protects the flow of traffic entering into an internal network under administrative control. Ingress filtering is typically performed through firewall or IDS rules to control inbound traffic originated from the public Internet. On the other hand, egress filtering controls the flow of traffic originating from the internal network. Thus, internal machines are typically the origin of this outbound traffic in view of an egress filter. As a result, the filtering is performed at the AD edge [45]. Fig. 1 conceptually illustrates the various filtering points for inspection of the network traffic. Outbound filtering has been advocated for limiting the possibility of address spoofing i.e., to make sure that source addresses correspond to the designated addresses for the AD. With such filtering in place, we can focus on destination addresses and port numbers of the outgoing traffic for analysis purposes.

Our approach is based on the following observations: the outbound traffic from an AD is likely to have a strong correlation with itself over time since the individual accesses have strong correlation over time. Recent studies have shown that the traffic

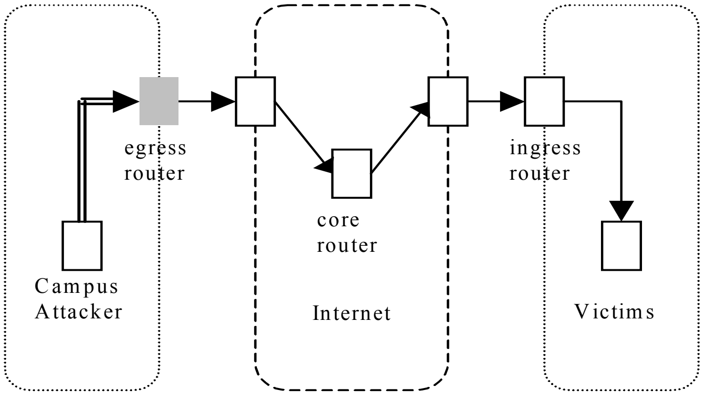


Fig. 1. The various filtering points.
Our approach is based on the outbound traffic at the source.

can have strong patterns of behavior over several timescales [4] and a time series of packet bytes per time slot are not independent but indeed rather strongly correlated [11]. For example, the traffic over a week looks very similar to the next week. Similarly, the traffic over a day exhibits a strong correlation across the next few days. It is possible to infer that some correlation exists on their weekly or daily consumption patterns. We hypothesize that the destination addresses will have a high degree of correlation for a number of reasons: (i) popular web sites, such as yahoo.com and google.com, are shown to receive a significant portion of the traffic, (ii) individual users are shown to access similar web sites over time due to their habits, and (iii) long-term flows, such as ftp download and video accesses, tend to correlate addresses over longer timescales. If this is the case, sudden changes in correlation of outgoing addresses can be used to detect anomalies in traffic behavior. For example, a denial of service attack on a single target machine will likely increase the correlation of addresses during the attack period. Similarly, a worm attack on many random addresses will likely decrease the correlation of addresses. This hypothesis (which needs to be verified) suggests that address correlation across samples could be a useful signal.

TABLE 1
ADDRESS PERSISTENCE IN SUCCESSIVE PERIODS

Hit ratio (%)	**Sampling instances**							
	2 am	*5 am*	*8 am*	*11 am*	*2 pm*	*5 pm*	*8 pm*	*11 pm*
adjacency	33	38	44	32	38	33	35	33
persistency	100	38	27	21	20	19	18	16
previous day	24	37	33	39	37	38	31	29

This hypothesis is corroborated by the data shown in Table 1. The table shows the correlation of addresses across different times of day based on NZIX-II traces from NLANR (National Laboratory for Applied Network Research) [46]. The traces are sampled for 90 seconds every 3 hours and analyzed from three different viewpoints. The flows are defined by triple of destination address, destination port and protocol, and specially using 24-bit prefix destination IP address. The subject of investigation is the top 100 flows in packet counts instead of all of the flows. The packets of the top 100 flows occupy about 95% of all the packets. The first row named 'adjacency' shows the recurrence of destination addresses between adjacent 3-hour periods. For the 8-am column, a total of 44 addresses reappear from the previous 5-am instant. The second row titled 'persistency' explains the lasting continuance of addresses from the 2-am instant through the day. It is observed that 27 of the top 100 addresses persist from 2-am point to the top 100 addresses of 8-am instant. The last row titled 'previous day' illustrates the persistence of popular addresses in the same time points across two consecutive days. It is observed that 33 of the top 100 addresses remained the same in the 8-am instant across two different days. The correlation would be higher if we considered the traffic volume to these addresses.

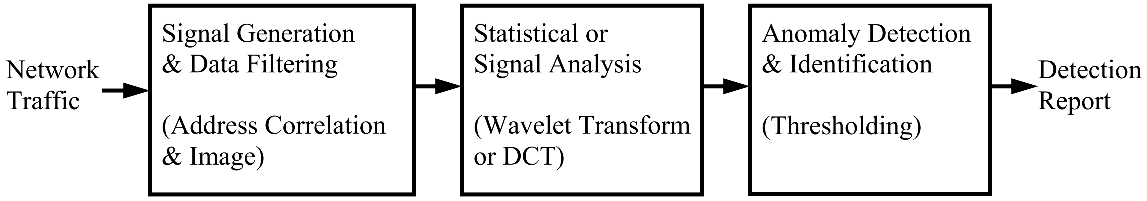

Network Traffic
Signal Generation & Data Filtering
(Address Correlation & Image)
Statistical or Signal Analysis
(Wavelet Transform or DCT)
Anomaly Detection & Identification
(Thresholding)
Detection Report

Fig. 2. The block diagram of our detector.

A second approach to using addresses as useful signals is based on the hypothesis that traffic (byte volume or packet volume or the number of flows) distributions over the address domain (source, destination or the two together) could be different during normal and attack periods. If this were the case, the anomalies can be detected by analyzing the traffic distributions over the underlying domain.

B. General Mechanism of the Detector

Our detection mechanisms can be explained in three major steps as shown in Fig. 2. Traffic is sampled at regular intervals to obtain a signal that can be analyzed through statistical techniques and compared to historical norms to detect anomalies.

The first step is a traffic parser, in which the address correlation/image signal is generated from packet header traces or NetFlow records as input. In this step, the network traffic is first filtered to produce a signal that can be analyzed. So far, we have discussed how correlation of destination addresses may be used as a potential signal. The particular signal that is employed may depend on the nature of the traffic. Fields in the packet header, such as addresses and port numbers, and traffic volume can be used as a signal. Packet header data, due to its discrete nature, poses interesting problems for analysis as discussed later. Sampling may be used to reduce the amount of data at this stage. The generations of address correlation and image signal are explained in chapter II and IV.

The second step involves data transformation for statistical analysis. In this dissertation, we employ Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) to study the address and port number correlation and distribution over several timescales. Wavelet transforms have been employed to study the traffic volume earlier [4], [9], [10]. Unlike previous work, we selectively/individually reconstruct decomposed signal across specific timescales based on the nature of attacks. Our wavelet analysis of traffic signals is explained in chapter III and DCT approach is explained in chapter IV.

The final stage is detection and identification, in which attacks and anomalies are checked using thresholds. The analyzed information will be compared with historical thresholds of traffic to see whether the traffic's characteristics are out of regular norms. Sudden changes in the analyzed signal are expected to indicate anomalies. This comparison will lead to some form of a detection signal that could be used to alert the network administrator of the potential anomalies in the network traffic. We report on our results employing correlation of destination addresses and distribution of image-based signal as traffic signals in chapter III and IV. In this dissertation, we identify the suspicious attackers/victims and estimate the movement of attack patterns using motion prediction algorithms as explained in chapter IV. The efficacy of these signals for anomaly detection is carefully evaluated in chapter V.

C. Approach

To verify the validity of our approach, we employ trace-driven evaluation of the developed techniques. We run our algorithm on four traces of network traffic.

First, we examine our method on traces from the University of Southern California, which contains real network attacks in the `pcap` header format. Additionally to inspect the performance of our detector on backbone links, we examine the mechanism on two

TABLE 2
THE DESCRIPTIONS OF FIVE ATTACKS IN KREONET2 TRACES

	1	**2**	**3**	**4**	**5**
Duration	5.3 h	4.5 h	4.1 hours	12.3 h	3.6 h
IP	semi-random	random[a]	random	semi-random random	random
Protocol	TCP	UDP	TCP/UDP	TCP/UDP/ICMP	UDP
Port	#80	#1434	random/#1434	#80 / #1434 / #0	#1434
Size	48B	404B	random/ 404B	48B / 404B/ 28B	404B

a. SQL Slammer

kinds of KREONet2 traces from July 21, 2003 to July 28, 2003, and from Oct. 12, 2003 to Oct. 26, 2003 which contain real worm attacks. Currently KREONET (Korea Research Environment Open NETwork) member institutions are over 230 organizations, which include 50 government research institutes, 72 universities, 15 industrial research laboratories, etc [47]. KREONet2 trace is a collection of NetFlow trace files generated by the router which is connecting KREONet2 and STAR TAP with the 155Mbps international ATM link. KREONet2 is also peering with Abilene using this link. Over tens of thousands of systems in Korea's universities and research institutes are using this link, and most of traffic from Abilene and STAR TAP bound for Korea's research institutes and universities is also using this link. In the 2nd traces employed, there are 5 major attacks as described in Table 2 and a few instantaneous probe attacks. It generates 4345 samples in case of 2-minute sampling period. Among the observations, the suspected activities reach to 782 times, which are judged by traffic engineering. Third, to compare our method with Snort (an IDS tool), we examine the mechanism on a live network in Texas A&M University (TAMU) campus.

Moreover, to evaluate the sensitivity of our detector’s performance over attacks of various configurations, we employ the packet traces from the NLANR (National Laboratory for Applied Network Research), which are later superimposed with

TABLE 3
THE NINE KINDS OF SIMULATED ATTACKS

	1 (2, I, SD)	**2 (2, I, SR)**	**3 (2, I, R)**	**4 (2, P, SD)**	**5 (2, P, SR)**	**6 (2, P, R)**	**7 (1, P, SD)**	**8 (1, P, SR)**	**9 (1, P, R)**
Duration	2h	2h	2h	2h	2h	2h	1h	1h	1h
Persistency	intermittence	int.	int.	persistence	per.	per.	per.	per.	per.
IP	single destination	semi-random[a]	random[b]	single destination	semi-random	random	single destination	semi-random	random
Protocol	ICMP	TCP	UDP	ICMP	TCP	UDP	ICMP	TCP	UDP
Port	#80	random	#1434	#80	random	#1434	#80	random	#1434
Size	random	4KB	404B	random	4KB	404B	random	4KB	404B

a. Code Red

b. SQL Slammer

simulated virtual attacks [46]. We employ Auckland-IV traces for these experiments collected at the University of Auckland Internet access link for 45 days. These IP header traces are utilized in length from 3 days to several weeks. Outbound traffic in these traces transmitted about 5000 connections at the rate of 5Mbps and 1500 packets/second. These traces were anonymized, but preserved IP prefix relationships.

1. Simulated Attacks

Besides the actual attacks observed in the USC, KREONet2 and TAMU traces, we construct virtual attacks on the Auckland-IV traces. This allows us to test the proposed technique under different conditions. We consider nine kinds of combinational attacks as shown in Table 3. These attacks cover a diversity of behaviors and allow us to deterministically test the efficacy of proposed mechanisms. The particular behaviors of these attacks have been motivated by recent SQL Slammer [48] and Code Red attacks. These are classified by following criteria.

- Duration: The first 6 attacks continue to assail for 2 hours. The remaining 3 attacks last for 1 hour.

- Persistency: The first 3 attacks send malicious packets for 3 minutes and pause for 3 minutes. Such pattern is repeated through the attack duration. While the filtering may mitigate the overhead of the attacker's continuing scan traffic, a more sophisticated attacker might have stopped scanning. It may be possible to conceal attacker's intentions through repeating attack and pause periods. So, it is intended to model intelligent and crafty attackers that attempt to dilute their trails. The other remnant attacks continue to assault throughout the attack period.
- IP address: The 1st attack among every 3 attacks targets for a single destination IP address. In a hypothetical situation, the attackers target a famous site such as the White House, CNN or Yahoo etc. This target may be really one host in case of 32-bit prefix, occasionally aggregated neighboring hosts in case of /*x* bit prefix. The 2nd attack style imitates from the IP address generation scheme of the notorious Code Red II worm. That is to say, a portion of addresses preserve the class-A and a partition of addresses preserve class-B for the infiltration efficiency. The 3rd type is randomly generated address that was used for the Code Red I and SQL Slammer worm.
- Protocol: The three major protocols, ICMP, TCP and UDP, are exploited in turn
- Port: The 1st port among every 3 attacks is a representative #80 that stands for the reserved ports for well-known services. The 2nd port targets for randomly generated destination ports. It is useful to detect port-scan that is used to probe a loosely defensive port. The 3rd port is a #1434 that acts for the ephemeral client ports, which was exploited in SQL Slammer worm
- Size: There are three different byte counts of packets. The three denominations are random size, 4KBytes and 404Bytes [45].

Our attacks can be described by a 3-tuple (duration, persistency and IP address). We superimpose these attacks on ambient traces.

The mixture ratios of attack traffic and normal traffic range from 1:2 to 1:10 in packet counts in our experiments. The detection performance is slightly affected by

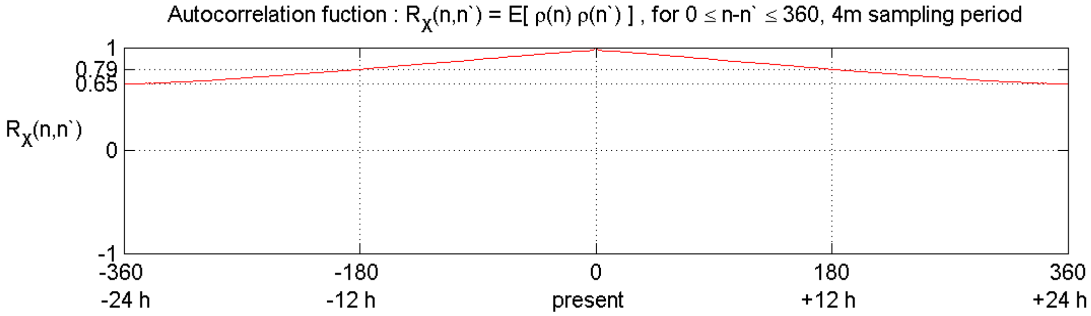


Fig. 3. The autocorrelation function of the correlation coefficient signal (Fig. 5(a)) over 2 days. We could infer that the traffic has a close positive address correlation.

mixture ratios. Replacement of normal traffic with attack traffic is easier to detect and hence not considered here.

D. Data Structure for Computing Signal

Our approach collects packet header data at an AD's edge over a time period that is the sampling period. Individual fields in the packet header are then analyzed to observe anomalies in the traffic. Individual fields in the traffic header data take discrete values and show discontinuities in the sample space. For example, IP address space can span 2^{32} possible addresses and addresses in a sample are likely to exhibit many discontinuities over this space making it harder to analyze the data over the address space. In order to overcome such discontinuities over a discrete space, we convert packet header data into a continuous signal through correlation/distribution of samples over successive samples.

To investigate the ensembles of a random process, a correlation coefficient which is a normalized measure of the strength of the linear relationship between random variables is usually employed [5], [49]. For each address, a_m, in the traffic, we count the number of packets, p_{mn}, sent in the sampling instant, s_n. We can define IP address correlation coefficient signal in sampling point *n* as follows.

$$\rho(n)=\frac{\sum_m (p_{mn-1}-\overline{p_{n-1}})*(p_{mn}-\overline{p_n})}{\sqrt{\sum_m (p_{mn-1}-\overline{p_{n-1}})^2}\sqrt{\sum_m (p_{mn}-\overline{p_n})^2}} \tag{2.1}$$

, where $\overline{p_{n-1}}$ and $\overline{p_n}$ are the mean values of packet counts in s_{n-1} and s_n

Fig. 5(a) shows the IP address correlation coefficient signal of the NLANR traces over 3 days by (2.1) which illustrates a close positive correlation between adjacent samples. Moreover, we analyze the similarity over time using autocorrelation function as shown in Fig. 3. If the level of aggregation in the number of flows and sampling duration is high such that spontaneous changes in the traffic get buried, we could assume that outbound traffic have a high degree of correlation over time.[1] Recent work in [11] shows that traffic would likely exhibit such properties at the edges of ADs.

In this dissertation, we employ a simplified correlation/distribution of time-series for computational efficiency in practice without compromising performance. This weighted correlation signal generation phase for destination addresses is explained below. Similarly the image-based signal generation scheme is illustrated in chapter IV.

In order to compute full 32-bit address correlation signal, we consider two adjacent sampling instants. We define address correlation signal in sampling point *n* as

$$C(n)=\sum_m p_{mn-1} * p_{mn} \,/\, \sum_m p_{mn} \tag{2.2}$$

If an address a_m spans the two sampling points *n-1* and *n*, we will obtain a positive contribution to *C(n)*. A popular destination address a_m contributes more to *C(n)* than an infrequently accessed destination, since we consider the number of packets being sent to the identical address.

---

[1] We do not claim that this assumption can be applied to all traffic.

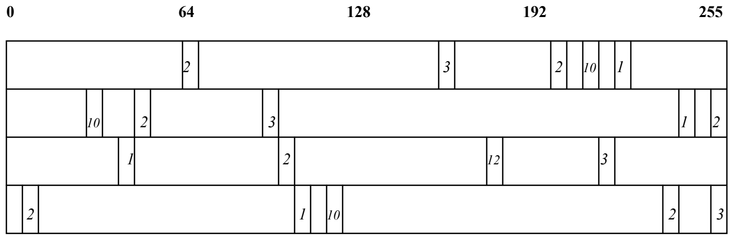


Fig. 4. The data structure for computing weighted correlation.

In order to minimize storage and processing complexity, we employ a simple but powerful data structure. This data structure, which is named *count*, consists of 4 arrays "*count [4]*". Each array expresses one of the 4 fields in an IP address. Within each array, we have byte-sized 256 locations, for a total of 4*256 bytes = 1024 bytes. A location *count [i][j][n]* is used to record the packet count for the address *j* in i^{th} field of the IP address in time interval *n* through scaling. This provides a concise description of the address instead of 2^{32} locations that would be required to store the address occurrence uniquely. We filter this signal by computing a correlation of the address in two success samples, i.e., by computing

$$C_{in} = \sum_{j=0}^{255} \frac{count\,[i][j][n-1]}{\sum_{j=0}^{255} count\,[i][j][n-1]} * \frac{count\,[i][j][n]}{\sum_{j=0}^{255} count\,[i][j][n]}, \quad i=0,1,2,3 \qquad (2.3)$$

Fig. 4 depicts the data structure that consists of the 2-dimensional array *count[i][j]*. The 1^{st} dimension array corresponds the 4 byte segments of the IP address (separated by a dot in the normal IP address representation), and is represented to be 4 rows in the data structure. The 2^{nd} dimension indicates the 256 entries of each IP address byte segment, and is expressed as the 256 columns in each row.

We illustrate this data structure through an example. Suppose that only five flows exist, their destination IP addresses and packet counts are as follows.

IP of Flow1 = 165. 91. 212. 255, P1 = 3

IP of Flow2 = 64. 58. 179. 230, P2 = 2

IP of Flow3 = 216. 239. 51. 100, P3 = 1

IP of Flow4 = 211. 40. 179. 102, P4 = 10

IP of Flow5 = 203. 255. 98. 2, P5 = 2

All entries in the count arrays are initialized to zeros. The packet counts of each flow are recorded to the corresponding position of each IP address segment as shown in Fig. 4.

In order to compute the correlation signal at the end of sampling point n, we simply multiply normalized values in the same position between the two data structures of samples $n-1$ and n, then sum up the multiplied values in each byte segment separately. Consequently four correlation signals are calculated as C_{0n} through C_{3n}.

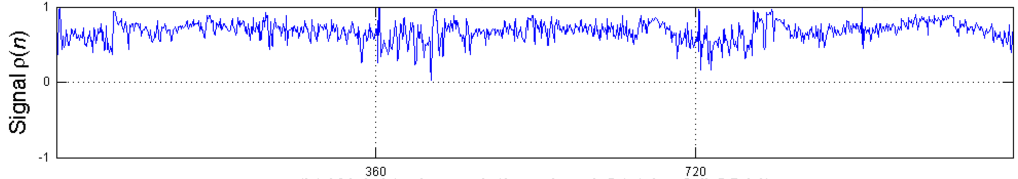
In general, such an approach of domain space reduction will reduce the domain space from 2^n to $(n/k)*2^k$, where k (= 8, above) is the basis for folding the domain space. The employment of this approximate representation of addresses allows us to reduce the computational and storage demands by a factor of 2^{22}.

In order to generate the address correlation signal $S(n)$ at the end of sampling point n, we multiply each segment correlation C_{in} with scaling factors α_i and generate $S(n)$ as

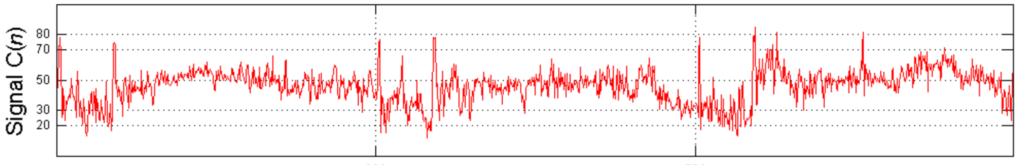
$$S(n)=A*(\sum_{i=0}^{3}\alpha_i C_{in})+B,\ where\ \sum_{i=0}^{3}\alpha_i=1 \tag{2.4}$$

This data structure has following advantages.

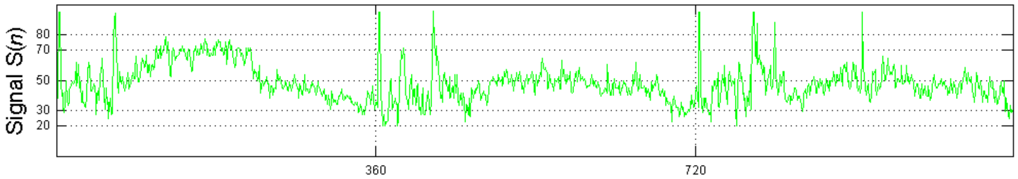
- The running time of the signal generation diminishes from $O(n)$ to $O(lgn)$.
- It uses a constant, small amount of memory regardless of the number of packets or flows.
- It is possible to identify the target IP addresses using reversibility of the data structure even though our approach can be considered to employ four specific



(a) Autocorrelation coefficient Signal ρ(*n*) on Auckland-IV traces : 4m sampling period



(b) Weighted correlation signal C(*n*) by full-32 bit



(c) Weighted correlation signal C_{in} by our data strucutre

Fig. 5. Comparison among signals of autocorrelation coefficient, full 32-bit correlation and our data structure.

> hash functions on the address space. By assembling the highest valued position in each of the 4 fields, specific attack objectives can be drawn. In Fig. 4 of the above example, we can induce that *211.40.179.102* as target IP address of *Flow4* when the positions of the highest value in each field are combined.

By properly choosing the scaling factors, we can obtain appropriate aggregation of address space. In this dissertation, we employ $\alpha_0 = \alpha_1 = \alpha_2 = \alpha_3 = 1/4$. On the other hand, we could employ different weights to give preferences to different portions of the address segments. For example, making $C_{2n} = C_{3n} = 0$ by setting $\alpha_2 = \alpha_3 = 0$, we only consider /16 addresses.

Our approach could introduce errors when the addresses segments match even though addresses themselves don't match. For example, if traffic consists of *w1.x1.y1.z1* and *w2.x2.y2.z2* in sample *n-1* and *w1.x1.y2.z1* and *w2.x2.y1.z2* in sample *n*, even though the actual address correlation is zero, our method of computing address correlation results in a high correlation between these sampling instants.

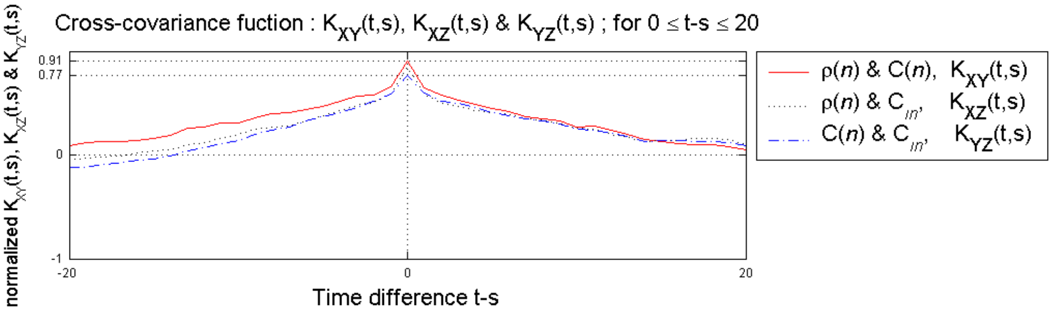


Fig. 6. The cross-covariance functions show that the random variables of three kinds of signals are to be correlated.
Specially, correlation coefficients are the zeroth lag of the covariance functions. These covariance functions normalize the sequence so the auto-covariances at zero lag are identically 1.0.

In normal traffic without attacks, we compared the full-32 bit address correlation with the correlation signal generated by our approach. Fig. 5(b) and 5(c) show the weighted correlation signal computed with the full-32 bit address by (2.2) and our data structure by (2.3) and (2.4) with respect to Auckland-IV traces. From the figure, we see that the differences are negligible i.e., our approach does not add significant noise. From a statistical point of view, they have an approximately same mean (about 50) and degree of dispersion (standard deviation $\cong$ 12.4 ~ 12.6). Moreover, we examine the similarity of above signals with cross-correlation coefficient which is normalized measure of linear relationship strength between variables. Suppose X(t) is correlation coefficient r.v. (random variable) defined by (2.1), Y(t) is full 32-bit r.v. by (2.2) and Z(t) is weighted correlation r.v. by (2.3) and (2.4). From Fig. 6, their cross-correlation coefficients are $\rho_{XY}(0) \approx 0.91$, $\rho_{XZ}(0) \approx 0.85$ and $\rho_{YZ}(0) \approx 0.77$, respectively.[2] Based on these results, we will employ signal of Fig. 5(c) for reducing the processing complexity in this dissertation.

---

[2] Too small sample size might grow incorrect statistical measures.

CHAPTER III

ADDRESS CORRELATION SIGNAL-BASED ANOMALY DETECTION

We propose a traffic anomaly detector, operated in postmortem and real-time, by passively monitoring packet headers of traffic. In this chapter, we suggest a technique for traffic anomaly detection based on analyzing correlation of destination IP addresses in outgoing traffic at an egress router. This address correlation data are transformed using discrete wavelet transform for effective detection of anomalies through statistical analysis. Results from trace-driven evaluation suggest that proposed approach could provide an effective means of detecting anomalies close to the source. We also present a multidimensional indicator using the correlation of port numbers as a means of detecting anomalies.

A. Discrete Wavelet Transform (DWT)

1. Data Transform

The generated signal can be, in general, analyzed by employing techniques such as FFT (Fast Fourier Transform) and wavelet transforms. The analysis carried out on the signal may exploit the statistical properties of the signal such as correlation over several timescales and its distribution properties. FFT of traffic arrivals may reveal inherent flow level information through frequency analysis. Wavelet transform of traffic traces has been employed in [4], [9]. We employ wavelet transforms, in this chapter, for analyzing the traffic signal.

Wavelet techniques are one of the most up-to-date modeling tools to exploit both non-stationary and long-range dependence [50], [51], [52], [53] and to analyze the properties of data series [54], [55]. In real situations, we encounter signals which are

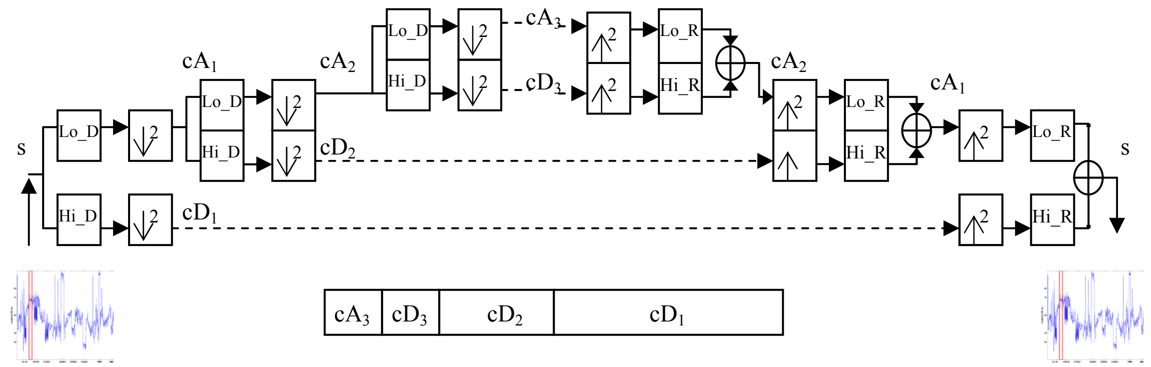


Fig. 7. A multilevel two-band wavelet decomposition and reconstruction.

characterized by abrupt changes and it becomes essential to relate to the occurrence of an event in time. Since wavelet analysis can reveal scaling properties of the temporal and frequency dynamics simultaneously unlike Fourier Transform used in [32], we compute a wavelet transform of the generated address correlation signal over several sampling points. Through signal can be detected in certain timescales and in certain positions of the timescales, we can induce the frequency and temporal components simultaneously.

Wavelet Transform plays a similar role as if a matched filter is employed to synchronize known signal and maximize SNR (Signal-to-Noise Ratio) from received noisy signal in communication system.

2. Discrete Wavelet Transform

We provide a brief overview of DWT (Discrete Wavelet Transform) in order to make our scheme clearer. DWT consists of decomposition (or analysis) and reconstruction (or synthesis). Fig. 7 illustrates a multilevel one-dimensional wavelet analysis using specific wavelet decomposition filters (Lo_D and Hi_D are the low-pass (or scaling) and high-pass (or wavelet) decomposition filters) and the reconstruction of the original signal [56].

For decomposition, starting from a signal *s*, the first level of the transform decomposes *s* into two sets of coefficients, namely approximation (or scaling) coefficients cA_1, and detail (or wavelet) coefficients cD_1. The input *s* is convolved with the low-pass filter Lo_D to yield the approximation coefficients. The detail coefficients are obtained by convolving *s* with the high-pass filter Hi_D. This procedure is followed by down sampling by 2. Suppose that the length of each filter is equal to *N*. If $T = length(s)$, the each convolved signal is of length $T + N - 1$ and the coefficients cA_1 and cD_1 are of length $L = floor((T+N-1)/2)$. The second level decomposes the approximation coefficient cA_1 into two sets of coefficients using the same method, substituting *s* by cA_1, and producing cA_2 and cD_2, and so forth. At level *j*, the wavelet analysis of the signal *s* has the following coefficients, $[cA_j, cD_j, cD_{j-1}, cD_{j-2}, \ldots , cD_2, cD_1]$.

Detail coefficients, $cD_{j,t}$ and approximation coefficients, $cA_{j,t}$ at level *j* are obtained as follows.

$$\begin{aligned}
&cD_{1,t}=\sum_{n=0}^{N-1}(Hi\_D)s_{2t+1-n \bmod T}, \quad && cA_{1,t}=\sum_{n=0}^{N-1}(Lo\_D)s_{2t+1-n \bmod T}\\
&cD_{2,t}=\sum_{n=0}^{N-1}(Hi\_D)cA_{1,2t+1-n \bmod T}, \quad && cA_{2,t}=\sum_{n=0}^{N-1}(Lo\_D)cA_{1,2t+1-n \bmod T}\\
&\ldots\ldots\ldots\\
&cD_{j,t}=\sum_{n=0}^{N-1}(Hi\_D)cA_{j-1,2t+1-n \bmod T}, \quad && cA_{j,t}=\sum_{n=0}^{N-1}(Lo\_D)cA_{j-1,2t+1-n \bmod T}\\
&\textit{where, } (Hi\_D) \textit{ and } (Lo\_D) \textit{ are the high-pass and low-pass filter}
\end{aligned} \tag{3.1}$$

For reconstruction, starting from two sets of coefficients at level *j*, that is the approximation coefficients cA_j and detail coefficients cD_j, the inverse DWT synthesizes cA_{j-1}, up-samples by inserting zeros and convolves the up-sampled result with the reconstruction filters Lo_R and Hi_R. Let *L* be the length of cA and cD, and *N* be the length of the filters Lo_R and Hi_R, then $length(s) = 2*L-N+2$. For a discrete signal of length *T*, DWT can consist of log_2T levels at most.

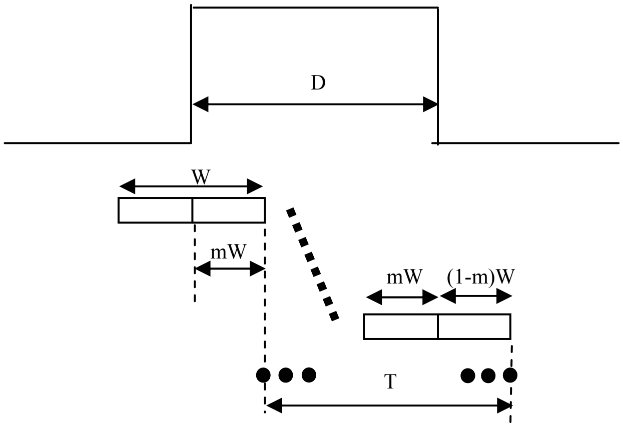
3. Timescale Selection

We iterate analysis level up to 8, which depends on the number of samples, in case of the postmortem analysis, so our final analysis coefficients are [cA_8, cD_8, cD_7, cD_6, cD_5, cD_4, cD_3, cD_2, cD_1]. We specify a daubechies-6 two-band filter. If we use all coefficients for reconstruction, we would restore the original weighted correlation signal. Due to the decimating operator, at level j, we have $T/2^j$-sized coefficients. That is, the filtered signal is down-sampled by 2 at each level of the analysis procedure; the signal of each level has an effect that sampling interval extends 2 times. Consequently it means that the wavelet transform identifies the changes in the signal over several timescales. When we use t minutes as sampling interval, the time range at level j spans $t*2^j$ minutes. For instance, when we use 1-minute sampling interval, the cD_1 equals to $1*2^1 = 2$ minute interval, the cD_2 equals $1*2^2 = 4$ minute interval and so on. These time ranges can independently sample and restore frequency components of $1/t*2^{j+1}$ by the Nyquist sampling theorem.

B. Statistical Analysis

1. Detection Mechanism

The reconstructed signal is used for detecting anomalies. The postmortem analysis and detectors can rely on data sets over long periods of time. However, real-time detection requires that the analysis and the detection mechanism rely on small data sets in order to keep such online analysis feasible. As explained earlier, smaller data sets raise the possibility of many false alarms. Larger data sets may increase the latency of detection even when online analysis of such data sets is feasible. We took the following



where,
D is the attack duration in wavelet signal
W is detection (DET) window size (=*qt*, *t* is sampling interval)
T is maximum indication time
m is the majority factor for decision (typically, 1/2)

Fig. 8. The timing diagram in detection mechanism.

moving window approach to accommodate faster detection while reducing the false alarms.

At each sampling instance, we construct the correlation signal S(n). We consider *p* samples, S(n-p+1), S(n-p+2),…, S(n-1) and S(n) for the computation of DWT at the sampling point *n*. We call *p*, the *analysis (DWT) window*. And we consider q $(\leq p)$ samples, S(n-q+1), S(n-q+2),…, S(n-1), and S(n) for anomaly detection. We call *q*, the *detection (DET) window*. To reduce false alarms due to instant noise, we use a majority over the detection window to detect anomalies. If *q/2* or more of the samples in the detection window are above the anomaly threshold, we consider that an anomaly is detected at the sampling point *n*. This majority detector requires that traffic exhibit anomalous behavior over several sampling points (at least *q/2* in a window of *q* samples) for a successful detection. When *q* is large, we can keep false alarms low. However, larger *q* also increases latency of anomaly detection since such a majority function delays the attack detection for at least *q/2* sampling periods. As a result, attacks smaller than *q/2* sampling periods are likely to be not detected. We illustrate these timing observations in Fig. 8 and (3.2).

$$\begin{aligned} & D \geq mW \\ & T = D + (1 - 2m)W, \qquad \text{for } 0 < m \leq \min(1, D/W) \\ & |W - D| \leq T < W + D \end{aligned} \tag{3.2}$$

Based on Fig. 8, the detectable attack duration time is the half of the detection widow width at least when *m* is ½. Detection indication signal, T, expands from *mW* to *D+(1-m)W*. According to selected *m* value, the period of *T* is changeable between |W-D| and W+D. And the latency of detection is *mW*. The empirical results, however, show the variable latency depending on the attack strength and threshold level.

2. Selective Reconstruction in DWT

We simulate two classes of attacks based on persistence, namely the first 3 attacks are ON/OFF styled attacks and remaining six attacks are persistent attacks. Our postmortem analysis allows the administrator to choose the timescales over which attacks/anomalies detection is desired. The network operator can analyze the traffic successively at different sampling times or choose to analyze the traffic at multiple timescales at the same time. Because of the time-scaled decomposition of the wavelets we are able to detect changes in the behavior of the network traffic that may appear at some resolution but go un-noticed at others.

The first three attacks described in (*,I,*) have an ON/OFF timing of 3 minutes. This signal could be effectively detected by only the 1st coefficient of the DWT in case of 1-minute sampling period. If the network administrator concentrates over 30-minute duration attack signal, he/she can select a higher-level coefficient (for example, cD_5), for detecting designated attacks instead of cD_1.

The last six attacks expressed in (*,P,*) are persistent attacks. Attacks last for 1 hour at a minimum. It means that we could choose the cD_5, cD_6 and cD_7 levels among all the coefficients for reconstruction that are equivalent to 32 minutes, 1 hour 4 minutes and 2 hour 8 minutes respectively, in the case of 1-minute sampling interval.

Our approach lets the network operator select the reconstructed levels over which the anomalies need to be analyzed. Our approach allows the administrator flexibility to analyze the anomalies at multiple timescales simultaneously. To demonstrate the feasibility of our composite approach, we focus on the evaluation of our scheme over the nine types of attacks discussed earlier. In order to detect these attacks, we extract only the 1st, 5th, 6th and 7th levels in decomposition and reconstruct the signal based only on coefficients at these levels.

3. Thresholds Setting Through Statistical Analysis

We develop a theoretical basis for deriving thresholds for anomaly detection. When a random variable X(t) possesses mean μ and variance σ^2, we can express Chebyshev's inequality in terms of the number of standard deviations from the mean:

$$P(|X-\mu| \geq k\sigma) \leq 1/k^2 \tag{3.3}$$

The Chebyshev inequality can suggest the lower bound of confidence level, however, the inequality does not take into account the actual distribution and therefore it is often rather loose. If we assume that the reconstructed signal has a normal distribution, we can design suitable analysis and detection techniques to detect anomalies with high confidence while reducing the false acceptances. Study in [11] has shown that Gaussian approximation should work well for aggregated traffic if the level of aggregation in the number of traffic sources and observed time scales is high enough such that individual

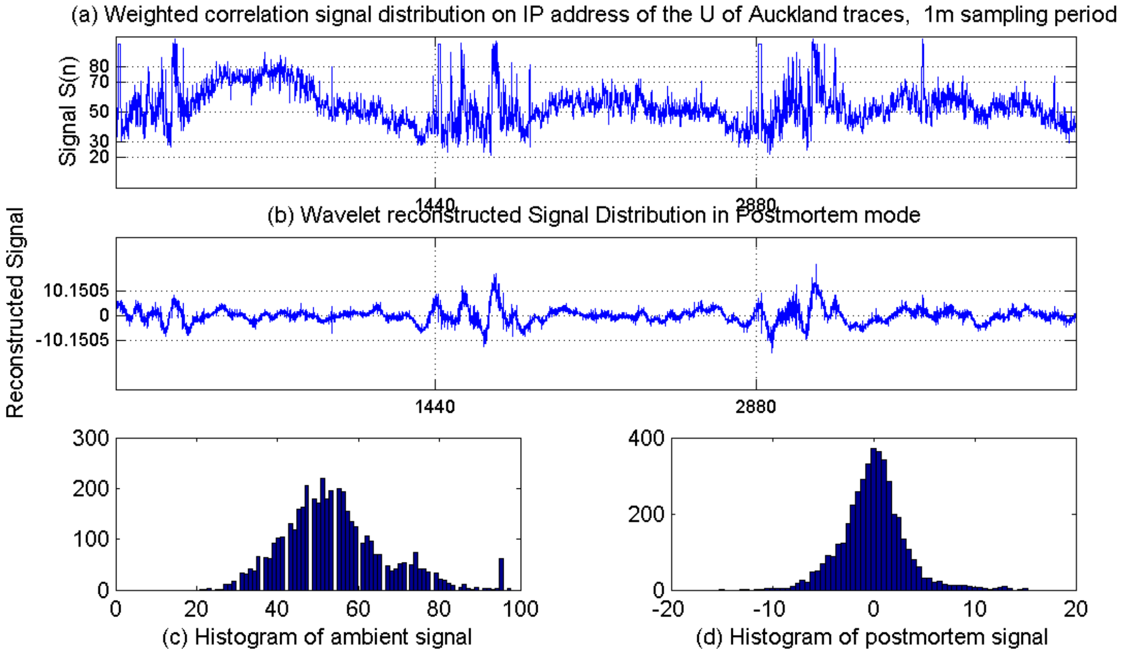


Fig. 9. The distribution of the ambient traces without attacks.
(a) and (c) are distribution before wavelet transform, (b) and (d) are distribution after wavelet transform.

sources are swallowed due to Central Limit Theorem. Our data sets satisfy the necessary criterion for the minimal level of such an approximation.

To verify our methodology, we select only some levels of the DWT decomposition of the ambient trace free of attacks and reconstruct the signal based on those levels. We then look at some statistical properties. Fig. 9(d) shows the histogram of the reconstructed signal of the ambient Auckland-IV traces in postmortem mode. The postmortem transformed data without attacks have mean 0 and standard deviation 3.38 as shown in Fig. 9(b). We verify normality of the Fri/Sun data in Table 4 through the Lilliefors test for goodness of fit to a normal distribution with unspecified mean and variance [56]. The postmortem transformed data have a normal distribution at 5% significance level, namely $X \sim N(0, 3.38^2)$. The original weighted correlation data fail to pass the null hypothesis of normality; however, the DWT transforms it into normal distribution. By selecting some of the levels through selective reconstruction, we have removed some of the features from the signal that were responsible for the non-normality in the original signal.

When we set the thresholds to –10.15 and 10.15 respectively, these figures are equivalent to $\pm 3.0\sigma$ confidence interval for random process X.

$$P(\mu - 3.0\sigma < X \le \mu + 3.0\sigma) \approx 99.7\% \tag{3.4}$$

This interval matches 99.7% confidence level by (3.4). With such thresholds, we can detect attacks with error rate of 0.3%, which can be expected as target false alarm rates.

a. Statistical Consideration of Threshold: Wide-Sense Stationary

If statistical parameters of network traffic, such as mean and standard deviation, are stationary distributed under given traffic, thresholds of specific day could be applied to other days. We gather the 4-week traces and analyze their statistical summary measures. Table 4 shows the distribution in other days. Suppose that the wavelet reconstructed signal is random process X(t) and the sequence of each week is a sample path. We can conclude the X(t) of these traces is WSS in postmortem analysis from the following: (i) X(t) is practically mean-ergodic, i.e., the ensemble average is not dependent on time, (ii) X(t) has an approximately similar standard deviation ($R_X(0)^{1/2}$) over time, as observed in each sample path from Table 4, and (iii) autocorrelation function $R_X(t,t`)$ is a function of time difference t-t` regardless of sample path as observed in Fig. 10. Moreover X(t) could be regarded as white noise from the concentration of average power of X(t) as shown at zeroth lag in Fig. 10. The wavelet reconstructed signal of the weighted correlation with ambient trace could be considered as WSS Gaussian white noise, on the other hand, the attack and anomaly could be considered as random signal of interest.

TABLE 4
THE STATISTICAL MEASURES IN THE SAMPLE PATHS

	Weighted correlation		*Wavelet reconstructed*	
	$\overline{x}$	s	$\overline{x}$	s , $[R_X(0)]^{1/2}$
1st week	53.0	13.5	-0.0	3.3
Mon/Tue	58.1	13.0	-0.2	3.8
Wed/Thr	55.3	13.2	-0.2	3.5
Fri/Sun	48.2	12.4	-0.0	3.4
2nd week	51.5	14.5	-0.0	3.9
3rd week	50.6	14.1	+0.1	3.3
4th week	47.8	13.5	-0.0	4.1

Wavelet transformed signal through selective reconstruction in postmortem mode shows characteristics close to a first-order stationary and ergodic condition for WSS, which measures have a constant mean. And it has an approximately similar standard deviation, which equivalents to the square root value of the autocorrelation function at zero lag ($R_X(0)^{1/2}$) in each sample path due to zero mean.

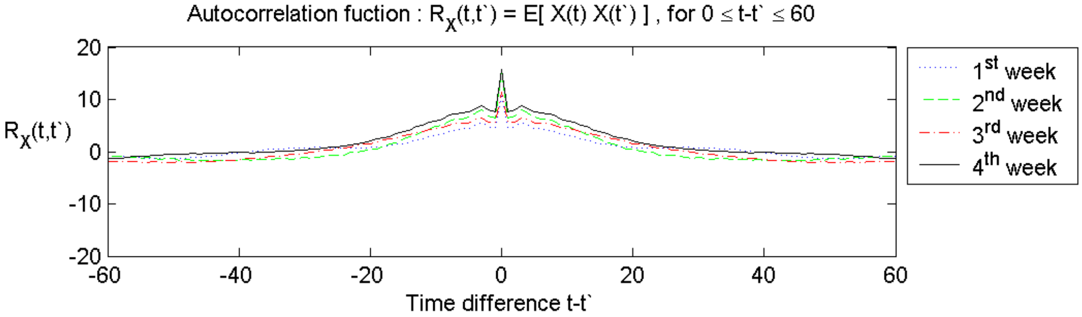


Fig. 10. The autocorrelation function of wavelet transformed signal in the sample paths. Autocorrelation shows second-order stationary condition for WSS, which autocorrelation $R_X(t-t`)$ has an approximately similar distribution in each sample path, therefore it is also a function of time difference t-t`.
And from the impulse characteristics of average power of X(t), namely $R_X(0)$, wavelet transformed ambient signal could be regarded as white noise.

Let's suppose we know only the 1st week data and use the three-sigma limits. The thresholds for postmortem detection are ± 9.9 at 99.7% confidence level. When the very same thresholds are applied in 2nd week traffic, they equal to $\pm 2.54\sigma$ at 99% level. It illustrates that the thresholds could remain approximately the same over several days by virtue of WSS.

C. Detection in Postmortem Analysis

1. Detection of Anomalies Using the Real Attack Traces

Detection results of our composite approach with respect to 7-day KREONet2 traces are shown in Fig. 11. Fig. 11(a) illustrates a weighted correlation signal of IP addresses that is used for wavelet transform with real attacks. Fig. 11(b) is the wavelet-transformed and reconstructed signal in postmortem and its detection results. The actual attacks assail between the vertical lines, and the detection signal is shown with dots at the bottom of the second sub-picture.

A sampling interval is 2 minutes and a sampling duration is 60 seconds. That is to say, we sampled for 1 minute and paused for 1 minute for reducing the processing requirements. A 7-day wide DWT window and a 20-minute wide DET window are used for analysis and detection. To evaluate the reconstructed signal we use $\pm 4.0\sigma$ as statistical threshold in Fig 11(b). Overall, our results show that our approach may provide a good detector of attacks. Let's discuss the detection results in detail.

First 2 attacks attempted to attack a web-server with sequential source port numbers and targeted for port #80. A single source machine sent 48 byte-sized packets to (semi) single destination IP addresses in /24 address which preserved first 3 bytes of IP and randomly changed the last 1 byte. These attacks continued for about 2 to 4 hours. Through traffic engineering, we can identify that many of these attacks finally targeted European hosts via American Abilene networks.

The last attack is the SQL slammer worm attack which generated random IP addresses at a specific port number. A few compromised machines sent a large number

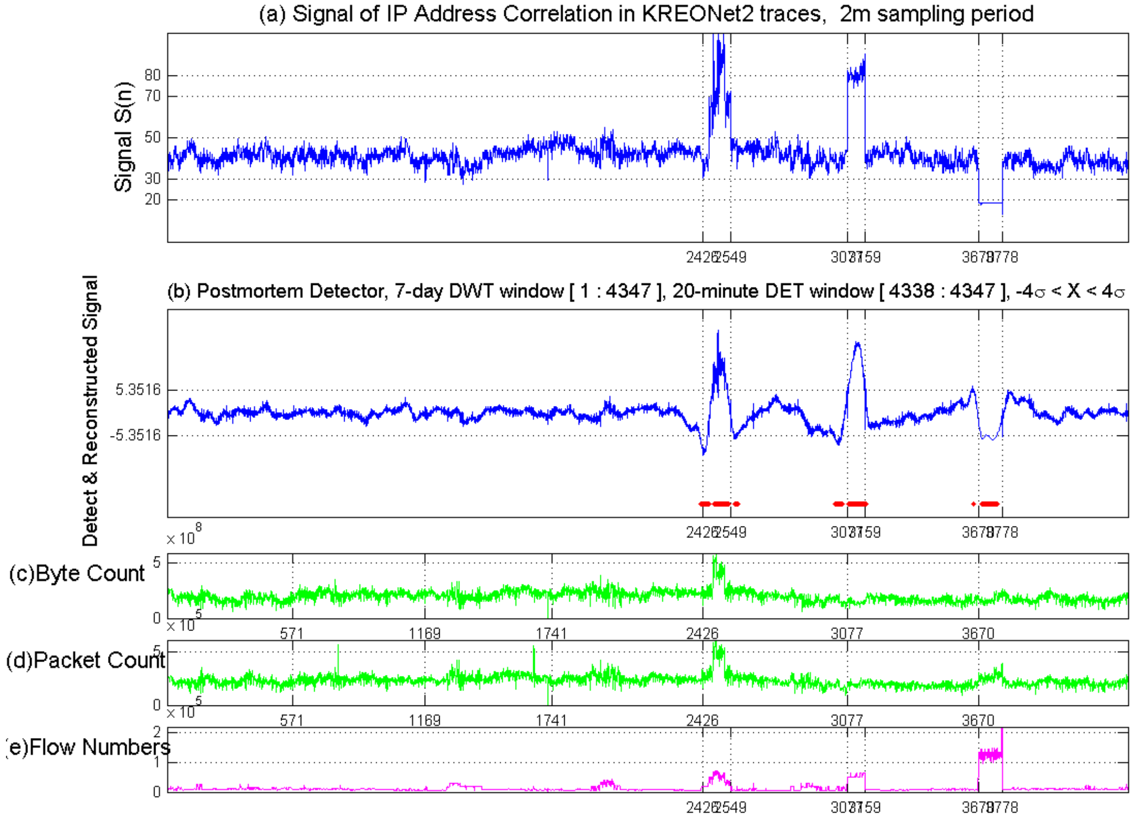


Fig. 11. IP address-based detection results using KREONet2 real attack traces in postmortem.

of 404 byte-sized packets to randomly generated destination IP addresses using #1434 UDP port. This attack persisted for about 3 hours.

Fig. 11(c) and (d) show the traffic volume such as byte counts and packet counts. As the picture shows, except the first attack, the remaining 2 attacks didn't set off any distinguishable variance in volume. It shows that the approach using simply traffic volume itself is hard to appropriately detect the bandwidth attacks.

Fig 12 shows another postmortem result with respect to the USC traces. The left sub-picture illustrates a correlation signal of IP addresses used for wavelet transform, and the right sub-picture is the wavelet- transformed and reconstructed signal in postmortem and

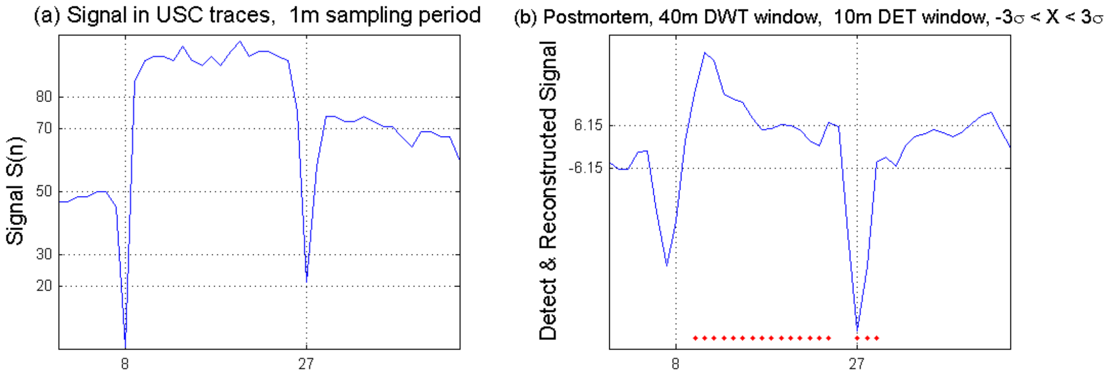


Fig. 12. IP address-based detection results using USC real attack traces in postmortem.

its detection results. Through further analysis, we can identify that many internal compromised machines continued to attack a few external destinations.

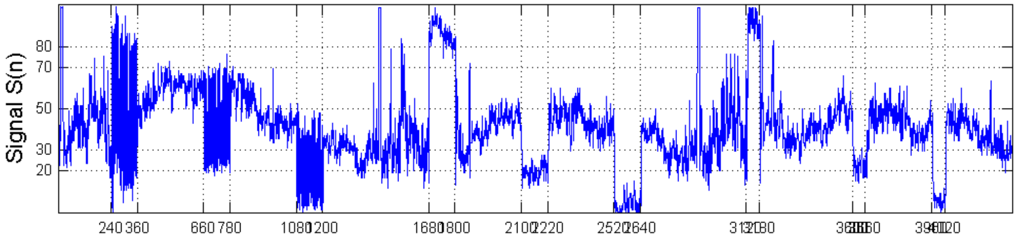
2. Detection of Anomalies Using the Simulated Attack Traces

Detection results on Auckland-IV traces included simulated attacks are shown in Fig. 13. Fig. 13(a) illustrates a weighted correlation signal of IP addresses that is used for wavelet transform with attacks. Fig. 13(b) is the wavelet-transformed and reconstructed signal in postmortem and its detection results.

We employ 3-day traces of addresses collected over a campus access link for these experiments. The sampling interval is 1 minute and the sampling duration is 30 seconds. That is to say, we sampled for 30 seconds and paused for 30 seconds. The simulated 9 attacks are staged between the vertical lines, shown in the figure. A 3-day wide DWT window and a 20-minute wide DET window are used for analysis and detection.

The postmortem analysis uses whole 3-day correlation data all at once. To evaluate the reconstructed signal we use $\pm 3.0\sigma$ as statistical threshold in second sub-picture of Fig 13. The reconstructed signals of first 3 attacks (*,I,*) show an oscillatory fashion

(a) Signal of IP Address Correlation in 3-day Auckland-IV traces, 1m sampling period



(b) Postmortem Detector, 3-day DWT window [1 : 4320], 20-minute DET window [4301 : 4320], -3.0σ < X < 3.0σ

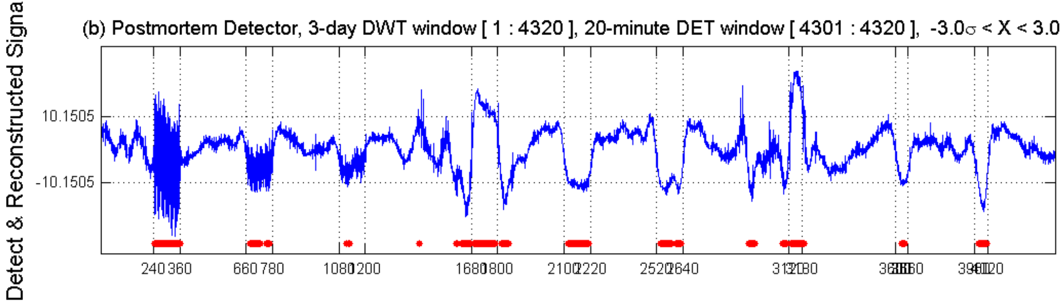


Fig. 13. IP address-based detection results using simulated attack traces in postmortem.

because of their intermittent attack patterns, while the remaining six attacks, namely (*,P,*), give a shape of a hill and a dale at attack times due to persistence.

The attacks on a single machine, especially the 1st attack among every 3 attacks described in (*,*,SD), reveal the high valued correlation which means the current traffic is concentrated on a (aggregated) single destination. Detection signals in the form of dots show that these typed attacks can be detected effectively. On the other hand, the semi-random typed attacks, that is (*,*,SR), and random styled attacks, namely (*,*,R), illustrate low correlations which means traffic is behaving in inconsistent pattern. These attacks can be also captured across attack time. Consecutive detection signals indicate the length of attacks and also imply the strength of anomalies.

Moreover, the detections in the early points of every day, sampling points near 1450 and 2900, turned out to be regular flash crowds included in the original traces.

TABLE 5
THE DETECTIONABILITY OF THE IP CORRELATION SIGNAL AND THE DWT SIGNAL

	conf. level	***DWT***	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	***false positive*** [e]	***false negative***
1.0σ	68 %	IP[a]	.[c]	.	.	.	.	.	.	.	.	5	0
		DWT[b]	.	.	.	.	.	.	.	.	.	6	0
1.5σ	86 %	IP	.	.	.	.	.	.	.	.	.	4	0
		DWT	.	.	.	.	.	.	.	.	.	5	0
2.0σ	95.5 %	IP	.	x[d]	.	.	.	.	.	.	.	3	1
		DWT	.	.	.	.	.	.	.	.	.	3	0
2.5σ	98.5 %	IP	.	x	x	.	x	.	.	x	.	1	4
		DWT	.	.	.	.	.	.	.	.	.	2	0
3.0σ	99.7 %	IP	.	x	x	.	x	.	.	x	x	0	5
		DWT	.	.	.	.	.	.	.	.	.	0	0
3.5σ	99.95 %	IP	x	x	x	.	x	x	.	x	x	0	7
		DWT	.	.	x	.	.	.	.	x	.	0	2
4.0σ	99.99 %	IP	x	x	x	x	x	x	x	x	x	0	9
		DWT	.	x	x	.	x	.	.	x	.	0	4

a. IP means the original IP address weighted correlation signal without applying the DWT

b. DWT means the DWT transformed signal

c. . means a detection

d. x means a non-detection

e. False positive is counted a series of relevant signal as 1

3. Effect of DWT

In order to evaluate the effectiveness of employing DWT, we compare the detection results of our scheme employing DWT with a scheme that directly employs statistical analysis of the IP address weighted correlation signal. The anomaly detection results are shown in Table 5. At low confidence levels (below 90%), DWT doesn't offer any advantage. However, when confidence levels of most interest (90% ~ 99.7%) are considered, DWT provides significantly better detection results than the simpler statistical analysis. This clearly shows that DWT offers significant improvement in the detection of anomalies.

D. Detection in Real-time Analysis

1. Individual Reconstruction in DWT

In real-time analysis, the administrator may not have the luxury to selectively analyze the traffic at different timescales since anomalies need to be detected as they occur. Due to this lack of a priori knowledge of timescales of attacks or anomalies, real-time analysis requires analysis of data at all the time scales. Because of these two needs of analyzing data at all timescales, and the need to have lower latencies of attack/anomaly detection, real-time analysis is much more challenging. In order to complete the analysis in a short time (between two sampling intervals), real-time analysis can only focus on small recent data sets. Because the number of the transformable samples is closely connected with the size of DWT window, the maximum allowable levels are restricted at $\log_2 n$, where n is the number of samples. If we want to investigate a specific level j, it requires 2^j samples for reconstruction at least. In our analysis here, we employed the most recent 2-hour data of traffic. Detecting anomalies through all individual levels will have a number of advantages: (i) By setting a high threshold at each level, anomalies can be detected with high confidence, (ii) Depending on network administrator's filtering criteria, he/she can adjust the threshold between accuracy and flexibility as shown in Table 6, and (iii) the attributes of attacks, such as the frequency and pattern, can be straightforwardly determined.

a. Thresholds Setting Through Statistical Analysis

We establish a statistical baseline for ambient traffic as a means for deriving thresholds for anomaly detection. We classify each level of DWT decomposition of the ambient trace and reconstruct the signal based on each level. The statistical parameters of reconstructed signal at each level are independently calculated. These parameters are updated on an appropriate period with current values and old parameters. When we set $-3.0\sigma < X < 3.0\sigma$ confidence interval at each level as Fig. 15, it corresponds to the 99.7% confidence level and error rate of 0.3%.

2. Detection of Anomalies Using the Real Attack Traces

The reconstructed signal of each level is used for detecting anomalies. As explained earlier, because the real-time detection requires small data sets and quicker identification, a 2-hour wide DWT window and a 10-minute wide DET window are used for analysis and detection. A sampling interval of 2 minutes and a sampling duration of 60 seconds are employed. Through the following approach, we accommodate swifter detection while diminishing the false positives.

First, at each sampling instance, DWT of the samples over the last 2-hour window (60 samples with a 2-minute sampling interval) is computed. We carry out a statistical analysis of each level of the DWT signal separately to analyze the signal over all timescales. At each level of the DWT signal, we employ a 10-minute detection window. Second, the detection mechanism is employed in two dimensions: horizontal and vertical. The horizontal dimension checks for anomaly detection in successive time-samples at the same wavelet signal level. The vertical dimension checks for anomaly detection at multiple wavelet signal levels at the same time. When a specific attack continues in regular pattern, it has a strong probability of being captured in specific level. On the other hand, the vertical one represents how many possible attacks at the specific sampling instant are distributed in different timescales. When a certain attack continues in irregular period, it would be captured over various levels simultaneously.

The combination of the horizontal and the vertical evaluation is used for attack detection. The number of the probable attack detectors is counted using the 2-

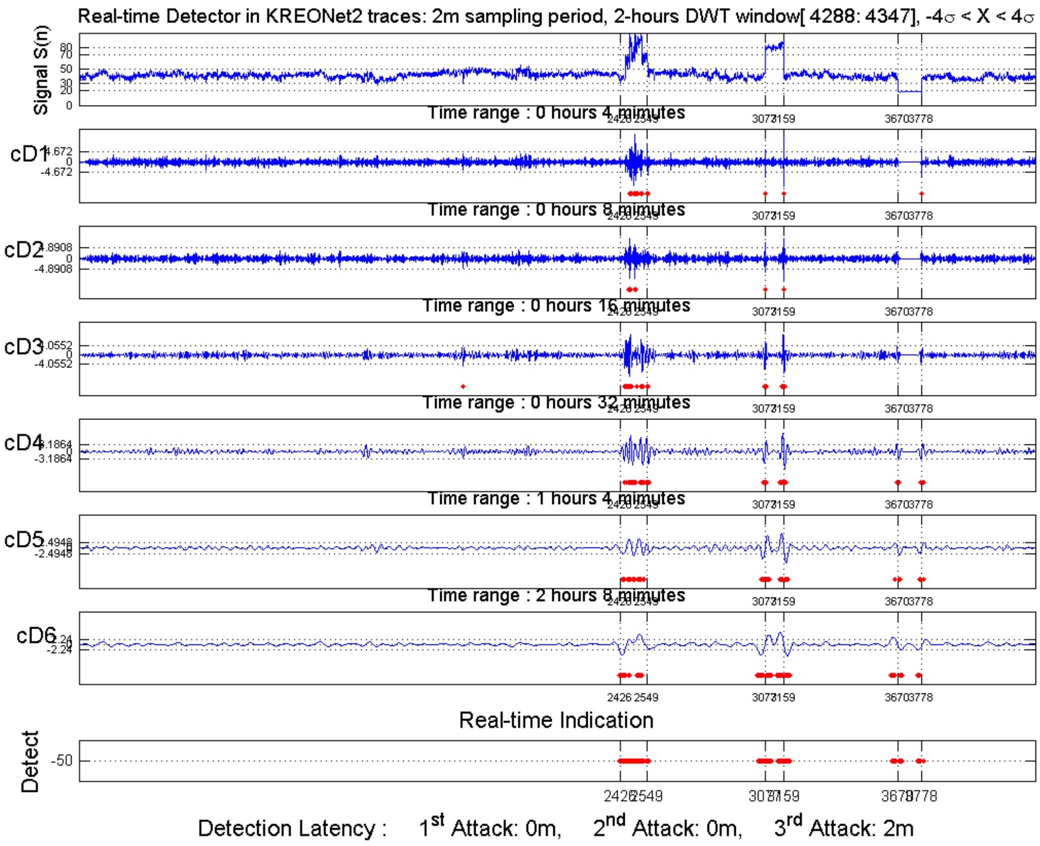


Fig. 14. Address-based detection results using real attack traces in real-time. The signal *S(n)* of the top-most sub-picture is input into 2-dimensional real-time detection window. The cD_1 through cD_6 show the intermediate horizontal detection results at each DWT coefficient level. The real-time indicator in the bottom-most sub-picture shows the final detection results and latencies using the vertical as well as horizontal.

dimensional detection window consisting of the horizontal and vertical components. An attack is detected when the number of detectors exceeds a threshold in the 2-dimensional window. We employed a 2-hour DWT window in 2-minute sampling interval. It can be decomposed up to level 6. The results of our real-time analysis with respect to KREONet2 traces are shown in Fig. 14. The intermediate detection results at each level are shown in upper sub-pictures and the final detection result using 2-dimensional

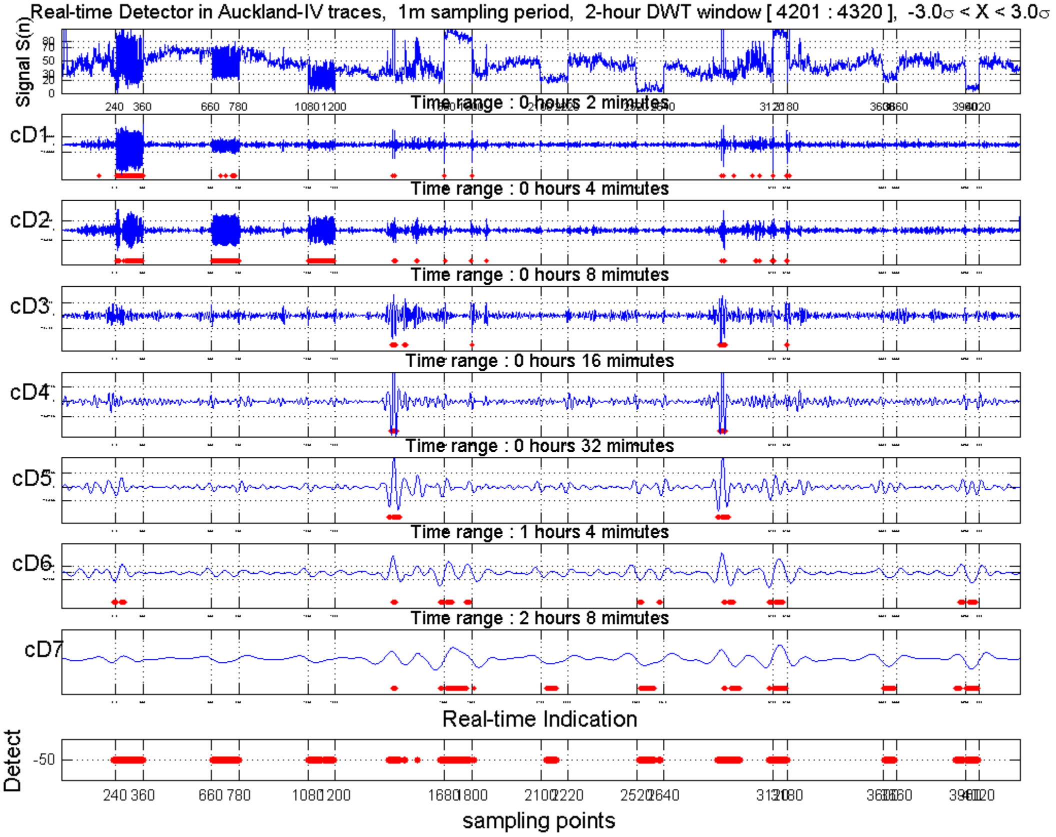


Fig. 15. Address-based detection results using simulated attack traces in real-time. The real-time indicator in the bottom-most detects the originally contained anomalies as well as all kinds of simulated attacks.

window is shown in the bottom-most sub-picture. Our detector achieves acceptable attack detection performance in on-line analysis as well as in off-line analysis.

3. Detection of Anomalies Using the Simulated Attack Traces

We employed a 2-hour DWT window in 1-minute sampling interval. It can be decomposed up to level 7. The results of our real-time analysis are shown in Fig. 15. The DWT signal at each timescales is shown along with the horizontal detector (an anomaly detected over successive samples at the same level). The bottom most picture in Fig. 15

TABLE 6
THE RELATION BETWEEN LATENCIES AND CONFIDENCE LEVELS IN NINE KINDS OF ATTACKS IN REAL-TIME MODE

	confidence level	**1 (2,I,SD)**	**2 (2,I,SR)**	**3 (2,I,R)**	**4 (2,P,SD)**	**5 (2,P,SR)**	**6 (2,P,R)**	**7 (1,P,SD)**	**8 (1,P,SR)**	**9 (1,P,R)**	***false pos.***	***false neg.***
1.0 σ	68 %	0[a]	0	0	0	0	0	0	0	0	11	0
1.5 σ	86 %	0	0	0	0	0	0	0	0	0	7	0
2.0 σ	95.5 %	0	0	0	0	0	0	0	0	0	5	0
2.5 σ	98.5 %	0	0	0	0	0	1	0	0	0	3	0
3.0 σ	99.7 %	0	0	0	0	0	2	0	2	0	2	0
3.5 σ	99.95 %	0	0	1	0	20	9	0	3	2	2	0
4.0 σ	99.99 %	1	0	1	0	X[b]	11	0	5	3	1	1

a. Latency is measured by minute unit
b. X means non-detection

shows the composite detector that employs two-dimensional mechanism discussed earlier. The results indicate that the real-time analysis detects all the attacks along with a few anomalies present in the base signal.

Table 6 shows the overall timing relationship between detection latency and the setting of the confidence level of our attacks in real-time mode. Because the entire DWTed signal is also influence by the latest (attacked) sample, attacks can be detected in low latency such as ‘0’ minute regardless of majority vote. As we expect, the higher the confidence level, the higher the detection latency. When the confidence level is low, many false alarms are incurred because of imprudence of detection; on the other hand, almost all of the attacks can be detected without false negatives. As the threshold is increased, the false acceptance is diminished; however, the false rejection is induced sometimes. According to the network administrator’s security standard, the appropriate confidence level could be established. Even though, our real-time analysis results are promising that attacks may be detected in a few sampling instances, recent studies [2] indicate that worm propagation control measures need to react even faster to be effective. In the future, we plan to develop techniques for swifter identification of these attacks through an interlaced window scheme and multidimensional indicators.

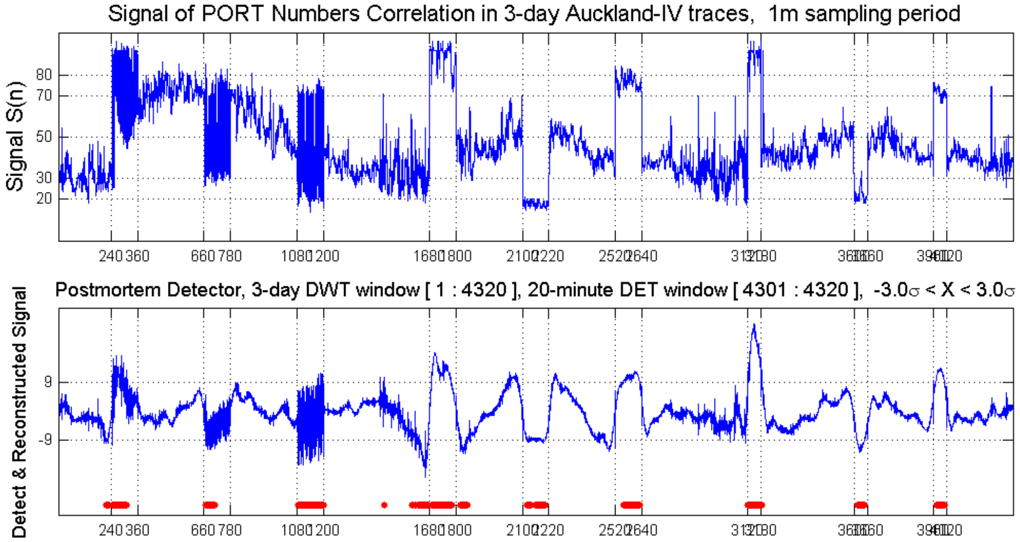


Fig. 16. Port numbers-based detection results in postmortem.

E. Multidimensional Indicators

1. Analysis of Network Traffic by Port Numbers

It seems feasible to carry out a similar correlation and wavelet-based analysis of network packets based on their port numbers. This is particularly motivated by the recent large-scale attacks Code Red and SQL Slammer. Both attacks have been spawned on particular ports exploiting unique weaknesses of end applications. It has been observed that in both cases, an unusually large number of packets were generated on these ports during the attack. We simulate the attacks in Table 3, and repeat the same procedure in postmortem mode as earlier, but now with port numbers as the traffic signal.

Our correlation-based analysis shows marked variations during an attack when we consider port numbers of packets as data. Detection results of our approach are shown in Fig. 16.

The top-most sub-picture illustrates a weighted correlation signal of port numbers that are used for wavelet transform. The second sub-picture shows the wavelet-transformed signal in postmortem analysis and its detection results. The transformed data without attack, which shows approximately normal distribution, namely $X \sim N(0, 3.0^2)$. The -9 and 9 values as thresholds equivalent to $-3.0\sigma < X < 3.0\sigma$ confidence interval. They correspond to 99.7% confidence level. Unlike address-based signal in Fig. 13, it shows the high correlation in (*,*,R) attacks. It means that the packets are focused on the specific (#1434) port. The results indicate that correlation of port numbers over samples of network traffic could provide a reliable signal for analyzing and detecting traffic anomalies. When attacks are staged on a particular port, we find high correlation and when attacks are staged on random ports, we find the correlation to be low.

2. Number of Flows

The number of flows could vary from the norm at the outbreak of network attacks. Through monitoring changes in the number of flows, it is feasible to perceive the anomalies. We used the triple of destination address, destination port and protocol as the definition of a flow. The traces from the NLANR and KREONet2 contained packets in both directions, so we selected packets belonging to only the outgoing direction. Our flow number-based analysis shows marked variations during an attack when we consider the changes in number of flows as data, as shown in Fig. 11(e).

The statistical measures of the wavelet transformed data without attack, which are close to normal distribution with a little heavy-tailed, have 0 as mean and 60.0 as standard deviation.

Moreover, the flow number is uncorrelated with address and port number signals we have investigated so far. Suppose that X is IP address variable, Y is port number and Z is the variable indicating number of flows. Fig. 17 illustrates the cross-covariance function

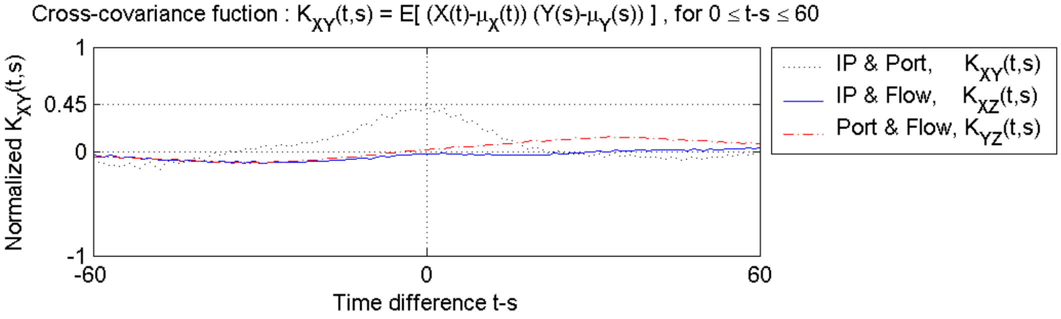


Fig. 17. Cross-correlation function of IP address (X), port numbers (Y) and number of flows (Z).
The Number of flows and either IP address or Port numbers have nearly zero correlation coefficient, which means to be uncorrelated.

among these variables. The cross-correlation coefficient $\rho_{XZ}(0)$ and $\rho_{YZ}(0)$, which are cross-covariance function values when lag is zero, are close to zero. On the basis of normality and uncorrelated property, we can consider that Z is independent of X or Y.

3. Comprehensive Traffic Analysis

Up to now, our work has shown that analysis of addresses, port numbers and flow numbers may individually provide indicators of traffic anomalies. Is it possible to combine several indicators to build a more robust anomaly detector that is less prone to false alarms? We consider two combinations.

First, we design the comprehensive anomaly detector based on a combination of addresses and port numbers. The two-dimensional joint (bivariate) Gaussian density can be determined using in [57], where $\rho_{XY}(0) \approx 0.45$, as follows.

$$f_{XY}(x,y) \approx \frac{1}{18.1\pi} \exp\left[-0.63\left(\frac{x^2}{11.4} - 0.09xy + \frac{y^2}{9}\right)\right] \tag{3.5}$$

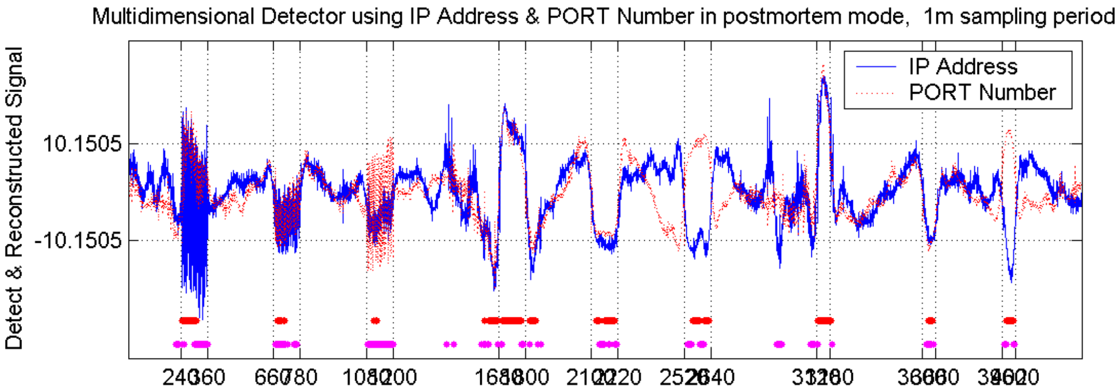


Fig. 18. The multidimensional detection results using IP address and port numbers.

Fig. 18 shows the comprehensive detection results in case of individual 3σ setting as thresholds. The two kinds of dots at the bottom of the picture show detection results. The dots located on top are marked when both the address and port methods detect anomalies simultaneously. The probability of these dots in normal traffic, namely error rate of our detector, is

$$\begin{aligned}&P(|X-\mu_X|>3\sigma_X, |Y-\mu_Y|>3\sigma_Y)\\&=2*\int_{3\sigma_Y}^{\infty}\int_{3\sigma_X}^{\infty} f_{XY}(x,y)\,dx\,dy\\&\approx 0.0125\%\end{aligned} \tag{3.6}$$

And the dots located on the bottom are displayed when only one of the two detection methods detects anomalies. The probability of these dots is

$$\begin{aligned}&P(|X-\mu_X|>3\sigma_X, |Y-\mu_Y|\leq 3\sigma_Y)+P(|X-\mu_X|\leq 3\sigma_X, |Y-\mu_Y|>3\sigma_Y)\\&=4*\int_{0}^{3\sigma_Y}\int_{3\sigma_X}^{\infty} f_{XY}(x,y)\,dx\,dy\\&\approx 0.49\%\end{aligned} \tag{3.7}$$

It can be understood that the above markings imply very high confidence of 99.9875% and the lower dots imply slightly lower confidence of 99.51%.

Second, we can design the comprehensive anomaly detector based on a combination of addresses and flow numbers. Because IP address and flow numbers are independent, where $\rho_{XZ}(0) \cong 0.0$, the joint density $f_{X,Z}(x,z)$ is defined with marginal densities of $f_X(x)$ and $f_Z(z)$ as follows.

$$\begin{aligned} f_{XZ}(x,z) &= f_X(x) * f_Z(z) \\ &\approx \frac{1}{405.6\pi} \exp\left[-0.50\left(\frac{x^2}{11.4} + \frac{z^2}{3600}\right)\right] \end{aligned} \tag{3.8}$$

In case of 3σ setting, we can identify attacks in the comprehensive IP address and flow number detector with error rate of $(0.3\%)^2$. We can improve the accuracy of the detector using X and Y, or Z and either X or Y, or X, Y and Z at the same time as signal.

4. Attack Volume

We carried out similar analysis of traffic to study the sensitivity of our detectors to the relative volume of attack traffic in Auckland-IV traces. We varied the ratio of attack traffic to normal traffic volume from 1:2, to 1:5 to 1:10. The results of this study are shown in Table 7 and 8. The results show that the proposed schemes are effective even when the attack traffic volume as low as 10% of the normal traffic. The latencies for real-time detection get longer with smaller attack traffic volume as to be expected. The results indicate that the DWT analysis of address correlation signal is useful over a wide range of attack traffic volumes.

TABLE 7
THE DETECTIONABILITY OF THE VARIOUS MIXTURE RATIOS OF AUCKLAND-IV TRACES IN POSTMORTEM MODE

	conf. level	*mix. ratio*	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	*false pos.*	*false neg.*
1.0σ	68 %	1 : 2[a]	.[b]	.	.	.	.	.	.	.	.	6	0
		1 : 5	.	.	.	.	.	.	.	.	.	6	0
		1: 10	.	.	.	.	.	.	.	.	.	6	0
1.5σ	86 %	1 : 2	.	.	.	.	.	.	.	.	.	5	0
		1 : 5	.	.	.	.	.	.	.	.	.	5	0
		1 : 10	.	.	.	.	.	.	.	.	.	4	0
2.0σ	95.5 %	1 : 2	.	.	.	.	.	.	.	.	.	3	0
		1 : 5	.	.	.	.	.	.	.	.	.	4	0
		1 : 10	.	.	.	.	.	.	.	.	.	2	0
2.5σ	98.5 %	1 : 2	.	.	.	.	.	.	.	.	.	2	0
		1 : 5	.	.	.	.	.	.	.	.	.	3	0
		1 : 10	.	.	.	.	.	.	.	.	.	2	0
3.0σ	99.7 %	1 : 2	.	.	.	.	.	.	.	.	.	0	0
		1 : 5	.	.	x[c]	.	.	.	.	.	.	2	1
		1 : 10	.	.	x	.	.	.	.	.	.	1	1
3.5σ	99.95 %	1 : 2	.	.	x	.	.	.	.	x	.	0	2
		1 : 5	.	.	x	.	.	.	.	.	.	1	1
		1 : 10	.	x	x	.	x	x	x	.	.	1	5
4.0σ	99.99 %	1 : 2	.	x	x	.	x	.	.	x	.	0	4
		1 : 5	.	.	x	.	.	.	x	x	.	0	3
		1 : 10	.	x	x	.	x	x	x	x	.	1	6

a. Mixture ratio is attack traffic to normal traffic in packet counts

b. . means a detection

c. x means a non-detection

TABLE 8
THE DETECTION LATENCY OF THE VARIOUS MIXTURE RATIOS OF AUCKLAND-IV TRACES IN REAL-TIME MODE

	mix. ratio	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	*false pos.*	*false neg.*
1.0σ	1 : 2	0[d]	0	0	0	0	0	0	0	0	11	0
	1 : 5	0	0	0	0	0	0	0	0	0	9	0
	1: 10	0	0	0	0	0	0	0	0	0	9	0
1.5σ	1 : 2	0	0	0	0	0	0	0	0	0	7	0
	1 : 5	0	0	1	0	0	0	0	0	0	7	0
	1: 10	0	2	2	0	0	2	0	0	0	6	0
2.0σ	1 : 2	0	0	0	0	0	0	0	0	0	5	0
	1 : 5	0	0	2	0	6	0	0	0	0	5	0
	1: 10	0	4	2	0	7	10	0	0	9	5	0
2.5σ	1 : 2	0	0	0	0	0	1	0	0	0	3	0
	1 : 5	0	0	2	0	8	14	0	2	4	3	0
	1: 10	0	5	34	0	24	32	0	2	12	2	0
3.0σ	1 : 2	0	0	0	0	0	2	0	2	0	2	0
	1 : 5	0	0	5	1	8	30	0	5	6	2	0
	1: 10	0	7	38	0	28	32	0	5	16	2	0
3.5σ	1 : 2	0	0	1	0	20	9	0	3	2	2	0
	1 : 5	0	2	34	1	10	40	0	10	9	1	0
	1: 10	0	8	40	1	28	X[e]	0	6	20	1	1
4.0σ	1 : 2	1	0	1	0	X	11	0	5	3	1	1
	1 : 5	0	2	40	3	50	X	4	12	12	1	1
	1: 10	0	10	X	1	X	X	0	15	24	1	3

d. Latency is measured by minute unit

e. X means a non-detection

F. Summary

We studied the feasibility of analyzing packet header data through wavelet analysis for detecting traffic anomalies. Specifically, we proposed the use of correlation of destination IP addresses and port numbers in the outgoing traffic at an egress router. Our results show that statistical analysis of aggregate traffic header data may provide an effective mechanism for the detection of anomalies within a campus or edge network. We studied the effectiveness of our approach in postmortem and real-time analysis of network traffic. The results of our analysis are encouraging and point to a number of interesting directions for future research budge.

CHAPTER IV

TRAFFIC IMAGE-BASED ANOMALY DETECTION

This chapter presents NetViewer, a network measurement approach that can simultaneously detect, identify and visualize attacks and anomalous traffic in real-time by passively monitoring packet headers. We propose to represent samples of network packet header data as frames or images. With such a formulation, a series of samples can be seen as a sequence of frames or video. This enables techniques from image processing and video compression to be applied to the packet header data to reveal interesting properties of traffic. We show that traffic images can reveal sudden changes in traffic behavior or anomalies. Using a combination of visual modeling and trace-driven simulation, we evaluate how the design factors impact the representation of dynamic network traffic. In particular, we study the impact of sampling rate and retained DCT coefficients on the network traffic data representation. We show that "scene change analysis" can reveal sudden changes in traffic behavior or anomalies. We also show that "motion prediction" techniques can be employed to understand the patterns of some of the attacks. We show that it may be feasible to represent multiple pieces of data as different colors of an image enabling a uniform treatment of multidimensional packet header data. We compare NetViewer with an IDS tool.

A. Network Traffic as Image

1. Network Traffic

We employ packet header data collected at a network access point for traffic analysis. This data includes source, destination addresses, port numbers, traffic volume in bytes, packets and other useful information. Each sample of data is represented as an image.

For example, a pixel in such an image may represent traffic volume originating from each source address. Similarly, a dot in the image may represent traffic volume in bytes or packets going to a destination, or in a flow between a (source, destination) pair. Similarly, the image may represent the number of port numbers or flows seen between a (source, destination) pair.

Such a representation allows simple visualization of traffic data as each sample is seen as a frame in a video sequence. Traffic data can then be efficiently stored through such techniques as video compression. Multiple pieces of data, IP address, port numbers and flows, can be represented as different colors of an image leading to unified treatment and multidimensional analysis of traffic data.

Image processing and video analysis techniques can be applied to such a representation to decipher patterns of traffic. Scene change analysis could reveal sudden changes in traffic patterns leading to traffic anomaly detection. For example, single source attacking multiple destinations such as worm propagation will be represented by horizontal lines in the (source, destination) traffic volume image. Similarly, a DDoS attack against a single destination would be represented by vertical lines in the (source, destination) image. A portscan attack would be similarly visible in the port number-based images.

2. Visual Representation

We illustrate our approach with specific examples of image generation and analysis. There are several possibilities for generating images over address domain, port number domain, protocol domain etc. and for utilizing various metrics for generating each pixel in such a domain through the use of traffic volume in bytes, packet numbers, number of flows etc. We use packet counts in the address domain in the following example.

For each address, a_m, in the traffic, we count the number of packets, p_{mn}, sent in the sampling instant, s_n. We can define normalized packet count in the sampling point n as (4.1).

$$p(m,n) = p_{mn} / \sum_m p_{mn} \tag{4.1}$$

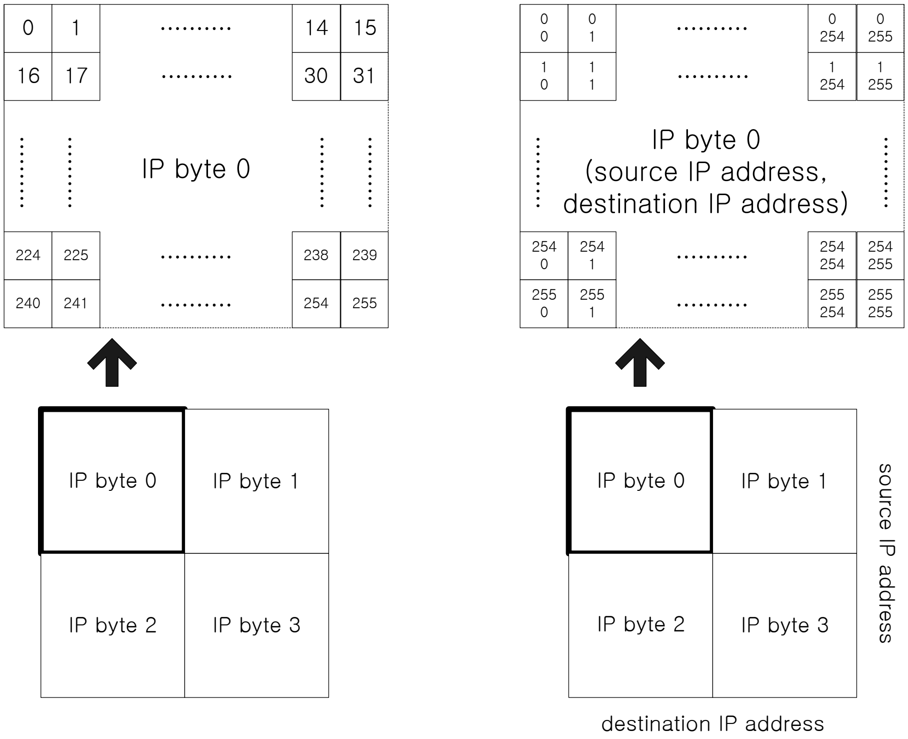
We employ a simpler alternative data structure as explained in chapter II.D for reducing the storage and computation complexity over 2^{32} discontinuous address space from *O(n)* to *O(lgn)*. This data structure consists of 4 arrays "*count[4]*". Each array expresses one of the 4 bytes in an IP address structure. A location *count[i][j][n]* is used to record the packet count for the address j in the i^{th} byte of the IP address in time interval n. The packet counts of the entire traffic are recorded to the corresponding position of each IP address byte-segment and the normalized packet count is quantized and represented in sampling point n as shown in (4.2a).

$$p_{ijn} = \frac{count[i][j][n]}{\sum_{j=0}^{255} count[i][j][n]}, \begin{array}{l} i=0,1,2,3 \\ j=0,..,255 \end{array} \tag{4.2a}$$

$$p_{ijkn} = \frac{count[i][j][k][n]}{\sum_{j=0}^{255}\sum_{k=0}^{255} count[i][j][k][n]}, \begin{array}{l} i=0,1,2,3 \\ j=0,..,255 \text{ in source IP address} \\ k=0,..,255 \text{ in destination IP address} \end{array} \tag{4.2b}$$

Each resultant normalized packet count is converted to corresponding pixel intensity in the image representation of the traffic.

Each byte of the IP address has 256 entries. We arrange the normalized packet count of the 256 entries of the each byte in to a 16-by-16 square for visual representation at the sampling point. Due to 4-byte structure of IP address, we have four such 16-by-16 squares as a frame for the source and destination addresses respectively as shown in (4.2a) and Fig. 19(a). Similarly, instead of 16-by-16, with 256-by-256 squares, we can express the normalized values for a (source, destination) pair simultaneously as in (4.2b)



(a) 1 dimension (b) 2 dimension

Fig. 19. The visualization of network traffic signal in IP address.

and Fig. 19(b). Here the intensity (gray-level) of the pixel is directly proportional to the normalized packet count.

The data structure (or image) processes each byte of the IP address independently. This allows classification of anomalous traffic based on a targeted address range, especially useful in a local-preferential spreading approach employed by some worms. However, our approach could introduce errors during the identification of the complete IP addresses of attackers or victims when the segments of different addresses from each quadrant are combined. Apparent patterns like solid lines could not, statistically, result from the simple result of a union of disconnected dots, but reflect the nature of anomalous traffic. In particular, a horizontal line such as in IP byte 3 of Fig. 25(e) is highly likely to be generated through a hostscan of several destinations by a single source machine. If the number of large distinct flows in byte 3 domain is k ($\leq 256^2$) and the flows are distributed uniformly, the probability of generating single dot (on the line) is $k/256^2$ and of forming m consecutive dots is $(k/256^2)^m$ respectively. So the probability of forming the horizontal line from unrelated flows is $(k/256^2)^{256}$.

A horizontal line in such an image indicates that a source is accessing multiple destinations (with proximate addresses), for example during worm propagation. This indicates a hostscan of destination machines by a single source with a high probability. A vertical line indicates that several sources are accessing a single destination. This could indicate the accesses to popular servers (such as google.com) or DDoS attacks being staged from multiple machines (with proximate addresses) on a single destination.

B. Requirements for Representing Network Traffic as Images

A study of required parameters for presenting network traffic as images like sampling rate and the number of DCT coefficients retained are required to be studied.

1. Sampling Rates

Gaussian approximation requires a high level of aggregation in both vertical aggregation, i.e., presence of a large number of independent traffic flows, and horizontal aggregation, i.e., working at a sufficiently large time scale [11]. Our data sets satisfy the necessary criterion for the minimal level of such a Gaussian modeling. Work in [5] has shown the possibility of analysis of WSS property in network traffic. If the traffic is rather short-term stationary, we could use the Kalman filter or update the statistical analysis frequently for eliminating the non-stationary effects. Based on these results, if we appropriately select the sampling rate for generating images, we could acquire normally distributed and stationary images. For discriminating current traffic situation

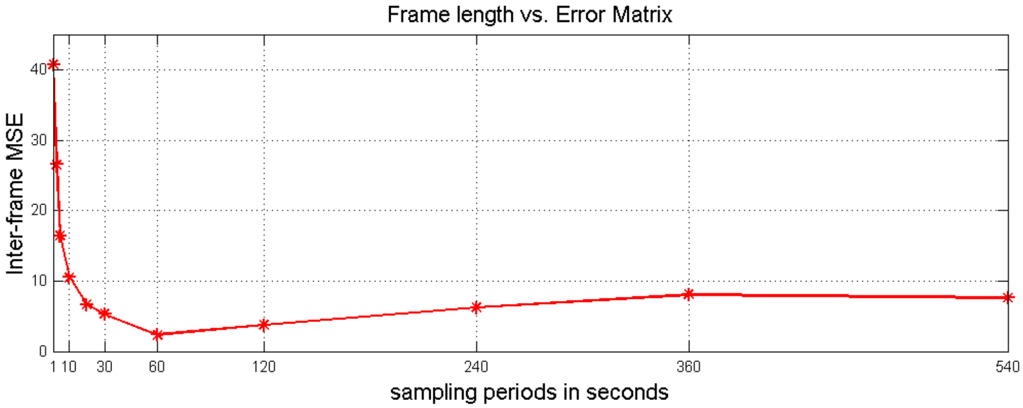


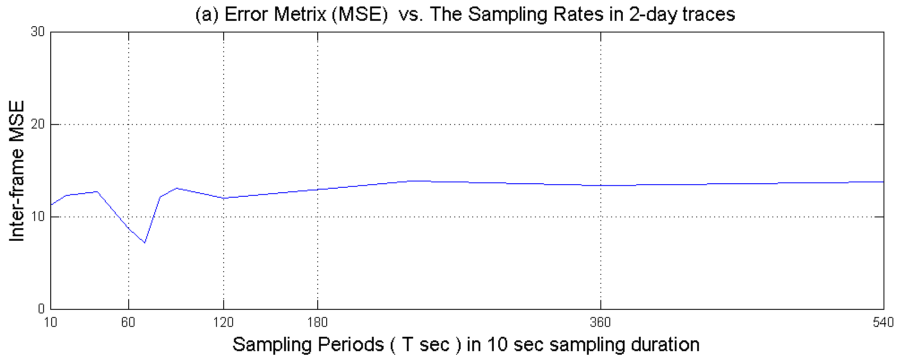
Fig. 20. The relationship between MSE and sampling rates.

based on the stationary property, we should select a sampling frequency for deriving the most stable images [58].

Fig. 20 shows the inter-frame Mean Square Error (MSE) by (4.3) of various sampling intervals. As the sampling interval exceeds 10 seconds, the inter-frame MSE decreases significantly. When sampling intervals are small, the variation of traffic information in successive frames is not negligible. Larger sampling intervals lead to larger latencies for analysis and detection of anomalies. Based on these two observations, we choose 30-60 second samples of traffic data.

$$MSE=\frac{\sum[I(i,j)-I'(i,j)]^2}{N^2},$$
$$\text{for intra-frame}\begin{cases} I(i,j) \text{ is original image} \\ I'(i,j) \text{ is reconstructed image}\end{cases} \tag{4.3}$$
$$\text{for inter-frame, } I(i,j) \text{ and } I'(i,j) \text{ are consecutive images}$$

Fig. 21 illustrates the effects of sampling periods by the type of network traffic. It is sampled only for 10 seconds of every T seconds (10/T sec), where T varies from 10 seconds to 540 seconds. In normal traffic, as shown in Fig. 21(a), the inter-frame MSE remains at a nearly similar level regardless of the sampling periods. It shows that the



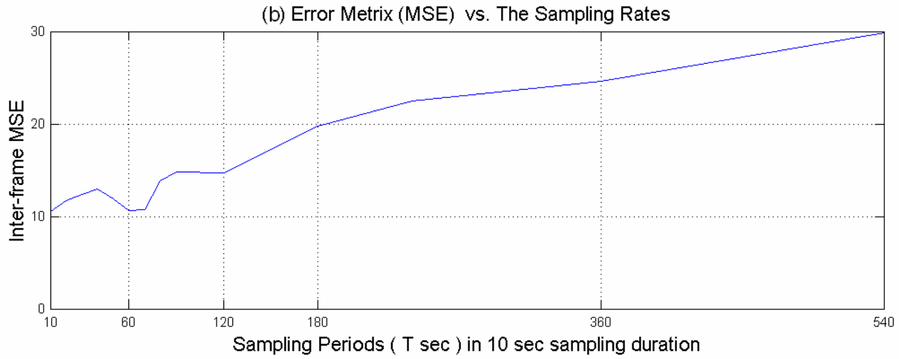


Fig. 21. The relationship between sampling rates and nature of the traffic. The (a) sub-picture shows ambient traffic and the (b) sub-picture illustrates attack traffic.

traffic is stationary in normal times and the selection of the sampling period is not crucial. However, in attack traffic in Fig. 21(b), the MSE increases significantly with increasing sampling periods. It means that the traffic changes dynamically with time in attack times and the sampling period is a critical factor.

2. Discrete Cosine Transform (DCT)

For efficient storage and processing, the images could be compressed. For a 2-dimensional image of (source, destination) domain with 16 bits per pixel, memory of about 0.5Mbytes/frame is required. We could employ image compression using the 2-D discrete cosine transform (DCT) to reduce the requirements.

As the 1st method, we can simply consider one 32-by-32 block DCT. Each frame with four 16-by-16 squares of Fig. 19(a) is treated as one 32-by-32 block for DCT. We transformed the 32-by-32 normalized packet counts all at once using DCT for analyzing the network traffic. The DCT tends to concentrate information, making it useful for image compression and approximation. It is noted that most of the energy is concentrated in the upper-left corner of the DCT matrix. The top upper-left component is called DC component; due to normalization of traffic volume, we always have the same DC components regardless of traffic state. Among 32-by-32 DCT coefficients, we select only 4-by-4 coefficients in the upper-left corner. These coefficients can represent a good approximation of the energy in a sequence.

As the 2nd method, we employ practical 8-by-8 blocks for DCT in the dissertation. We transform the sixteen 8-by-8 blocks of Fig. 19(a) using DCT. The DCT concentrates information in the upper-left corner of the DCT matrices. The inverse DCT can be performed using a subset of the DCT coefficients. Among the 8-by-8 DCT coefficients, we select only the *n* most significant DCT coefficients in the zigzag pattern by discarding coefficients close to zero for compression. We can find out how many coefficients are necessary to create a reasonable approximation of the original traffic image. Fig. 22(a) shows the relationship, the error matrices and the number of retained DCT coefficients in three kinds of different traffic images. Without loosing the properties of traffic, we can choose the suitable number of the DCT coefficients according to system resources. The DCT coefficients could then be quantized, coded, and stored for future analysis.

a. Validity of Intra-frame DCT

The decreasing rates in Fig. 22(a) depend on the characteristics of traffic such as ratio of frequency components. It is worthy to note that the MSE and the decreasing rates are closely related with the variances of pixel intensity in traffic. Transformation Coding Gain (TCG or G_T) by (4.4) measures the amount of energy packed in the low frequency coefficients. So the higher TCG leads to smaller MSE and higher compression. This is useful because if most of information is stored in the first *n* coefficients, then we can reduce the data storage size by retaining only the first *n* coefficients.

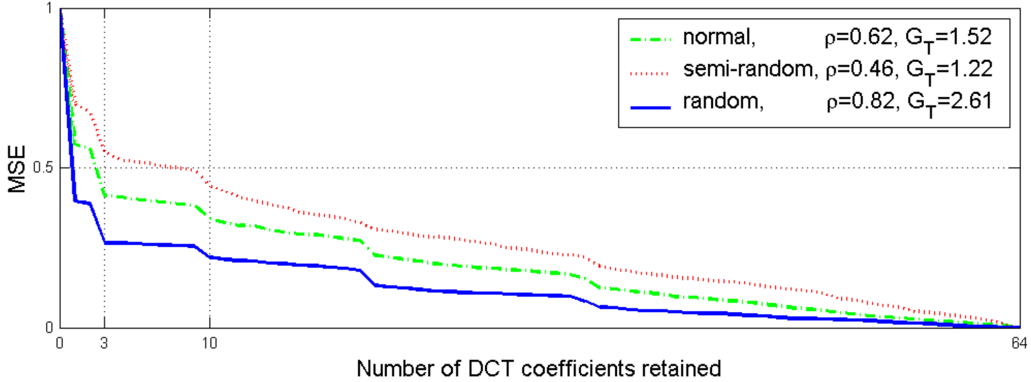
$$\begin{aligned}
&\text{DCT transform matrix } [A]_{i,k}=a_i\cos\frac{\pi(2k+1)i}{2N}\text{, for } i,k=0,\ldots,\text{N-1},\\
&\qquad\qquad\text{,with } a_0=\sqrt{1/\text{N}}\text{ , } a_i=\sqrt{2/\text{N}}\text{ ,}\forall i\neq 0\\
&\sigma_n^2=\text{diagonal elements of } A\Theta A^T\text{, where } \Theta \text{ is covariance matrix}\\
&\qquad\qquad\qquad\qquad\rho \text{ is correlation coefficient}\\
&G_T=\frac{1}{N}\sum_{n=0}^{N-1}\sigma_n^2\Big/\sqrt[N]{\prod_{n=0}^{N-1}\sigma_n^2}
\end{aligned}\tag{4.4}$$

With high TCG, such as in a random traffic image as shown in Fig. 22(a), most of the information can be packed within a few coefficients. With only one coefficient, we can see that DCT transforms of random traffic reduce the MSE by more than 50 %. When we increased the number of coefficients to 3, MSE of most traffic falls by 50%. Traffic images lack significant spatial redundancy and contain more high frequency components than normal images. This results in slower decrease of the Normalized MSE and hence requires cautious consideration of DCT coefficient selection.

b. Inter-frame Differential Coding

In terms of coefficient reduction, if traffic has a low TCG, some significant portion of the original information is spread out into higher frequency coefficients. For improving low TCG due to high components of traffic image, we can exploit the delta of consecutive frame that could be additionally applied for frame coding.

(a) The *n* most significant DCT coefficients vs. Error Matrix



(b) The *n* most significant DCT coefficients vs. MSE for inter-frame delta

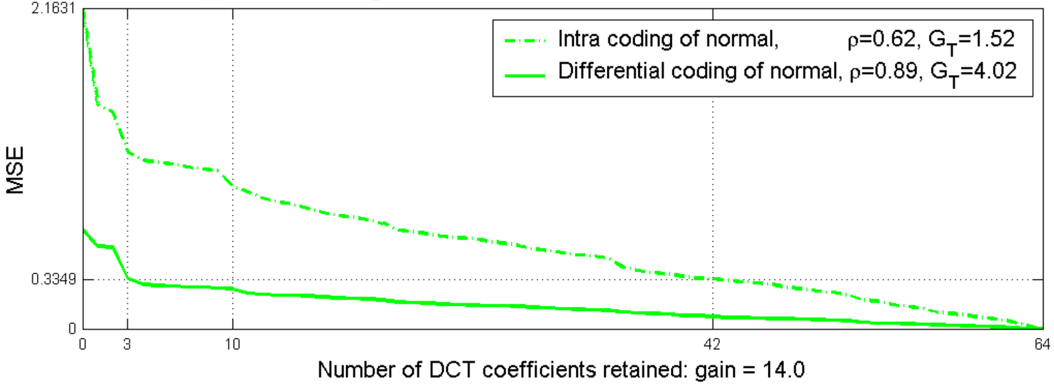


Fig. 22. The relationship between intra-frame MSE and DCT coefficients.

The first ‘Intra’ frame (I-frame) in Group of Pictures (GOP), which consists of *k* frames, is coded without reference to other frames and may be used as a reference. Compression in I-frame is achieved by reducing only spatial redundancy, which is effectively filtering out less significant coefficients, but not temporal redundancy. It can be used periodically to provide access to future frames. On the other hand, the *k-1* ‘Differential’ frames (D-frames) in GOP can use the previous I-frame for differential encoding. Each block in a D-frame can be encoded with delta from an I-frame. By reducing both of spatial and temporal redundancies, D-frames have improved compression compared to I-frame (i.e., non differential coding) as shown in Fig. 22(b).

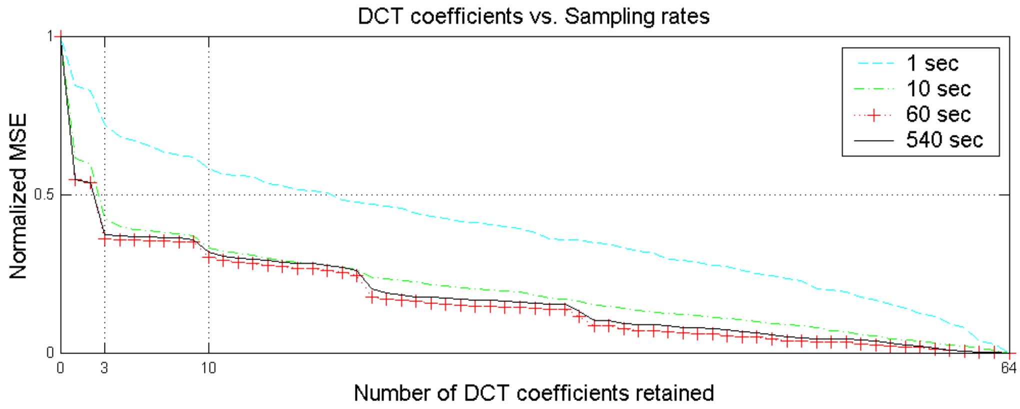


Fig. 23. The relationship between DCT coefficients and sampling rates in normalized intra-frame MSE.

Fig. 22(b) shows the effects of the delta coding through an increased TCG and a gain of about 14.0 compared to the non differential coding in the number of required DCT coefficients. For achieving intra-frame MSE of 0.3349, the differential coding requires only 3 DCT coefficients retained whereas the non differential coding necessaries up to 42 coefficients.

c. Effect of Sampling Rates on DCT Coefficients

The effect of sampling rates on the number of retained DCT coefficients is illustrated in Fig. 23. A sampling rate of 60 seconds maintains the minimum intra-frame MSE over the entire range of retained DCT coefficients. When the sampling period is small, say 1 second, the effect of intra-frame compression is relatively insignificant. When the sampling period is 10 seconds, the intra-frame MSE is close to the optimal value seen at a sampling period of 60 seconds. Based on these results on intra-frame and inter-frame MSE, we conclude that a sample period of 30-120 seconds is a good choice.

C. Visual Modeling Network Traffic as Images

Self-propagating and automated malicious codes usually disturb normal network usage patterns. By observing the traffic and correlating it to the normal states, we can judge if the current traffic is operating in a normal manner. In the case of abnormal traffic, the traffic pattern of network may change and these changes could be exhibited in the visual images.

Automated attacks could be generally classified by their spreading approaches and convergence to the destination into (i) a single target, (ii) semi-random targets (subnet and other prefix-based attacks), and (iii) random targets. Single destination attack can be considered as a special case of a semi-random target case. We look generally at traffic as in normal behavior mode, in semi-random and in random attack modes.

1. Visual Patterns in Normal Network Traffic

Fig. 24 shows the visual measurement of P_{ijn} of the source/destination IP addresses in normal traffic state based on a portion of the KREONet2 traces. The Fig. 24(a) and 24(b) sub-pictures show the standard deviation of pixel intensities in real-time by (4.5) (or 16 MSB DCT coefficients in postmortem by (4.7) after DCT) in source/destination addresses respectively.

The lower 3 sub-pictures visually illustrate the normalized packet counts as outlined in Fig. 19. The aggregate traffic does not form any regular shape due to dispersibility of traffic of various and numerous flows in time and space. The color and darkness of each pixel point up the intensity of traffic of corresponding IP address. During normal traffic, the standard deviations of pixels (or DCT coefficients) of traffic frames maintain the middle level between the two anomalous cases in the Fig. 24(a) and 24(b) as set in (4.8).

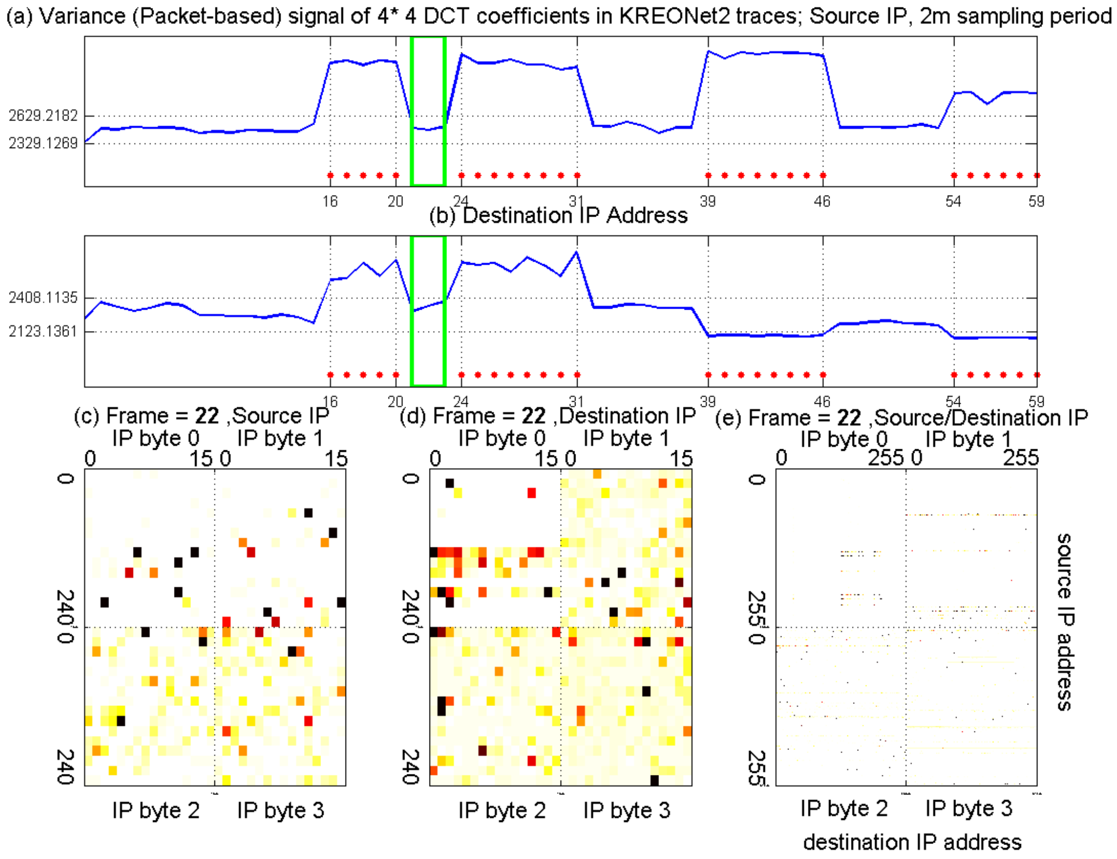


Fig. 24. Visual measurement of normal network traffic.
A green rectangular time-window in (a) and (b) sub-pictures indicates the current sampling points. The bottom red dots in (a) and (b) illustrate the anomaly detection signals and the vertical lines are the periods of actual anomalies. The (c) and (d) sub-pictures show the intensity of network traffic (P_{ijn}) of the source and destination IP addresses respectively. The color of each pixel shows the intensity of traffic at the source or destination, and the descending order of intensity is black, red, orange, yellow and white. The (e) sub-picture shows the intensity of network traffic (P_{ijkn}) of the (source, destination) pair in 2 dimensions simultaneously. The x-axis corresponds to the distribution of the destination IP addresses, and the y-axis does that of the source addresses. In each quadrant, source and destination addresses consist of 256*256 pixels. Over all, the visual measurement shows irregular distribution without a specific pattern. It is noted that the pixel data is actually monochrome (or unidimension) regardless of color representation.

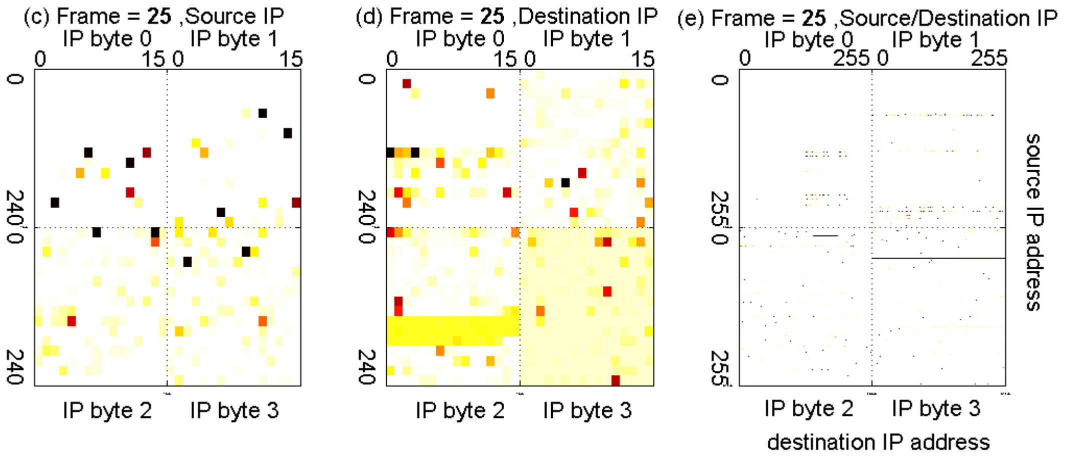


Fig. 25. Visual measurement of semi-random typed attack.

2. Visual Patterns in Semi-random Targeted Attacks

Fig. 25 shows the visual measurement of P_{ijn} of the source/destination IP addresses in a semi-random targeted traffic state. From Fig. 25(d) destination IP addresses, a specific area of IP byte2 is shown in a darker yellow shade. It illustrates that the current traffic is concentrated on a (aggregated) single destination or a subnet. It is observed that this darker portion is shifted with the sampling points during attacks. We estimate the next potential attack using "motion prediction" in chapter IV.D.3. From the 3rd and 4th bytes of Fig. 25(e), it shows that a specific source, i.e., an attacker, monopolizes network traffic, shown in the form of a stripe. During statistical analysis, because the difference in network traffic volume between attackers (or victims) and legitimate users is remarkable, the variance shows much higher values than normal traffic cases.

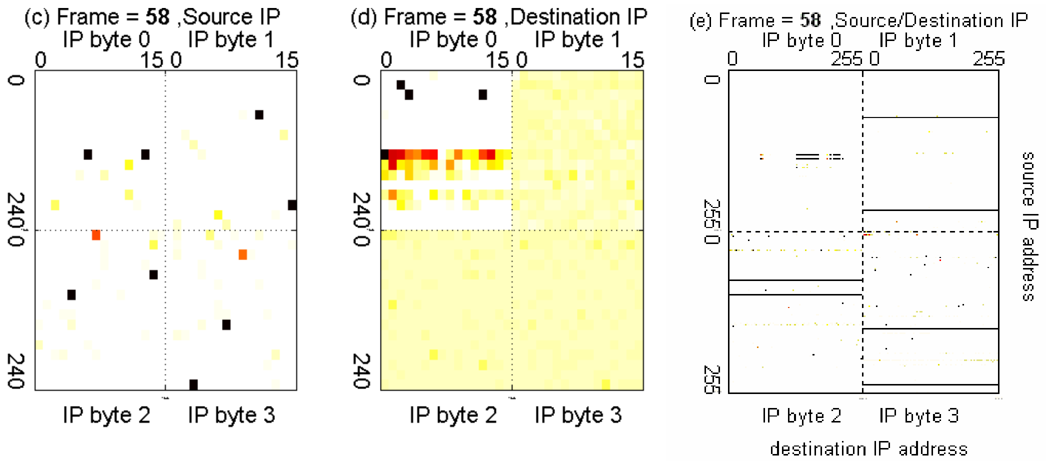


Fig. 26. Visual measurement of (horizontal) random styled attack.

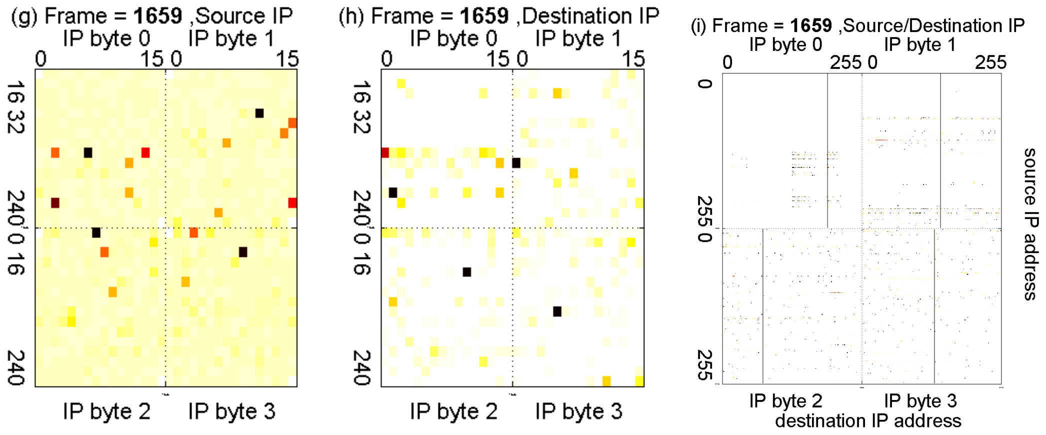


Fig. 27. Visual measurement of (vertical) random styled attack.

3. Visual Patterns in Random Targeted Attacks

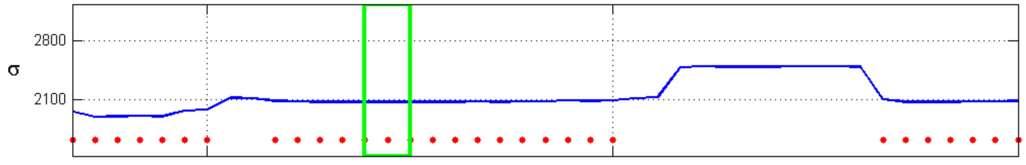
Fig. 26 and Fig. 27 show the traffic during a random attack. From Fig. 26(d), bytes 1, 2 and 3 of the destination address show uniform intensity. It means that, in general, traffic is behaving in an inconsistent pattern and attacks are targeting randomly generated destinations. Because almost all of the destination addresses are exploited in such hostscan attacks, the distribution is highly homogenous such that variances among the IP addresses exhibit lower values relative to normal traffic. From Fig. 26(e), it shows that two specific sources, i.e., two attackers (visible through black pixels in Fig. 26(c) and horizontal lines in 26(e)), scan all possible destinations. Through traffic engineering we verify that two sources, #134.75.100.243 and #141.223.78.151, are staging dictionary mode attacks.

We categorize random attacks into two types.

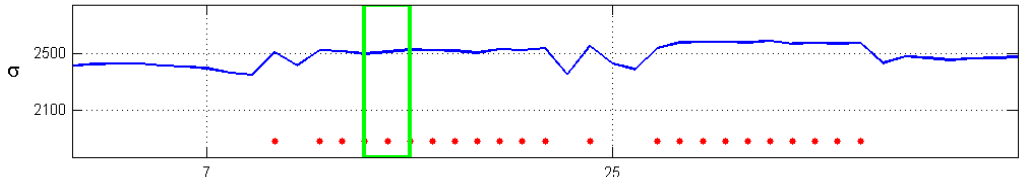
- Horizontal scan - is a scan from the same source IP address aimed at multiple target addresses. It is also known as strobe scan (or worm propagation) which is intended to probe various vulnerabilities of unspecified recipients.
- Vertical scan - is defined as a sequential or random scan from several machines (in a subnet) to a single destination address. Attackers are likely staging DDoS against a specific machine.

4. Visual Patterns in Complicated Attacks

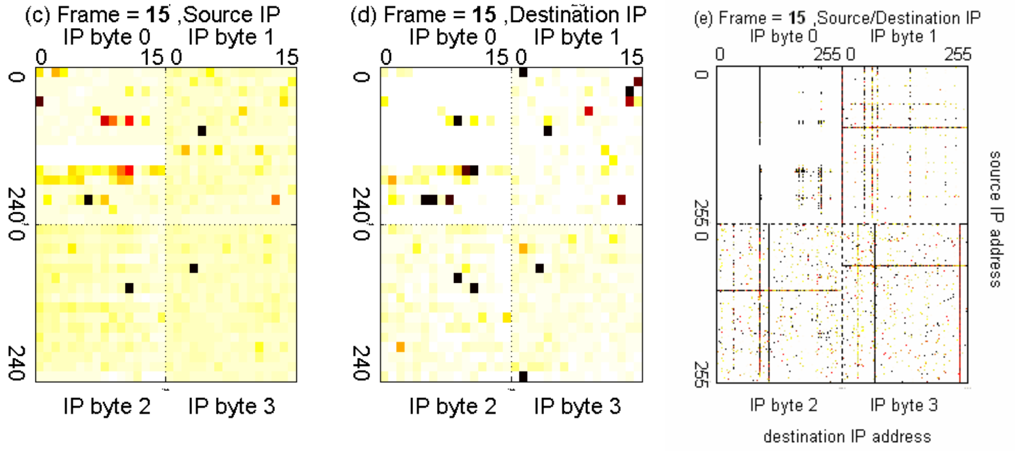
We illustrate complicated and mixed attack patterns using USC traces in Fig. 28. Between the 7th and the 25th frames, randomly generated source addresses attack specific destination addresses. Moreover, from Fig. 28(c) and 28(e), we can infer that a few specific source addresses lead the attack. That is, the (dotted or solid) horizontal line in Fig. 28(e) means specific source scans destination addresses randomly; on the other hand, the vertical line implies randomly generated sources assail specific destination address. This particular trace has a combination of attacks, i.e., a type of worm and DDoS, resulting in multiple indications of possible anomalies.



(a) Variance signal of 4* 4 DCT coefficients in the USC traces; Source IP, 1m sampling period



(b) Destination IP Address



(c) Frame = **15** ,Source IP

(d) Frame = **15** ,Destination IP

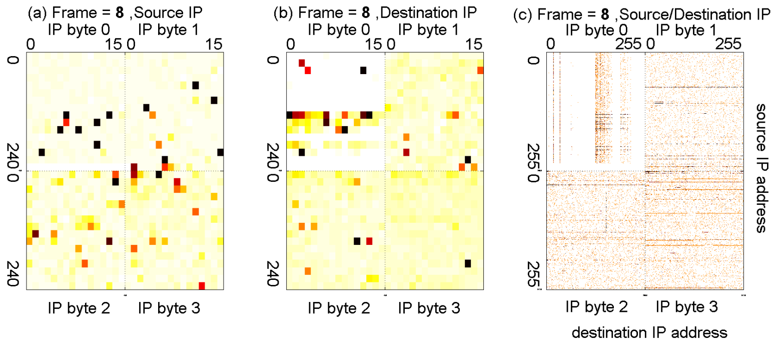
(e) Frame = **15** ,Source/Destination IP

Fig. 28. Visual measurement of complicated network attack.

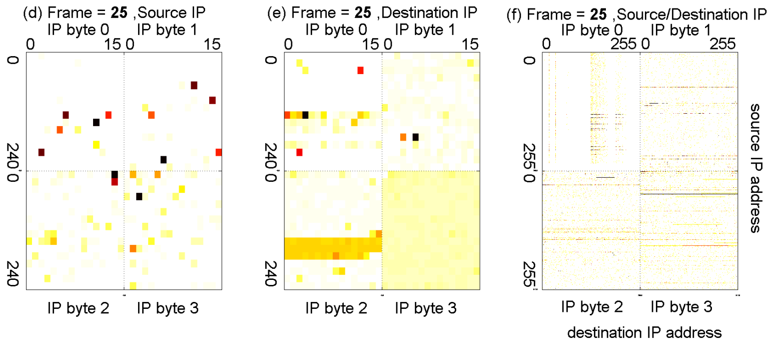
The (c) sub-picture shows intensity of network traffic in a few of the source IP addresses. For example, the IP address 100 in 2nd byte, the 107 in 3rd byte, and the 67 in 4th byte can be considered as suspicious attack sources. From (e) sub-picture, they form the horizontal line in each byte quadrant which means specific source scans all possible targets.

The (d) sub-picture illustrates the concentration of traffic in the destination IP address. For instant, the IP address 1 in 2nd byte, the 89 in 3rd byte, and the 241 in 4th byte can be considered as suspicious attack victims. From (e) sub-picture, they shape the vertical line in each byte quadrant which means randomly generated source targets specific destination.

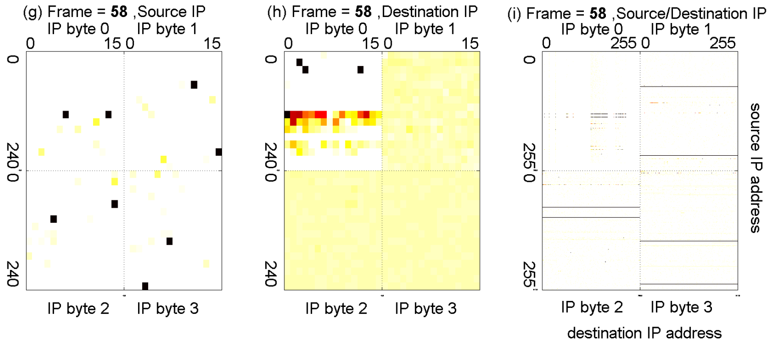
In actual implementation, NetViewer offers these visual measurements as a real-time motion picture. It could help the network operators recognize the traffic transition trends.



[Normal network traffic]



[Semi-random attack traffic]



[Random attack traffic]

Fig. 29. Visual measurement of the *flow-based* network traffic.

5. Flow-based Visual Representation

Similarly, a flow-based visual representation (F_{ijn} and F_{ijkn}) employs the number of flows instead of packet counts over the address domain. The number of flows could vary from the norm at the outbreak of network attacks. Through monitoring changes in the number of flows, it is feasible to perceive the anomalies. We used the triple of source address, destination address and destination port as the definition of a flow.

Fig. 29(a) through 29(c) visually illustrates the normalized flow numbers during the absence of anomalous traffic. The color and darkness of each pixel points up the intensity of traffic (i.e., the number of flows) of corresponding IP address(es). The black-colored lines in Fig. 29(f) and 29(i) explicitly illustrate more concentrated traffic than the orange-colored lines in Fig. 29(c).

Fig. 29(d) through 29(f) visually illustrates the measurement of F_{ijn} and F_{ijkn} in the source/destination IP address domains during a semi-random target attack. From the 3rd and 4th byte of Fig. 29(f), it shows that a specific source, i.e. an attacker, monopolizes network traffic in the form of a horizontal stripe. The attacker employed large number of flows while it sequentially changed the source port and only the 3rd and 4th bytes of victims' IP addresses at a specific destination port.

Fig. 29(g) through 29(i) visualizes the traffic during a random attack. From Fig. 29(h), bytes 1, 2 and 3 of the destination address data structure are shown in darkness. Actually the two compromised attackers employ a large number of flows while they randomly change the victims' IP addresses with specific source/destination port.

6. Multidimensional Visualization

Up to now, we have focused on normalized packet counts and the number of flows in the address domain independently. Besides normalized packet counts and flow numbers, we can use the correlation of the normalized packet counts over the address space for analyzing a variety of aspects of traffic, independently or jointly.[3] The various pieces of

---

[3] We have represented different levels of gray or one of these components of such an image here in color in order to present the data more effectively.

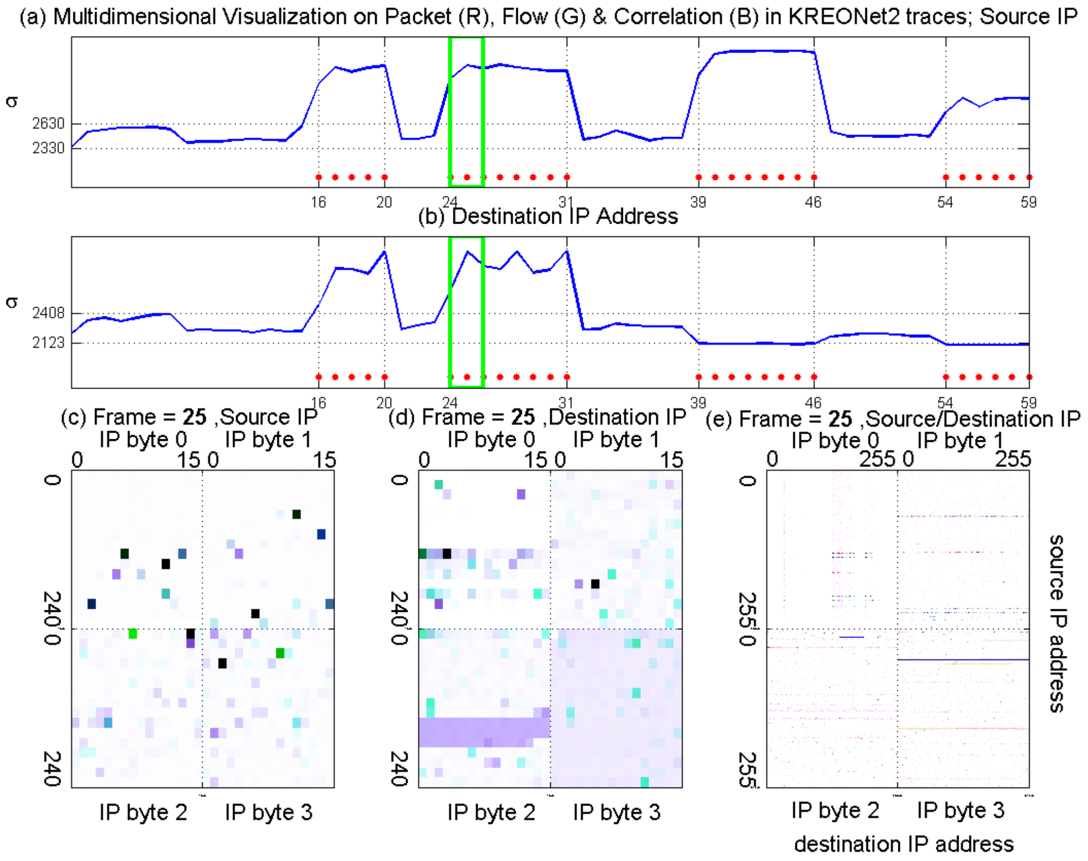


Fig. 30. Multidimensional visualization of semi-random attack.
The normalized packet count in IP address space is represented as the red, the number of flow as the green, and the correlation of the packet count as the blue components of the corresponding pixel image in (c), (d) and (e). The average of 3 kinds of standard intensity deviation is shown in (a) and (b). For effective presentation, all the colors combined make black instead of white using a complement color. In case of semi-random attack, packet (*R*) and flow-based (*G*) components are prominent in comparison with correlation (*B*). So the combined color is close to blue as a complementary color of yellow.

data can be represented as different components (for example, Y, U, V) of an image or different primary colors (R, G, B).

An analysis of the flow-based component of the image is effective for revealing portscan types of attacks. When a flow is defined as the triple of (source address, destination address, destination port), portscan attacks increase the number of flows, when scanning multiple destination ports.

An analysis of the correlation-based image gives an idea of continuance over specific address space. Correlation by (4.10) informs us of flash crowds as well as attacks.

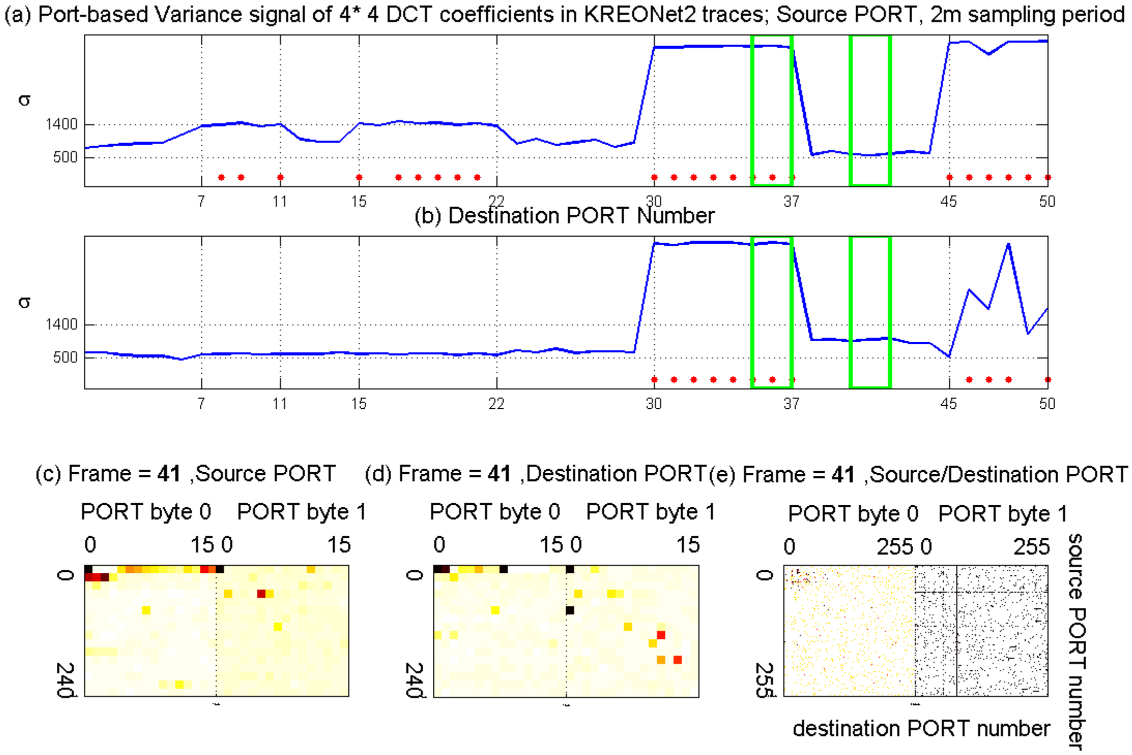
We develop a multi-component image-based analysis of traffic data. The visualization of traffic data with multiple components requires some careful consideration of colors for different pieces of traffic data. Fig. 30 illustrates multidimensional visualization. From the intensity and color of the pixel in the traffic image, we can be informed of the comprehensive characteristics of the traffic in the address domain. We showed the independence between the packet count and the number of flows in chapter III.E.2. With the three distinct traffic signals, we can analyze the traffic properties of each IP address from diverse viewpoints.

a. Visual Patterns in Port Number Domain

So far, we have exploited the packet header information, such as normalized packet counts, the number of flows and the correlation, within the address domain. We could analyze and visualize the packet header information in other domains, for example, the port number domain.

An analysis of the port number-based component of the image can reveal portscan types of attacks. When a machine is the target of a portscan, the distribution of the exploited port numbers would be different from its normal distribution.

As an illustrative example, using the normalized packet counts, Fig. 31 shows the visual measurement of P_{ijn} (or P_{ijkn}) in the source/destination port number domain for detecting portscans. Fig. 31(c) through 31(e) illustrates normal network traffic, where destination port #80 (visible through vertical line in Fig. 31(e)) occupies a large portion of traffic generally. On the other hand, Fig. 31(f) and 31(g) visualize that traffic


(a) Port-based Variance signal of 4* 4 DCT coefficients in KREONet2 traces; Source PORT, 2m sampling period

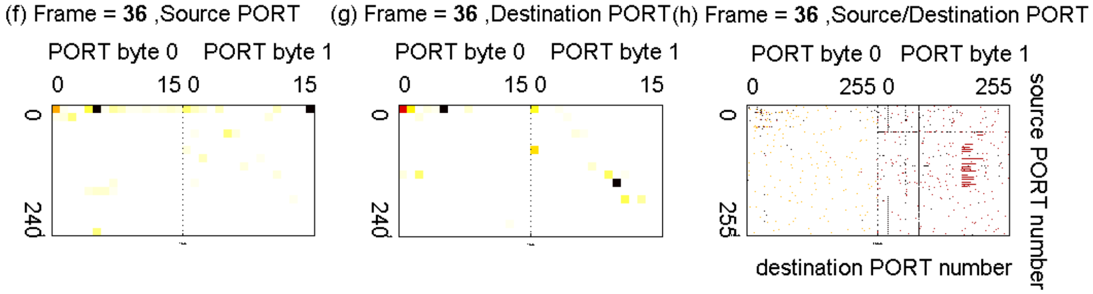
(b) Destination PORT Number

(c) Frame = **41** ,Source PORT

(d) Frame = **41** ,Destination PORT

(e) Frame = **41** ,Source/Destination PORT

[normal network traffic]


(f) Frame = **36** ,Source PORT

(g) Frame = **36** ,Destination PORT

(h) Frame = **36** ,Source/Destination PORT

[attack traffic: SQL Slammer worm]

Fig. 31. *Port-based* visual measurement
The (f) and (g) sub-pictures show the concentration of network traffic from source port #1295 ((05, 15) in (byte0, byte1) to destination port #1434 (05, 154).

concentrates from #1295 of source port to #1434 of destination port in SQL Slammer worm. Fig 31(h) shows the emergency of a novel concentrated attack in destination port in red color.

NetViewer will allow the user to choose the domain(s) of data representation.

D. Anomaly Detection Using Scene Change Analysis

1. Threshold Setting Through Statistical Analysis

We develop a theoretical basis for deriving thresholds for analyzing traffic images and anomaly detection. Based on the possibility of Gaussian approximation and WSS, if the sampling rate is appropriately selected for generating images, for example 1 minute, we could acquire normally distributed and stationary images.

Two different analysis methods, real-time and postmortem, have to be considered.

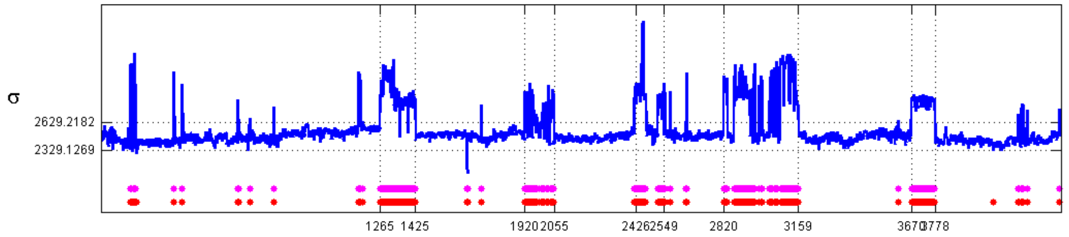
a. Real-time Analysis

Using the variance of pixel intensities in the image as thresholds, we can obtain a feature of the energy distribution of the normalized traffic for real-time.

In real-time analysis, the network operator may not have the luxury to sophisticatedly analyze the traffic since anomalies need to be detected on the fly. Because of these two conflicting constraints of analyzing data uncomplicatedly, and of having lower latencies of attack/anomaly detection, real-time analysis is much more challenging. In order to complete the analysis in a short time, real-time analysis can only focus on the pixel set of the latest frame as (4.5). We can judge the current traffic status by calculating the standard intensity deviation of pixels in each frame as (4.5) and (4.8).

$$\sigma=\left[\frac{1}{1024}\sum_{k=1}^{1024}(x_k-\bar{x})^2\right]^{\frac{1}{2}} \tag{4.5}$$

$$\text{, where } x_k \text{ are pixel intensities and } \bar{x}=\frac{1}{1024}\sum_{k=1}^{1024}x_k$$


(a) Variance (Packet-based) signal of 4* 4 DCT coefficients in KREONet2 traces; Source IP, 2m sampling period

(b) Destination IP Address

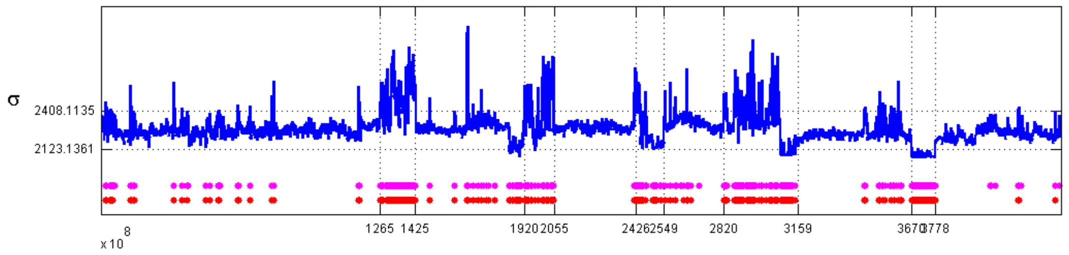


Fig. 32. The results from trace-driven evaluation for detecting attacks in real-time. The magenta dots located on the top are marked when NetViewer declares abnormalities. The Red dots located on the bottom show real anomalies. The dotted horizontal lines in (a) and (b) show the T_H and T_L thresholds based on 3σ method.

The detection signal can be calculated instantaneously upon sampling instants. Fig. 32 shows the detection results from KREONet2 trace-driven evaluation for 8 days. In KREONet2 traces, there are 5 major attacks and a few instantaneous probe attacks.

b. Postmortem Analysis

For post-attack forensic analysis and traffic engineering, the captured images need to be stored. Instead of storing the entire image of each sample, a few DCT coefficients for each sample could be stored as explained in chapter IV.B. We investigate the impacts of using only *n* leading DCT coefficients in detecting anomalies.

The relationship of two stationary random processes can be estimated using cross-covariance function which is the cross-correlation of mean-removed sequence as follows.

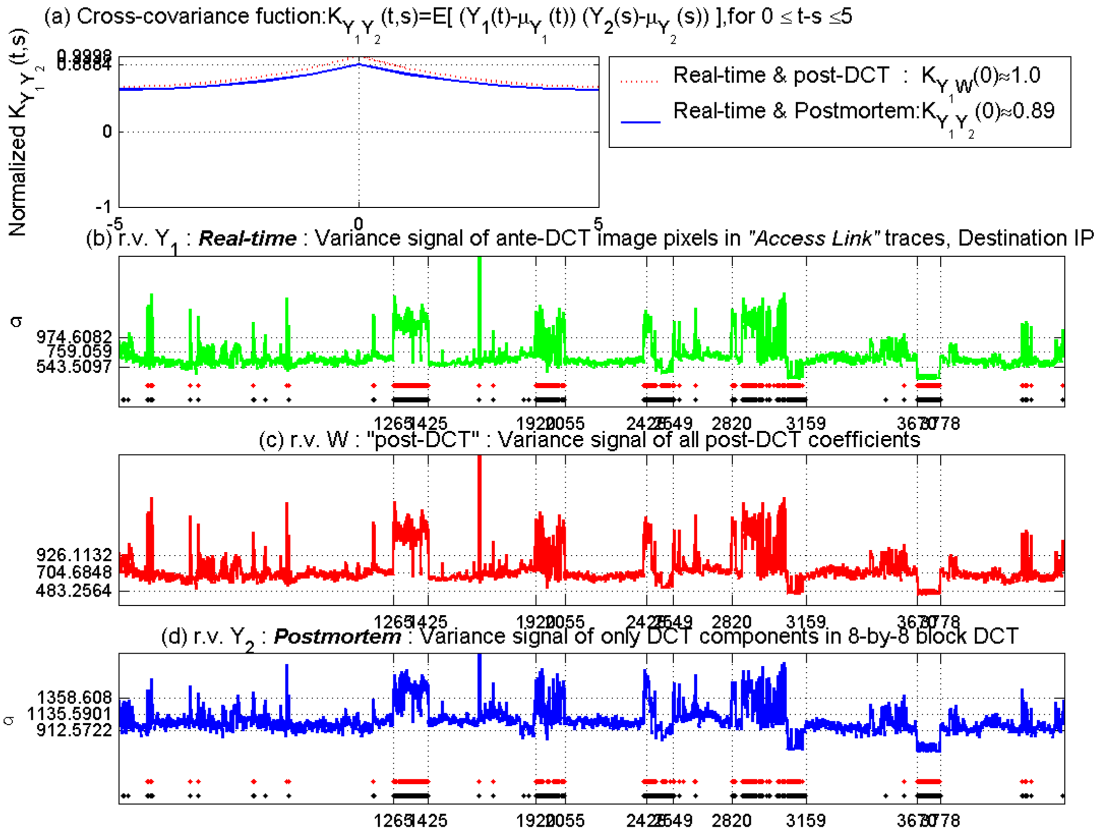


Fig. 33. The trace-driven evaluation results from address-based image signal in destination address for detecting attacks.

The (b) subpicture shows the standard deviation of pixels in the image in real-time analysis. The (c) and (d) show the standard deviation of all of the DCT coefficients and only 1 most significant DCT coefficient respectively during postmortem analysis. The dotted horizontal lines show the T_H, mean and T_L thresholds based on 3σ method. The bottom red dots in (b) and (d) illustrate the anomaly detection results. The black dots located on the bottom show real anomalies.The (a) subpicture shows the cross-correlation coefficients between these signals.

$$K_{XY}(t,s)=E\left[\left(X(t)-\overline{X(t)}\right)\left(Y(s)-\overline{Y(s)}\right)\right] \tag{4.6}$$

, where $\overline{X(t)}, \overline{Y(s)}$ are the mean values and E[.] is the expected value operator

If all of the DCT coefficients were used in the detection, this method would be approximately equivalent to the variance of pixels in real-time (as in Parseval's

Theorem) as shown in $K_{Y1W(0)} \approx 1.0$ in Fig 33(a), and the variance signal in Fig. 33(b) and 33(c). However, by using only the *n* most significant DCT coefficients, we filter the image and are able to focus on the broader characteristics of each image type. We can vary *n* to see how many coefficients we ought to compare to get optimal results in variance. For simplicity, we can use only the top most significant DCT coefficient (*n*=1) in an 8-by-8 block. Using the variance of only DCT component in DCT blocks, we can obtain an approximation of the energy distribution in postmortem analysis as shown in Fig. 33(d).

Instead of performing the computationally intensive task of reconstruction, it is possible to analyze the image by analyzing the DCT coefficients directly. However, by taking only few DCT coefficients, we could potentially perform worse than the reconstruction scheme if the traffic image is not represented well by the retained set. As shown in $K_{Y1\ Y2\ (0)} = 0.89$ in Fig. 33(a) and the approximated variance signal in Fig. 33(d), this technique would classify nearly as well as the reconstruction algorithm.

To model the distribution of traffic in postmortem, we select only ambient trace, free of attacks, as samples and look at some statistical properties. Fig. 43 shows the histogram and normal probability plot of the variations of 16 top most DCT coefficients based on the ambient KREONet2 traces. Using the variations of these leading DCT coefficients for deriving thresholds, we can obtain an approximation of the energy distribution of the normalized traffic within IP address domain as follows.

$$\sigma = \left[\frac{1}{16} \sum_{k=1}^{16} (x_k - \bar{x})^2 \right]^{\frac{1}{2}} \tag{4.7}$$

, where x_k are DCT coefficients and $\bar{x} = \frac{1}{16} \sum_{k=1}^{16} x_k$

c. Thresholds Setting

We set 2 kinds of thresholds, which are the high threshold T_H indicating the traffic is heterogeneously distributed abnormally and the low threshold T_L signifying the network is inordinately homogeneously distributed. We can judge the current traffic status by calculating the standard intensity deviation of the pixels (or DCT components) of each frame as (4.8).

$$\textit{traffic status}\begin{cases} \textit{semi-random}, & \text{if } \sigma > T_H \\ \textit{normal}, & \text{if } T_L \le \sigma \le T_H \\ \textit{random}, & \text{if } \sigma < T_L \end{cases} \tag{4.8}$$

For example, when we respectively set the T_H and T_L thresholds to $\pm 3.0\sigma$ in ambient traffic, these figures of about 2329 and 2629 in source IP address and of about 2123 and 2408 in destination IP address correspond to $\pm 3.0\sigma$ confidence interval for random process X and Y as shown in Fig. 32.

This interval matches 99.7% confidence level by (3.4). With such thresholds, we can detect attacks with error rate of 0.3% (if the signal is normal distributed), which can be expected as target false alarm rates.

2. Anomaly Detection

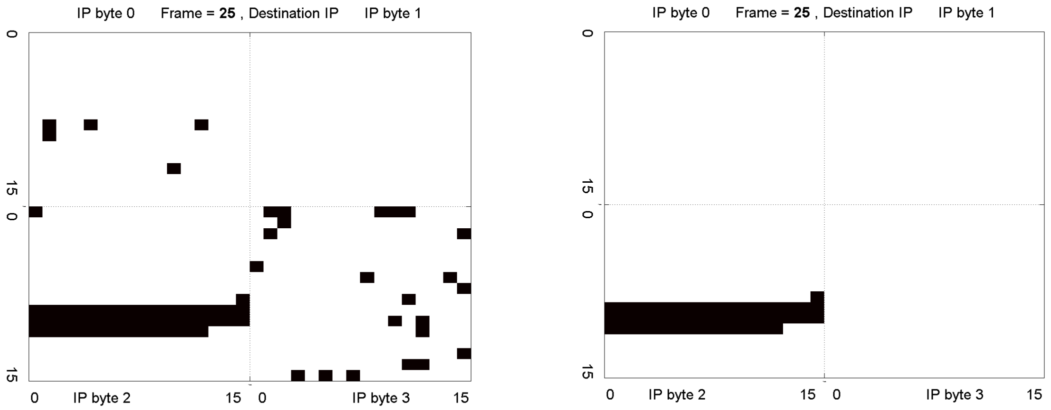
If the variance within frame in current sampling instance is above the T_H or below the T_L, we consider that an anomaly is detected at the sampling point as set in (4.8). The real-time detection requires that the analysis and the detection mechanism rely on small data sets in order to keep such on-line analysis feasible. Our detection signal can be calculated instantaneously at the sampling instants. Results from real trace-driven evaluation in real-time for 8 days are shown in the Fig. 32 and 33(b), and the postmortem detection results are shown in the Fig. 33(d). The major real attacks assail between the vertical lines and the resulting detection signal is shown with red dots located on the bottom of the each sub-picture. This detection signal can be used to alert traffic anomalies to network operators. In KREONet2 traces, there are 5 major attacks and a few instantaneous probe attacks. Through existing traffic reports and detailed traffic analysis, we could also confirm the existence of these attacks. A systematic study of detection power will be evaluated in chapter V.

3. Attack Estimation Using Motion Prediction

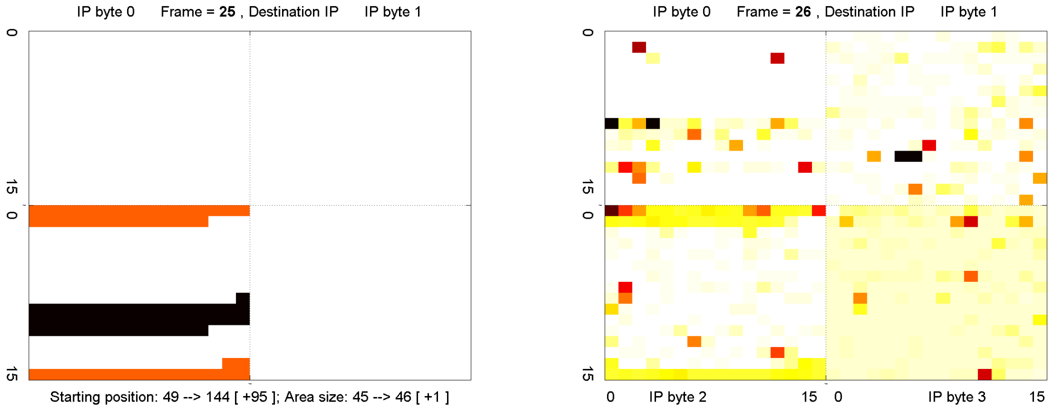
During some attacks, a concentrated attack is circulated on the address space in a semi-random fashion. A semi-random targeted attack could be observed when i) traffic is actually concentrated on a (aggregated) single destination or a subnet, ii) random targeted attacks which have longer period than sampling duration are staged. Using motion prediction, it is possible to expect or anticipate the next set of target addresses in such attacks. We estimate the locations of the next attack using modified motion prediction scheme as explained in the following 3 steps. Fig. 34 illustrates the intermediate results in each sequence based on the destination IP address of the 25th frame in Fig. 25(d).

The 1st step is the complexity reduction. To reduce the subject of investigation, the pixels falling into the following constraints can be excluded from consideration range. By considering only the non-filtered pixels from this pre-processing phase, we can efficiently improve the searching time, avoiding the exhaustive and brute force search of entire address space.

- Pixels below a mean packet count.
- The change in packet counts is remarkable between adjacent pixels using the following *normalized absolute difference (NAD)* similarity measure.



(a) The reduced range of investigation. (b) A continuous block of addresses



(c) The estimated block in different color. (d) The next actual frame image.

Fig. 34. Illustrative procedures show potential attack estimation using motion prediction.

$$\frac{\left|count[i][j][n]-count[i][j\pm1][n]\right|}{count[i][j][n]}\geq 1.0 \tag{4.9}$$

In the 2nd stage, to find a block of addresses, a continuity check is carried out. For improving the continuity, a few non-continuous pixels between continuous pixels, which results from the aperture problem, are considered as a portion of the continuous block. The aperture problem appears in situations where the objects of interest have uniform color. The blocks which are inside the objects do not appear as moving because all of the blocks around them have same color. As a result, the size of the attacking/attacked area can be estimated. In classical image processing, the predefined block size is usually utilized for matching [59]. However, in our method, flexible block size is more desirable due to uncertainty of attack address range.

In the 3rd step, to calculate the quantitative components, the starting positions of attack area and motion vectors for object tracking are calculated. The result of the matching operation is a motion vector with the length of the distance between the positions of the blocks in two consecutive frames. The next potential attack ranges are estimated based on the starting positions and the motion vector length. If the estimation error between the estimated area and the actual area is generated, we could compensate the motion vector.

The results from such an analysis on a semi-random attack are shown in Fig. 34(a) through Fig. 34(d) as representative illustration. Fig. 34(a) shows the non-filtered pixels from the complexity reduction with starting pixel data and Fig. 34(b) shows the identified block of addresses. Fig. 34(c) shows the result of motion prediction (in red pixels) indicating the next set of addresses that may be a target of the attack. Fig. 34(d) is the actual traffic data for the next sampling point, validating the utility of such motion prediction techniques.

4. Processing and Memory Complexity

Our work requires two samples of packet header data $2*P$, where P is the size of the sample data. We also maintain summary information (correlation signals, DCT coefficients etc.) over a larger number of samples S, for statistical evaluation of the current data sample. So, the total space requirement is $O(P+S)$. In our example of address domain analysis, P is originally 2^{32} (2^{64} for 2-dim (source, destination) images), reduced to 4*256 = 1024 (256K for 2-dim images), and *S* is 32*32, reduced to 16. In general, such a domain space reduction will reduce the domain space from 2^n to $(n/k)*2^k$, where *k* (= 8, above) is the folding basis.

Processing requirements are *O(P)* to reduce the sample data to an evaluation signal. Wavelet-based signal analysis requires *O(SlogS)* processing and DCT-based image analysis requires *O(P+S)* processing.

These requirements are sufficiently small that the proposed approach can be implemented in real-time. Sampling periods can be made larger to accommodate available resources. For example, the address analysis requires about 258Kbytes (1K for source/destination domain each and 256K for 2-dim (source, destination) domain) of memory, which can be accommodated in SRAM. For each packet, we require updates of 4 counters (8 memory accesses) per domain, keeping per-packet data-plane cost low.

Our approach can work with traffic records in postmortem or work with more aggregate data upon packet arrival in real-time. It can parse network packet header traces with pcap (packet capture library), Cisco's NetFlow and NLANR formats, for example DAG, Coral, TSH (time sequenced headers).

E. Identification

1. Identification of Attackers and Victims in Byte-segment Level of IP Address

Once anomalies are detected through scene change analysis, we scrutinize the image at higher resolutions for identification purposes. From the position of the (dotted or solid) horizontal/vertical line in the 2-dimesion image, we can be informed of the concentration of the attack. Through line detection algorithm similar to the 1st and 2nd step in the aforementioned motion prediction, we can identify the IP addresses of attackers and victims. Based on the revealed IP addresses, we closely investigate each address on the basis of statistical measurements. In order to quantitatively analyze the network traffic anomalies, we employ an address correlation based on normalized packet count. For computing correlation, we consider two adjacent sampling instants. We can define IP address correlation signal in sampling point *n* as (4.10).

$$C_{ijn}=\frac{count[i][j][n]}{\sum_{j=0}^{255}count[i][j][n]}*\frac{count[i][j][n-1]}{\sum_{j=0}^{255}count[i][j][n-1]},\begin{array}{l}i=0,1,2,3\\ j=0,..,255\end{array} \tag{4.10}$$

We define delta as the difference of normalized packet counts by (4.11).

$$\Delta p_{ijn}=\frac{count[i][j][n]}{\sum_{j=0}^{255}count[i][j][n]}-\frac{count[i][j][n-1]}{\sum_{j=0}^{255}count[i][j][n-1]},\begin{array}{l}i=0,1,2,3\\ j=0,..,255\end{array} \tag{4.11}$$

Correlation is calculated by (4.10), possession ratio by (4.2a), and delta by (4.11). Delta is remarkable at the instant of beginning and ending of attacks. Correlation of each pixel would have probability of $(1/256)^2$ in case of perfectly uniform distribution. We set 3.8% as correlation thresholds, which means the corresponding IP address successively send packets 50 times as many as the evenly distributed address in the average. Once an

```
**********************************************************
[ Time : Tue 10-14-2003 05:12:00 ]
----------------------------------------------------------
Source IP[1]     134.          correlation =  17.48%   possession =  18.77%   delta =   2.50%    S
Source IP[1]     141.          correlation =   4.33%   possession =   3.94%   delta =   0.79%    S
Source IP[1]     155.          correlation =  58.20%   possession =  56.80%   delta =   2.84%    S
Source IP[1]     210.          correlation =   5.66%   possession =   6.51%   delta =   1.60%    S
Source IP[2]          75.      correlation =  17.47%   possession =  18.77%   delta =   2.51%    S
Source IP[2]         110.      correlation =   4.62%   possession =   5.25%   delta =   1.21%    S
Source IP[2]         223.      correlation =   4.31%   possession =   3.94%   delta =   0.78%    S
Source IP[2]         230.      correlation =  58.21%   possession =  56.84%   delta =   2.76%    S
Source IP[3]              7.   correlation =  15.59%   possession =  17.02%   delta =   2.74%    S
Source IP[3]             14.   correlation =  53.99%   possession =  52.31%   delta =   3.41%    S
Source IP[4]                41 correlation =  15.16%   possession =  16.36%   delta =   2.30%    S
Source IP[4]                50 correlation =  52.58%   possession =  50.83%   delta =   3.54%    S
----------------------------------------------------------
Identified No. 1st = 4, 2nd = 4, 3rd = 2, 4th = 2
==========================================================
Destination IP[1]  18.         correlation =   4.37%   possession =   3.88%   delta =   1.01%    S
Destination IP[1] 128.         correlation =   6.08%   possession =   7.01%   delta =   1.75%    S
Destination IP[1] 131.         correlation =  53.65%   possession =  52.33%   delta =   2.67%    S
Destination IP[2]      181.    correlation =  56.03%   possession =  54.00%   delta =   4.15%    S
Destination IP[4]            26 correlation =  3.89%   possession =   3.58%   delta =   0.65%    S
----------------------------------------------------------
Identified No. 1st = 3, 2nd = 1, 3rd = 0, 4th = 1
==========================================================
* Identified Suspicious Source IP address(es)
            134.  75.   7.  41  correlation =  17.48%   possession =  18.77%   delta =   2.50%    S
            141.223.xxx.xxx correlation =   4.33%   possession =   3.94%   delta =   0.79%    S
            155.230. 14. 50   correlation =  58.20%   possession =  56.80%   delta =   2.84%    S
            210.xxx.xxx.xxx  correlation =   5.66%   possession =   6.51%   delta =   1.60%    S
------------------------
* Identified Suspicious Destination IP address(es)
             18.xxx.xxx.xxx   correlation =   4.37%   possession =   3.88%   delta =   1.01%
            128.xxx.xxx.xxx  correlation =   6.08%   possession =   7.01%   delta =   1.75%    S
            131.181.xxx.xxx  correlation =  53.65%  possession =  52.33%   delta =   2.67%
**********************************************************
```

Fig. 35. The detection report for the Fig. 25 of anomaly identification.[4]

attack candidate is identified by correlation, the possession rate and delta ascertain the suspicious byte. We continue this identification process to locate the address responsible

---

[4] For privacy, the IP addresses in Fig. 35 are appropriately sanitized.

for the anomalies over the four byte-segment levels independently as shown in the upper part of Fig. 35.

"*S*" recorded in the last column indicates black listing which is successively identified and refined over recent sampling instances. It could help network operators make a final decision.

2. Identification of Attackers' and Victims' Entire IP Address

Because our data structure processes each byte of the IP address independently, it needs to concatenate the identified entries in each byte into 4-byte whole IP address using the algorithm as shown in Fig. 36.

First, along with our image data representation, we employ 4 independent hash functions, h_1, h_2, h_3, h_4, each with range $\{1,\ldots, m\}$ as a Bloom filter [39]. For each IP address am in the sampling interval, the bits at positions $h_1(a_m), h_2(a_m), h_3(a_m), h_4(a_m)$ in bit vector are set to '1'. Second, for the concatenation of suspicious IP address bytes (to form the complete 4-byte address), we choose the identified most significant bytes of source (destination) IP addresses. We employ the ε-vicinity method in which the two neighboring bytes are concatenated if the measurement difference of the two bytes is less than the tolerable error range. This concatenation procedure continues to the 4th byte. Third, we reduce the false positive rates of the generated 4-byte IP addresses by querying the membership of the addresses through aforementioned Bloom filter data. Through this concatenation, the source and destination addresses of attacks could be identified as shown in the lower part of Fig. 35.

Based on identified attackers and victims, our mechanism can automatically attempt to mitigate the corresponding flows.

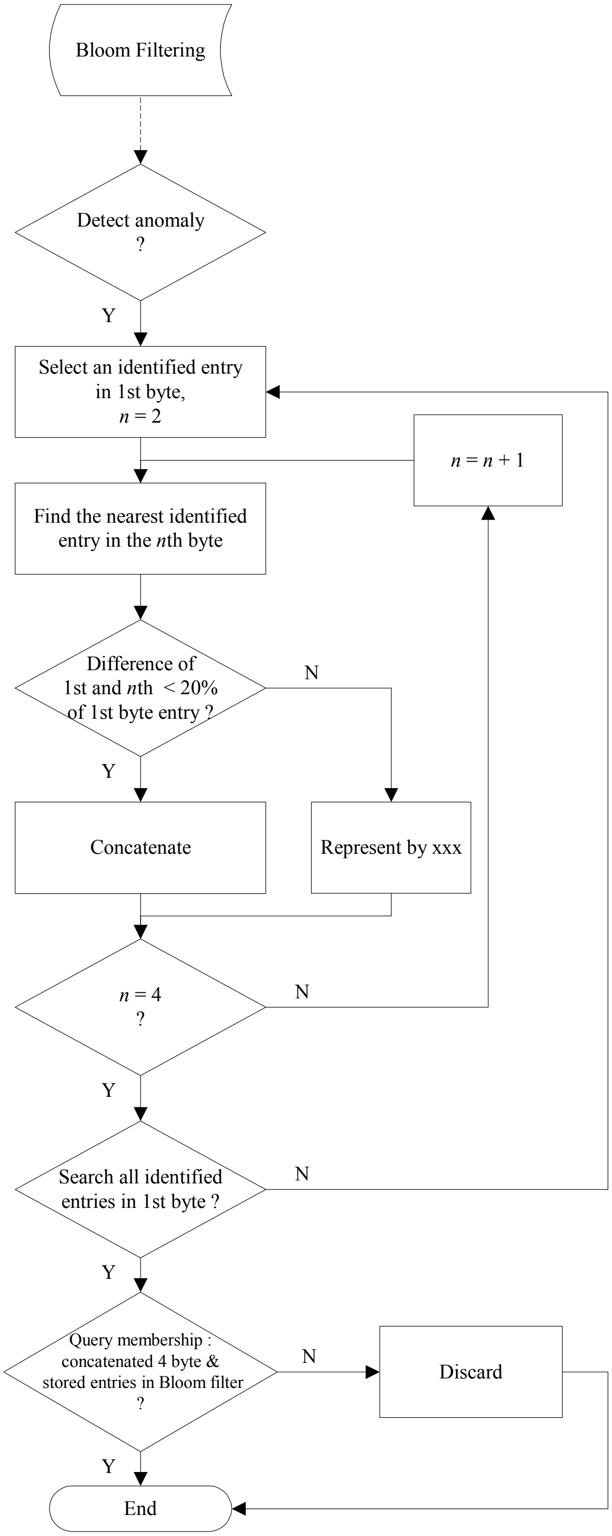

Bloom Filtering
Detect anomaly ?
Y
Select an identified entry in 1st byte, $n = 2$
$n = n + 1$
Find the nearest identified entry in the nth byte
Difference of 1st and nth < 20% of 1st byte entry ?
N
Y
Concatenate
Represent by xxx
$n = 4$?
N
Y
Search all identified entries in 1st byte ?
N
Y
Query membership : concatenated 4 byte & stored entries in Bloom filter ?
N
Discard
Y
End

Fig. 36. The flowchart in concatenation of identification.

F. Comparison with IDS

1. Intrusion Detection System and Snort

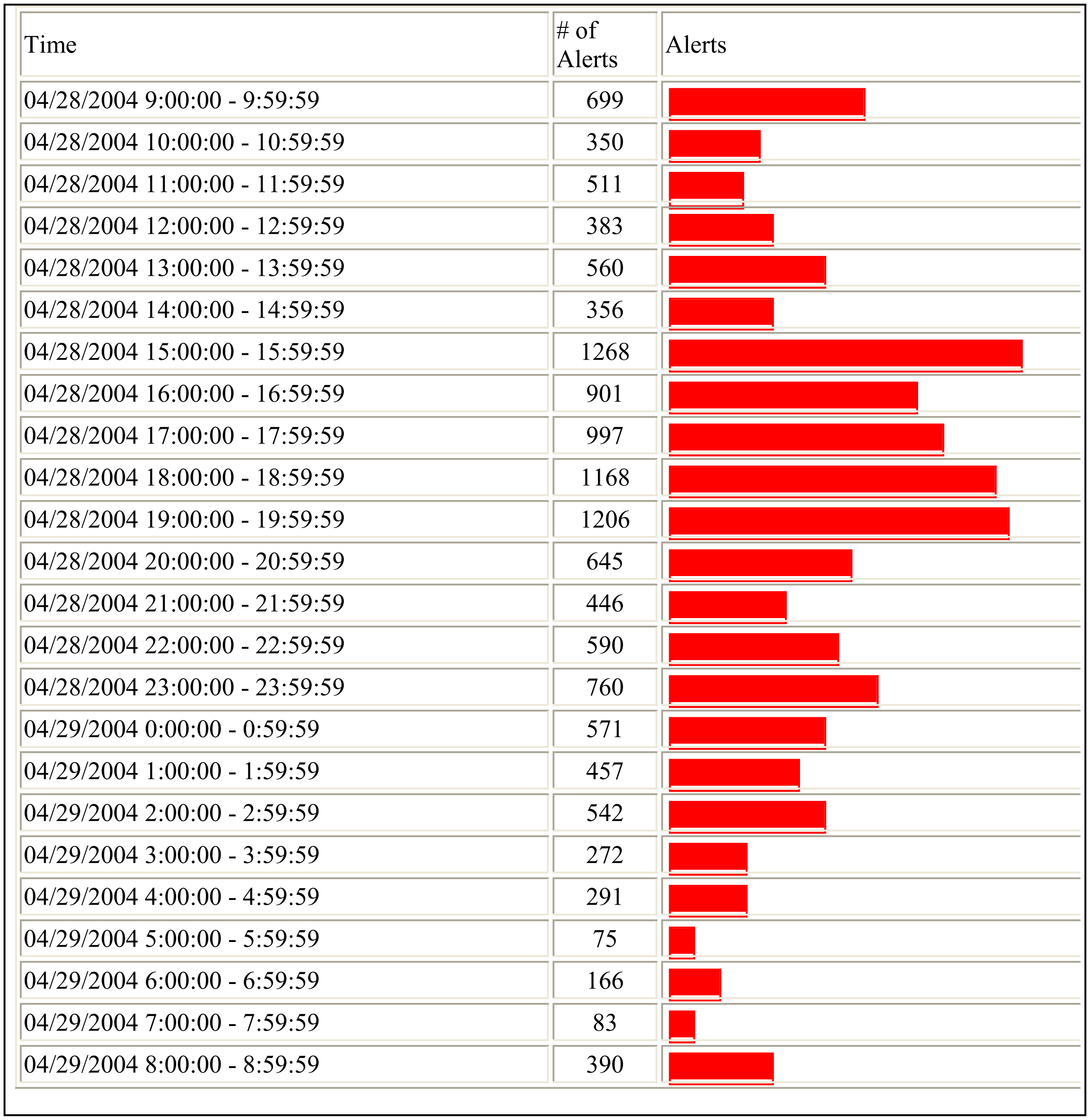
Intrusion detection system (IDS) is an important part of network security architecture and signature detection-based monitoring of network traffic for predefined suspicious activity or patterns is being widely deployed by network administrators. This detection principle relies on the availability of established rules of the anomalous or suspicious network traffic activity. To cope with new attacks, IDS tools are required to be updated with the latest rules. Currently there are a few available freeware/shareware and commercial IDS tools.

We review Snort as representative IDS [17], [18], and compare the properties of Snort and NetViewer. We perform this comparison by running the systems on a live, production network. We report results from a time period which contained a large number of anomalous traffic transactions.

For our experiment, we installed Snort in Texas A&M University network environment, and gathered the detection results of Snort. We evaluate NetViewer on a trace of network traffic analyzed by the Snort system. Our experiment is carried out by capturing 24 hours of data on April 28th and 29th, 2004. After the basic configuration is performed, we turn on the IDS rules, and begin to monitor the Analysis Console for Intrusion Databases (ACID) [60].

2. Overall Results of Snort and NetViewer

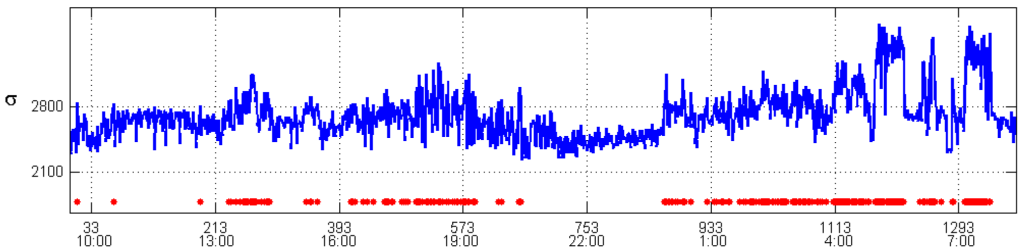
Snort system reported 13,257 alerts distributed over the experiment time period as shown in Fig. 37. Results from NetViewer based on normalized packet counts are shown in the top 2 sub-pictures in Fig. 38. In the trace, it is apparent that there are continuous



Time	# of Alerts	Alerts
04/28/2004 9:00:00 - 9:59:59	699	
04/28/2004 10:00:00 - 10:59:59	350	
04/28/2004 11:00:00 - 11:59:59	511	
04/28/2004 12:00:00 - 12:59:59	383	
04/28/2004 13:00:00 - 13:59:59	560	
04/28/2004 14:00:00 - 14:59:59	356	
04/28/2004 15:00:00 - 15:59:59	1268	
04/28/2004 16:00:00 - 16:59:59	901	
04/28/2004 17:00:00 - 17:59:59	997	
04/28/2004 18:00:00 - 18:59:59	1168	
04/28/2004 19:00:00 - 19:59:59	1206	
04/28/2004 20:00:00 - 20:59:59	645	
04/28/2004 21:00:00 - 21:59:59	446	
04/28/2004 22:00:00 - 22:59:59	590	
04/28/2004 23:00:00 - 23:59:59	760	
04/29/2004 0:00:00 - 0:59:59	571	
04/29/2004 1:00:00 - 1:59:59	457	
04/29/2004 2:00:00 - 2:59:59	542	
04/29/2004 3:00:00 - 3:59:59	272	
04/29/2004 4:00:00 - 4:59:59	291	
04/29/2004 5:00:00 - 5:59:59	75	
04/29/2004 6:00:00 - 6:59:59	166	
04/29/2004 7:00:00 - 7:59:59	83	
04/29/2004 8:00:00 - 8:59:59	390	

Fig. 37. Snort report during 24 hours, April 28 through 29, 2004.

anomalies over almost the entire time period. This detection result agrees with that of Snort.



(a) Variance (Packet-based) signal of 4* 4 DCT coefficients in TAMU traces; Source IP, 1m sampling period

(b) Destination IP Address

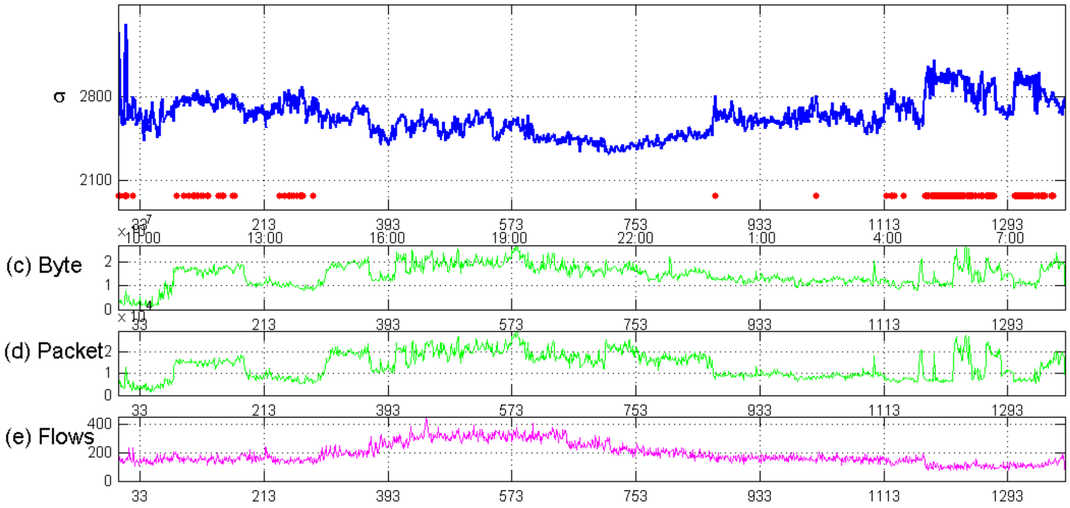


Fig. 38. NetViewer detection for April 28 through 29, 2004.

3. Comparison of Snort and NetViewer

Both Snort and NetViewer detect suspicious anomalies throughout the course of the trace capture. The detection performance can be considered at a similar level.

However, Snort’s identification mechanism is superior in granularity. When coupled with a mechanism such as ACID, Snort can more readily identify the source of malicious activity, and what exactly that activity consists of. Snort provides an easily managed display of IP addresses and port numbers of any suspicious activity. On the other hand, when NetViewer performs the analysis, it reports the suspicious IP addresses and the pattern of abnormality in an aggregated fashion.

```
************************************************************
Sampling count = 1174, Time is 29-Apr-2004 05:01:00
------------------------------------------------------------
Source IP[1]      77.         correlation =  56.54%   possesion =  58.64%   delta =   4.14%    S
Source IP[1]     128.         correlation =  19.38%   possesion =  16.59%   delta =   6.05%    S
Source IP[2]         47.      correlation =  56.54%   possesion =  58.64%   delta =   4.14%    S
Source IP[2]        194.      correlation =  19.38%   possesion =  16.59%   delta =   6.05%    S
Source IP[3]            128.  correlation =  68.09%   possesion =  69.86%   delta =   3.49%    S
Source IP[4]               194 correlation =  68.09%   possesion =  69.86%   delta =   3.49%    S
------------------------------------------------------------
Identified No. 1st = 2, 2nd = 2, 3rd = 1, 4th = 1
============================================================
Destination IP[1] 120.         correlation =  68.09%   possesion =  69.86%   delta =   3.49%    S
Destination IP[2]     120.     correlation =  56.54%   possesion =  58.64%   delta =   4.14%    S
Destination IP[3]          0.  correlation =  70.47%   possesion =  72.90%   delta =   4.77%    S
Destination IP[4]            0 correlation =  62.54%   possesion =  65.19%   delta =   5.19%    S
------------------------------------------------------------
Identified No. 1st = 1, 2nd = 1, 3rd = 1, 4th = 1
============================================================
* Identified Suspicious Source IP address(es)
              77. 47.128.194   correlation =  56.54%   possesion =  58.64%   delta =   4.14%
             128.194.xxx.xxx correlation =  19.38%   possesion =  16.59%   delta =   6.05%
-------------------------
* Identified Suspicious Destination IP address(es)
             120.120.  0.  0    correlation =  68.09%   possesion =  69.86%   delta =   3.49%    S

************************************************************
```

Fig. 39. The identification report on Apr 29, 2004, 05:01:00 by NetViewer.

Snort employs a qualitative analysis and NetViewer employs a quantitative analysis. During our evaluation, Snort missed the identification of many heavy traffic sources. Some flows as shown in Fig. 39, using the BitTorrent system run by one of the users of the network, accounted for about 30% to 60% traffic over certain time periods. However, without the operational rule, Snort did not detect this flow during its life period. However, NetViewer identified this flow as an anomalous event. This demonstrates the utility of measurement-based approaches in detecting previously unknown or undocumented anomalous behavior.

Regarding the computational complexity, Snort looks at the payload of packet as well as the packet header. And currently over 2,400 filter rules are established [18]. NetViewer works on aggregated information from traffic samples. Snort would require more computing resources to be able to match NetViewer performance against heavy traffic.

From these above observations, we feel the two methods could be combined to provide a more complete detection system capable of detecting a wide array of different network security violations.

G. Summary

In this chapter, we have presented an approach which represents traffic data as images or frames at each sampling point. Such an approach enabled us to view traffic data as a sequence of frames or video and allowed us to apply various image processing and video analysis techniques for studying traffic patterns. We have evaluated how the design factors impact the representation of dynamic network traffic. In particular, we have studied the impact of sampling rate and retained DCT coefficients on the network traffic data representation. We have demonstrated our approach through an analysis of traffic traces obtained at three major networks. Our results show that our approach leads to useful traffic visualization and analysis. We have studied detection and identification approaches along multiple dimensions of IP packet header data such as addresses, port numbers, and the number of flows.

We compared our statistical approach with a signature-based IDS system to evaluate the effectiveness of different traffic signals. Our results indicate that measurement-based statistical approaches can be simple and effective and could be combined with IDS approaches to enable more effective monitoring of network traffic.

We plan to study the effectiveness and quantitative evaluation of the image-based analysis of traffic with different packet header data and in diverse networks. We plan to release our tool to general public and combine it with an IDS tool such as Snort in the future.

CHAPTER V

EVALUATION OF DIFFERENT PACKET HEADER DATA AS SIGNALS FOR ANOMALY DETECTION

A number of recent studies have proposed measurement-based approaches to network traffic analysis. These techniques treat traffic volume and traffic header data as signals or images in order to make analysis feasible. In this chapter, we propose an approach based on classical Neyman-Pearson test employed in signal detection theory to evaluate these different strategies. We use both analytical models and trace-driven experiments, and compare the performance of different strategies. Our evaluations on real traces reveal differences in the effectiveness of different traffic header data as potential signals for traffic analysis in terms of their detection rates and false alarm rates. Our results show that address distributions and number of flows are better signals than traffic volume for anomaly detection. Our results also show that statistical techniques can be sometimes more effective than the NP Test when the attack patterns change over time.

Measurement-based tools analyze network traffic to observe statistical properties of traffic. Based on such measurements and some acceptable thresholds on normal network behavior, these tools try to classify traffic as normal or anomalous. Recently, a number of studies have proposed a number of diverse approaches. Some of these studies have treated network traffic as signals, which can be processed and analyzed to detect anomalies. These studies have considered traffic volume [4], [5], [61], [62], number of flows [7], address and port number distributions [37] as potential signals that can be analyzed in order to detect anomalies in network traffic. The traffic headers carry various pieces of information such as port numbers, protocol numbers which can be further treated as potential signals for analysis.

While all these signals have been shown to be useful for analyzing network traffic, so far, no comprehensive study has been carried out about the relative usefulness of these traffic signals. Which signals are more effective for detecting anomalies? Which signals provide low false alarm rates? Different studies have employed different analysis techniques and different traces making such a comparison difficult. In this chapter, we propose to employ classical detection theory-based NP Test to study the relative effectiveness of various pieces of information in traffic headers for detecting traffic anomalies.

Our approach employs the NP Test from the classical detection theory. In our approach, we treat normal traffic as noise and attack traffic as containing the signal along with noise and employ the NP Test to detect attacks in network traffic. NP Test is known to be optimal and hence can be employed to reveal the inherent strengths of the various traffic header signals. We also employ statistical analysis of the traffic data to compare its effectiveness against the NP Test.

Specially, the chapter makes the following significant contributions: (a) provides a comprehensive evaluation of effectiveness of a number of signals derived from network traffic headers, (b) provides an approach based on classical NP Test for evaluating the effectiveness of network traffic signals, (c) provides data from a number of real world traces to compare the different traffic signals and (d) shows that statistical techniques can be as effective or sometimes more effective than the NP Test because of changing attack patterns.

A. Traffic Signals

We broadly categorize anomaly detection schemes into two groups. The two groups are based on the amount of information maintained or kept per sample. The scalar signals keep track of a single variable (such as traffic volume) as a time series. The vector signals keep track of a vector (1 or 2 dimensions in this chapter) of values over a domain of traffic header data (such as addresses, protocol numbers etc.). We briefly outline the various traffic signals below before we outline our approach for evaluation.

1. Scalar Signals

The scalar signals typically employ the traffic volume such as byte counts, packet counts and the number of flows. Many of the commonly exploited malicious attacks are based on high-bandwidth floods, or other repetitive streams of packets. Whether individual packets are malicious or otherwise, flood-based attacks have been used for staging DoS attacks. Work in [4], [5] has analyzed traffic volume, measured in byte counts, packet counts and flow counts, using wavelets to detect anomalies in network traffic. For reasons of scalability, we look at the aggregate traffic volumes at the observation point (not per-flow measurements).

a. Byte Counting

This approach simply counts the number of bytes of traffic seen at the observation point during each sample. The traffic volume in bytes $b(t)$ at each sample t is constructed as a time varying signal that can be analyzed to detect anomalies. Sudden variations in $b(t)$ or traffic beyond normal statistical bounds can be considered as anomalies. Byte counting has $O(1)$ processing cost per packet and $O(1)$ storage cost per sample.

b. Packet Counting

In packet counting, traffic volume again is measured at each sample, but now in terms of packets. The traffic volume in packets $p(t)$ is the time-varying signal that is analyzed to detect anomalies. Packet counts can be more useful than byte counts when links are carrying traffic to the capacity most of the time. Most of automated self-

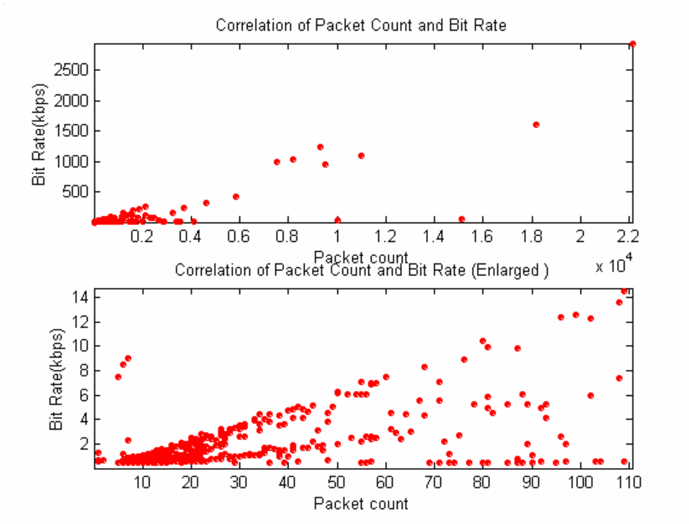


Fig. 40. The relation of byte count and paket count of flows.

propagating codes use constant-sized packets and hence it is possible the packet counts can change as a result of an attack compared to normal traffic. Packet counting has *O(1)* processing cost per packet and *O(1)* storage cost per sample.

Fig. 40 depicts the relationship of bit rates (i.e., byte counts) and packet counts of each flow in NLANR traces. Taking tolerable deviations into account, the packet counts have a linear proportional relation with byte counts in most flows.

c. Flow Counting

This approach counts the number of flows at the observation point. Claffy et al. [63] used the quintuple of source address, destination address, source port, destination port and protocol as the basis of their analysis. A flow can be classified by the 5-tuple (or by another definition) of source address, source port, destination address, destination port, protocol number. During each sampling period, the number of such distinct tuples are counted to generate the flow signal *f(t)*. Flow counting is inherently more difficult than byte counting or packet counting. Hashing and other such techniques would be required to reduce the 5-tuple space to a single flow count number. At the end of each sample, we would need to scan the flow space to count the number of distinct flows seen during the current sample. The costs of such scheme would depend on the hashing techniques employed, the number of tuples chosen to define a flow and the number of flows likely to be seen at the observation point. Hashing can be accomplished in *O(1)* time per packet, individual flow observations can be marked in *O(1)* assuming no hash collisions and the number of flows can be counted in *O(n)* time, where *n* is the size of the flow space. More sophisticated approaches based on bloom filters can be employed to reduce the cost of such approaches to *log(n)* or *loglog(n)* [27], [64].

The number of flows could vary from the norm due to DDoS attacks, wide-scale worm propagation etc. It is expected through monitoring changes in the number of flows, it would be feasible to recognize such anomalies. FlowScan tool analyzes, visualizes and reports Internet traffic flow profiling on flow-centric measurements [7]. Using FlowScan, characteristics of network traffic flow anomalies are illustrated at flow level [36].

2. Vector Signals

Vector signals maintain a vector of data for each sample. This requires more space per sample, but allows more sophisticated analysis of traffic header data. We investigate a new signal based on the protocol numbers. Additionally, we also investigate approaches based on representing network traffic as images. According to employed packet header data and observation domain, we categorize the image-based signals into address-based, flow-based and port-based signals.

a. Protocol Composition

This approach is based on the observation that during the attacks, the protocol employed by the attack traffic should see considerably more traffic than during normal traffic. For example, the recent SQL Slammer worm infected over 90% of the vulnerable hosts employing UDP protocol [48]. In this approach, the amount of ICMP traffic $i(t)$, TCP traffic $t(t)$, UDP traffic $u(t)$ and ETC. traffic $e(t)$ is monitored as a fraction of the total traffic volume at each sampling interval. Because the proportion of each protocol in traffic is closely interrelated to each other, the increase of a proportion of one protocol makes the proportions of other protocols to decrease. Hence, an abrupt increase or decrease of the proportion of traffic of a protocol can indicate anomalies. Protocol composition has $O(1)$ cost per packet and $O(n)$ storage cost per sample, where n is the number of protocols monitored.

b. Image-based Signals

In this approach, traffic distribution in a domain is used as a signal that can be analyzed as explained in chapter IV. First, the traffic volume, such as normalized packet counts and the number of flows, is measured along the packet header domain, such as IP addresses and port numbers. Each resultant traffic datum is converted to corresponding pixel intensity in image representation of traffic in the chosen domain. For example, traffic volume can be counted based on destination port numbers. Since the IP port number field is 16 bits, we would obtain 2^{16} = 64K values for each sample indicating the traffic distribution in the port number domain. For reducing the storage and computation complexity, the traffic header domain can be processed in byte-segments which separate out each byte of the IP address (or the port number) as shown in Fig. 24 through Fig. 31. The image-based signals then originate from the distribution of pixel intensity in each byte of the chosen domain. Based on the kinds of traffic data and the header domain, we categorize the image-based signals into address-based, flow-based and port-based signals. Address-based signal employs traffic volume distribution over address domain (either source address alone, or destination address alone, or a 2-dimensional source and destination address domain). Flow-based signal employs flow number distribution over address domain(s). Port-based signal employs traffic volume distribution over port number domain. And it could be consider the fourth possibility of employing the flow number distribution over port number domain.

Much of the work on image-based signals draws from the large body of work in image processing and video analysis [41], [42], [43], [44]. These techniques enable the detection of abrupt transitions in images or enable the detection of traffic anomalies.

B. Evaluation Methodology

To quantitatively evaluate different packet header data signals for detecting anomalies, we employ two kinds of measurement criteria, which allow comparative and normative studies. The former is to evaluate the performance of various signals based on the detection rates, false alarm rates and likelihood ratios. The latter is to judge the various signals effectiveness through NP Test-based measure which allows the potential of these signals-in-themselves to be compared.

We evaluate the signals through two analyses: first analysis based on statistical properties of the signals and appropriate thresholds (3σ or higher) and the second analysis based on the classical Neyman-Pearson detection methodology.

1. Type I and II Errors

Measurement-based anomaly detection techniques have to contend with two types of errors.

The true positive (sensitivity or detection β) is the probability that a statistical test will be positive for a true statistic. On the other hand, a type I error (false positive error α) occurs if a difference is declared when the null hypothesis is true. In other words, a false attack alarm is declared when the traffic is normal.

$$\begin{aligned}\alpha &= p(\text{anounce } \mathrm{H_1} \mid \mathrm{H_0} \text{ is true}) = p(\text{detect anomaly} \mid \text{traffic is normal}) \\ \beta &= p(\text{anounce } \mathrm{H_1} \mid \mathrm{H_1} \text{ is true}) = p(\text{detect anomaly} \mid \text{traffic is anomaly})\end{aligned} \tag{5.1}$$

The true negative (specificity 1-α) is the probability that a statistical test will be negative for a negative statistic. On the other hand, a type II error (false negative error 1-β) occurs if no difference is declared when the null hypothesis is false. In other words, a false negative is declared that the traffic is normal even when the traffic suffers from attacks.

2. Likelihood Ratio

To test non-nested complementary hypotheses, the LR (likelihood ratio) and NLR (negative likelihood ratio) are used as follows [65].

$$\begin{aligned}\mathrm{LR} &= \frac{\text{true positive rate}}{\text{false positive rate}} = \frac{\text{sensitivity}}{\text{1 - specificity}} = \frac{\beta}{\alpha} \\ \mathrm{NLR} &= \frac{\text{false negative rate}}{\text{true negative rate}} = \frac{\text{1 - sensitivity}}{\text{specificity}} = \frac{1-\beta}{1-\alpha}\end{aligned} \tag{5.2}$$

Because type I and type II errors are dependent on each other (e.g., as β increases, α increases), we carry out the evaluation experiments upon the same set of real traces for all the different traffic signals. LR and NLR help to estimate the efficient trade-off between the power of detection and false alarm. Ideally, LR is infinity and NLR is zero.

3. Statistical Analysis Based on 3σ

NP Test's requirement of parametric knowledge of distributions of the normal and attack traffic may make the test difficult as new attack patterns emerge. In order to overcome this difficulty, we also employ statistical analysis of the traffic data.

Statistical analysis of traffic data requires only a model of normal traffic and hence possibly can distinguish new forms of attacks.

C. NP Test

In order to compare the inherent strengths of various signals, we employ the classical and well-established detection theorem, NP detection. Briefly, here the normal traffic can be seen as noise and attack traffic can be seen to contain a signal (along with noise) that is of interest that needs to be detected as explained in Chapter III.B.3.a. In this section, we introduce the foundations of the considerably promising detection theory principles into the anomaly detection space in network traffic.

NP Test is optimal and works with any distribution of the underlying hypotheses. As explained below, NP Test employs apriori classified data sets for modeling the distributions of the noise and the signal. For anomaly detection purposes, NP Test would require samples of both normal traffic and attack traffic.

1. PDF of H_0 and H_1

In the binary hypothesis testing problem, each of two outputs corresponds to one of two statistical hypotheses [66], the null mode (H_0) and alternative mode (H_1), and an observed datum in the observation space maps into one of the hypotheses.

(i) Noise only, H_0: represents the null hypothesis or the normal network traffic. The probabilistic transition mechanism generates observed data according to a prior conditional probability density under H_0 by

$$p(X=x|H_0)=P(x|H_0) \tag{5.3}$$

,where X is a random variable denoting the observation.

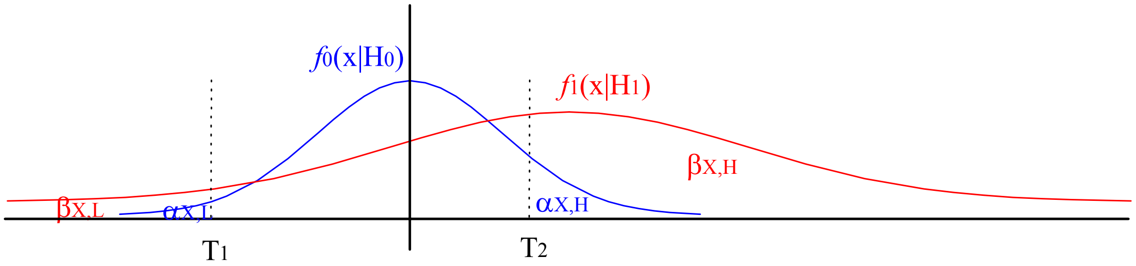
(ii) Signal with noise, H_1: represents the alternative hypothesis or anomalous network activity, i.e., the traffic contains the attack/flash crowd. The probability mass under H_1 is represented by

$$p(X=x|H_1)=P(x|H_1) \tag{5.4}$$

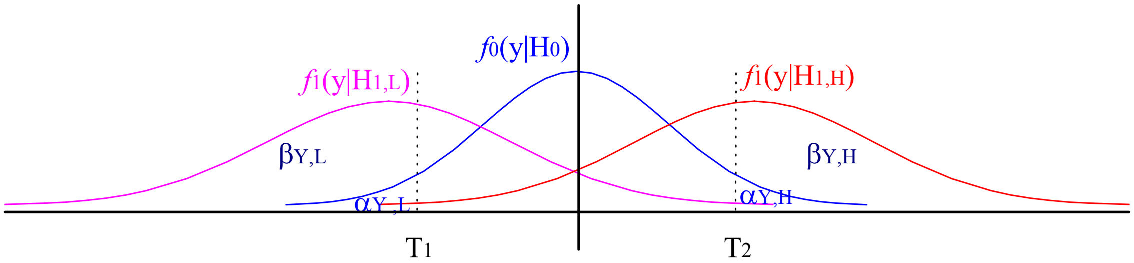
We can define the sample space of one-lateral signals, such as scalar signals, as a random variable *X*. Similarly the two-lateral signals, such as image-based signals, can be defined on two random variables *X* (source domain) and *Y* (destination domain).

The NP Test requires these density functions to be known. To implement this theorem, the total sample space *S* on the real traces is divided into two parts, S_0 and S_1. Observations that fall into S_0 elicit the H_0 hypothesis, and observations that fall into S_1 elicit the H_1 hypothesis.

For accurately detecting the anomalous behavior, it requires a solid model of normal behavior. We look at some statistical properties of aforesaid feasible signals in the normal mode. Based on the probability distribution, we assume that the short-term network traffic S_0 exhibits approximately normal distribution. For example, two random variables *X* and *Y* in the address-based signal are distributed with approximately Gaussian distribution as shown in Fig. 43(a) and 43(b). We verify that observation space S_0 has a normal distribution at 5% significance level through the Lilliefors test, namely H_0: $X \sim N(\mu_{XN}, \sigma_{XN}^2)$ in scalar signals and source domain of image-based signals, and H_0: $Y \sim N(\mu_{YN}, \sigma_{YN}^2)$ in destination domain of image-based signals, where the first subscript indicates the random variable and the second subscript means the noise. The probability density functions (PDF) of H_0 can be expressed as follows.



(a) PDF of H_0, and H_1 with unimodality



(b) PDF of H_0, and H_1 with bimodality

Fig. 41. Illustration of PDF and true / false positive rates.

$$f_0(x|H_0)=\frac{1}{\sigma_{XN}\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{x-\mu_{XN}}{\sigma_{XN}}\right)^2\right] \tag{5.5a}$$

$$f_0(y|H_0)=\frac{1}{\sigma_{YN}\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{y-\mu_{YN}}{\sigma_{YN}}\right)^2\right] \tag{5.5b}$$

Similarly, to model the distribution of abnormal traffic S_1, we excerpt only trace with attacks as samples and investigate the statistical measures. In the case of traffic volume-based signals, the distribution of traffic under H_1 could be considered as approximately normal distribution with large variance. Source domain in image-based signal has also unimodality as $f_1(x|H_1)$ in Fig. 41(a). In the case of destination domain, however, the PDF shows shape close to a bimodal distribution as $f_1(y|H_{1,L})$ and $f1(y|H_{1,H})$ in Fig. 41(b). These two separated modes are located in the tail of the normal distribution of H_0. Each of the modes can be locally modeled to have a rough normal distributed component. For instance, Fig. 43(c) and 43(d) illustrate two (low and high mode) normal probabilities of abnormal traffic in address-based signal Y. The histogram indicates that the data might be appropriately fitted with a mixture of two normal distributions with the different locations and standard deviations [67], [68].

$$\begin{aligned} f_1(y|H_1) &= \varepsilon\phi_1 + (1-\varepsilon)\phi_2 \\ &= \varepsilon f_1(y|H_{1,L}) + (1-\varepsilon) f_1(y|H_{1,H}) \\ &\text{, where } \varepsilon \text{ is mixing proportion} \\ &\phi_1, \phi_2 \text{ are normal PDFs with location and scale parameters } \mu_1, \sigma_1, \mu_2, \sigma_2. \end{aligned} \tag{5.6}$$

The mixing proportion (between 0 and 1) can be fitted using either least squares or maximum likelihood. We set the contamination factor from likelihood of the histogram. Through analysis of sample space, we embody the distribution of H_1, namely H_1: $X \sim N(\mu_{XN} + \mu_{XS}, \sigma_{XN}^2 + \sigma_{XS}^2)$ in scalar signals and source domain of image-based signals, and H_1: $Y_L \sim N(\mu_{YN} + \mu_{YLS}, \sigma_{YN}^2 + \sigma_{YLS}^2)$ and $Y_H \sim N(\mu_{YN} + \mu_{YHS}, \sigma_{YN}^2 + \sigma_{YHS}^2)$ in destination domain, where the first subscript indicates the random variable, with (low and high) mode if necessary, and the second subscript means the signal and noise. The PDF under H_1 can be expressed as follows.

$$f_1(x|H_1) = \frac{1}{\sqrt{2\pi}\sqrt{\sigma_{XN}^2 + \sigma_{XS}^2}} \exp\left[-\frac{1}{2}\frac{\left(x - (\mu_{XN} + \mu_{XS})\right)^2}{\sigma_{XN}^2 + \sigma_{XS}^2}\right] \tag{5.7a}$$

$$\begin{aligned} f_1(y|H_1) = {}& \varepsilon * \frac{1}{\sqrt{2\pi}\sqrt{\sigma_{YN}^2 + \sigma_{Y_LS}^2}} \exp\left[-\frac{1}{2}\frac{\left(y - (\mu_{YN} + \mu_{Y_LS})\right)^2}{\sigma_{YN}^2 + \sigma_{Y_LS}^2}\right] \\ & + (1-\varepsilon) * \frac{1}{\sqrt{2\pi}\sqrt{\sigma_{YN}^2 + \sigma_{Y_HS}^2}} \exp\left[-\frac{1}{2}\frac{\left(y - (\mu_{YN} + \mu_{Y_HS})\right)^2}{\sigma_{YN}^2 + \sigma_{Y_HS}^2}\right] \end{aligned} \tag{5.7b}$$

2. Bayes' Likelihood Ratio Test

The Bayesian criterion method assumes that the two outputs are governed by a priori probabilities, π0 and π1, and that a cost is assigned to each of the four outcomes. These apriori probabilities are *P(H_0 is true)* and *P(H_1 is true)*. These costs are denoted by C_{00}, C_{10}, C_{11}, and C_{01}, where the first subscript indicates the hypothesis accepted and the second indicates the unknown truth. These outcomes respectively map to true negative, false positive, true positive and false negative.

Bayesian criterion leads to likelihood ratio test [69], where a hypothesis is accepted when it is sufficiently likely relative to the other hypothesis. The optimal test is a threshold test of the likelihood ratio. The notion of using the magnitude of the ratio of two PDFs as the basis of a best test will help to provide an intuitively appealing method of constructing a test of a null hypothesis against an alternative hypothesis. The test is defined in scalar signals *X* and source domain *X* of image-based signals as follows.

$$\Lambda(x)=\frac{f_1(x|H_1)}{f_0(x|H_0)}=\frac{\dfrac{1}{\sqrt{2\pi}\sqrt{\sigma_{XN}{}^2+\sigma_{XS}{}^2}}\exp\left[-\dfrac{1}{2}\dfrac{\left(x-(\mu_{XN}+\mu_{XS})\right)^2}{\sigma_{XN}{}^2+\sigma_{XS}{}^2}\right]}{\dfrac{1}{\sigma_{XN}\sqrt{2\pi}}\exp\left[-\dfrac{1}{2}\left(\dfrac{x-\mu_{XN}}{\sigma_{XN}}\right)^2\right]} \qquad (5.8a)$$

$$\begin{cases}\text{if } \Lambda(x)\geq\eta, \text{anounce } \mathrm{H_1}\\ \text{if } \Lambda(x)<\eta, \text{anounce } \mathrm{H_0}\end{cases}$$

And detector is defined in destination domain *Y* of image-based signals as follows.

$$\Lambda(y)=\frac{f_1(y|H_1)}{f_0(y|H_0)}=\frac{\varepsilon*\frac{1}{\sqrt{2\pi}\sqrt{\sigma_{YN}{}^2+\sigma_{Y_LS}{}^2}}\exp\left[-\frac{1}{2}\frac{\left(y-(\mu_{YN}+\mu_{Y_LS})\right)^2}{\sigma_{YN}{}^2+\sigma_{Y_LS}{}^2}\right]}{\frac{1}{\sigma_{YN}\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{y-\mu_{YN}}{\sigma_{YN}}\right)^2\right]}$$

$$+\frac{(1-\varepsilon)*\frac{1}{\sqrt{2\pi}\sqrt{\sigma_{YN}{}^2+\sigma_{Y_HS}{}^2}}\exp\left[-\frac{1}{2}\frac{\left(y-(\mu_{YN}+\mu_{Y_HS})\right)^2}{\sigma_{YN}{}^2+\sigma_{Y_HS}{}^2}\right]}{\frac{1}{\sigma_{YN}\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{y-\mu_{YN}}{\sigma_{YN}}\right)^2\right]} \qquad (5.8b)$$

$$\begin{cases}\text{if } \Lambda(y)\geq\eta, \text{anounce } \mathrm{H_1}\\ \text{if } \Lambda(y)<\eta, \text{anounce } \mathrm{H_0}\end{cases}$$

This value is a random variable, and is tested against the threshold η as

$$\begin{aligned}\eta &= \frac{\pi_0(C_{10}-C_{00})}{\pi_1(C_{01}-C_{11})}\\ &= \frac{\pi_0}{\pi_1} \qquad \text{, when } \alpha+\beta=1\end{aligned} \qquad (5.9)$$

Threshold η can be defined by the ratio of *P(H$_0$ is true)* and *P(H$_1$ is true)*, which are to be determined from a prior knowledge. If the likelihood ratio is greater than η, the detection output is H_1, otherwise the output is H_0.

In many cases, it may be difficult to determine the costs or a prior distribution $\Pi = (\pi_0, \pi_1)$, and we have serious difficulties in setting proper threshold. The NP Test bypasses these factors by introducing the conditional probabilities as (5.10a), (5.10b), (5.11a) and (5.11b). For a practical fidelity criterion, given a constrained significance level α (i.e., false alarm rate) we can derive the threshold η of the test which correspondingly renders the maximum detection rate β.

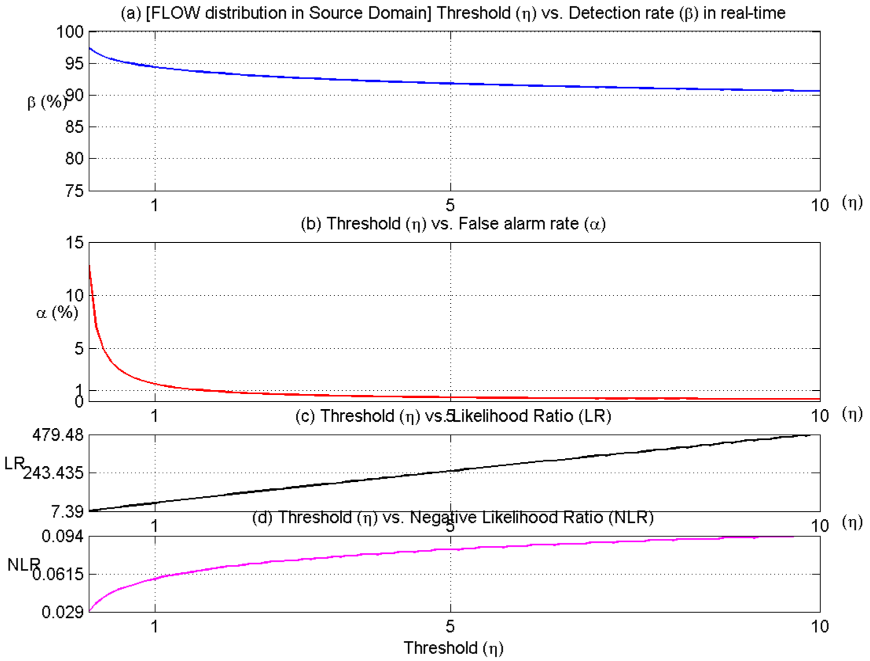
To evaluate our approach against the NP Test, we calculate the false alarm rate and the detection rate based on a given threshold. We set appropriate thresholds of the NP Test so as to correspond to statistical threshold of 3σ. Given the threshold, we solve the Bayesian detector in (5.8a) and (5.8b), and derive critical regions (Z_1) of either boundary (T_1 and T_2) in Fig. 41. These critical regions are close to those of 3σ-based method due to normality.

a. Effect of Threshold (η) in NP Test

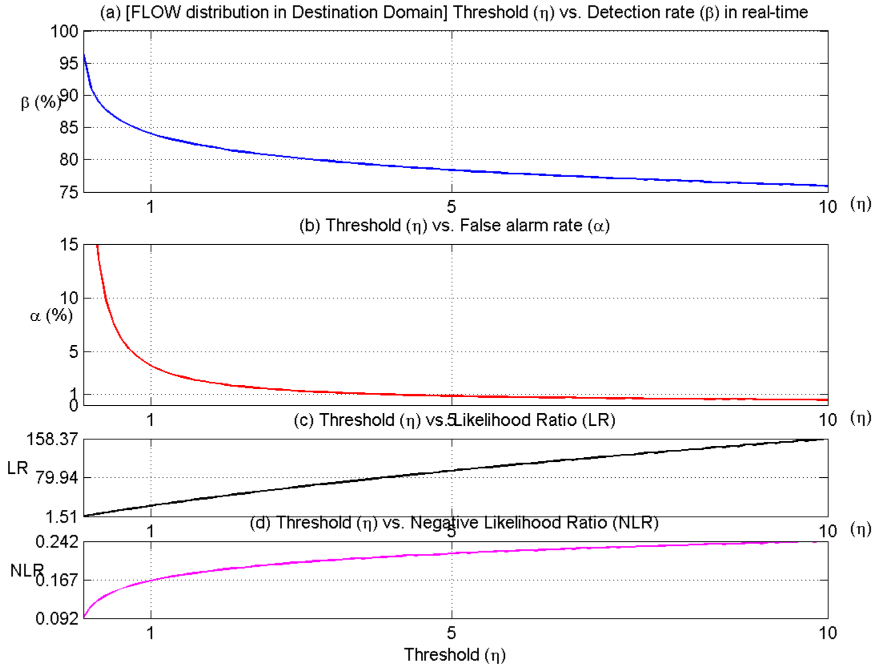
Bayesian criterion requires knowing a prior distribution π_0, π_1, where a threshold is determined as the ratio of π_0/π_1 for choosing decision region so as to minimize total probability of error. The optimal test is the threshold test of the likelihood ratio. Due to inexact knowledge of priors, however, we adopt an alternative fidelity criterion, NP criterion, in which threshold is not in terms of priors. Given a constrained significance level α, we can derive the threshold η of the NP Test for maximizing the detection rate β.

For example, when false alarm rate is constrained to 1.0%, the threshold is derived as about 2.0 and corresponding detection rate reaches 94.0% in source domain as shown in the Fig. 42. We can carry an acceptable trade-off between detection rates and false alarm rates by varying the threshold.

Fig 42 shows the relationship between various performance criteria and thresholds in flow-based image signal in real-time. As shown in Fig. 42, as the threshold increases, the detection rate degrades and the false alarm rate improves. Similarly, the increased threshold make likelihood ratio better and make negative likelihood ratio worse. Generally the measurement criteria of source domain are superior to those of destination as shown in LR and NLR. The β and α of destination domain change more drastically than those of source domain due to bimodality.



[Source Domain]



[Destination Domain]

Fig. 42. The relationship between thresholds (η) and measurements criteria.

3. Expected True and False Positive Rates

We can define the false alarm rate α (type I error) as the overall probability that H_0 is actually true and likelihood ratio test detects H_1 as (5.10a) and (5.10b) from blue-colored PDFs of Fig. 41.

$$\alpha_X = \int_{Z1} f_0(x|H_0)\,dx = \int_{-\infty}^{T1} f_0(x|H_0)\,dx + \int_{T2}^{\infty} f_0(x|H_0)\,dx \tag{5.10a}$$

$$\alpha_Y = \int_{Z1} f_0(y|H_0)\,dy = \int_{-\infty}^{T1} f_0(y|H_0)\,dy + \int_{T2}^{\infty} f_0(y|H_0)\,dy \tag{5.10b}$$

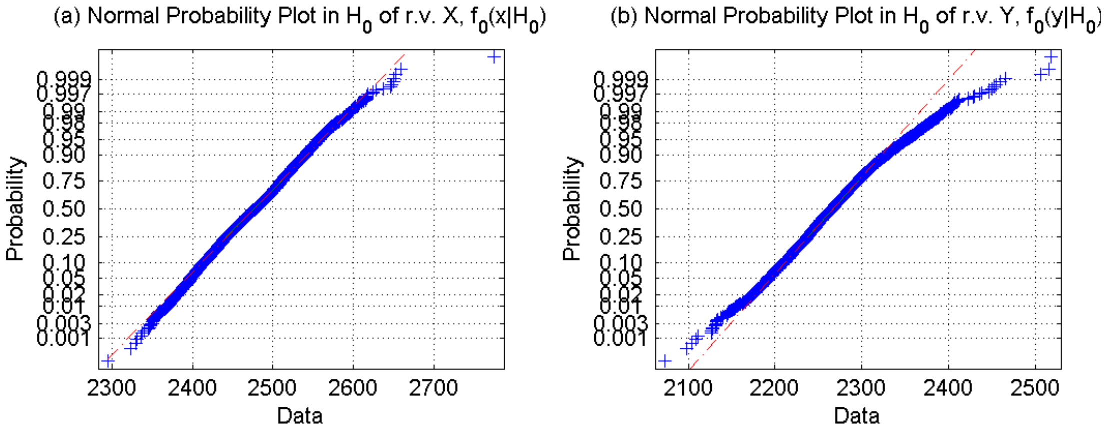
, where Z_1 is critical region : $[-\infty, T_1] \cup [T_2, \infty]$

And the detection rate β in scalar signals and source domain of image-based signals with unimodality is defined as the probability that we successfully detect the anomalies, i.e., H_1 is true and the likelihood ratio test detects H_1 as (5.11a) from red-colored PDF of Fig. 41(a). Consequently false negative rate (type II error) is calculated as 1-β. Similarly, the true positive rates in destination domain of image-based signals with bimodality can be defined as (5.11b) from red/pink-colored PDFs of Fig. 41(b).

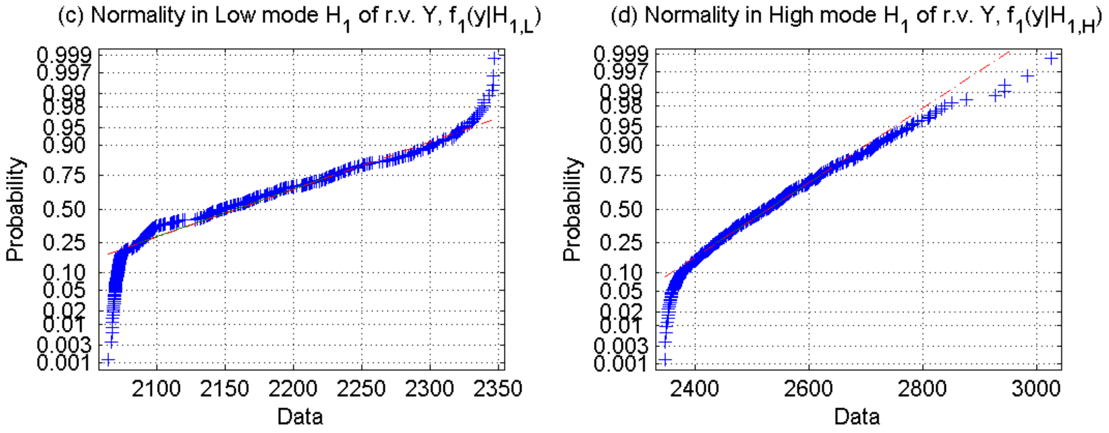
$$\beta_X = \int_{Z1} f_1(x|H_1)\,dx = \int_{-\infty}^{T1} f_1(x|H_1)\,dx + \int_{T2}^{\infty} f_1(x|H_1)\,dx \tag{5.11a}$$

$$\begin{aligned}\beta_Y = \int_{Z1} f_1(y|H_1)\,dy = \varepsilon * \left[\int_{-\infty}^{T1} f_1(y|H_{1,L})\,dy + \int_{T2}^{\infty} f_1(y|H_{1,L})\,dy\right] + \\ (1-\varepsilon) * \left[\int_{-\infty}^{T1} f_1(y|H_{1,H})\,dy + \int_{T2}^{\infty} f_1(y|H_{1,H})\,dy\right]\end{aligned} \tag{5.11b}$$

The objective of the NP Test is to make α as small as possible and β as large as possible. To accomplish this objective, α is constrained by a given tolerable lower bound, and β is maximized using Lagrange multipliers. From derived density function and given thresholds, we can induce the expected true positive rates and false positive rates of each feasible traffic signal.


(a) Normal Probability Plot in H_0 of r.v. X, $f_0(x|H_0)$

(b) Normal Probability Plot in H_0 of r.v. Y, $f_0(y|H_0)$


(c) Normality in Low mode H_1 of r.v. Y, $f_1(y|H_{1,L})$

(d) Normality in High mode H_1 of r.v. Y, $f_1(y|H_{1,H})$

Fig. 43. Normality of address-based signals in real-time mode.
If the data comes from a normal distribution, the plot will appear linear. Other probability functions will introduce curvature in the plot.

4. Application for Traffic Signals in NP Test

We use real-time image-based signal of flow distribution in destination address domain in Table 12 for explanation of how NP Test is to be applied. Through analysis of image-based signals, the distribution of H_0 has a normal distribution at 5% significance level, namely $Y{\sim}N(910.9,\ 146.5^2)$ in destination address. And, we simplify the distribution of H_1, namely $Y_L \sim N(405.9, 45.0^2)$ and $Y_H \sim N(1518.4, 462.2^2)$. The mixing ratios of low mode and high mode are 0.27 and 0.73.

$$f_0(y|H_0) \approx \frac{1}{146.5\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-910.9}{146.5}\right)^2\right]$$

$$f_1(y|H_1) = \varepsilon f_1(y|H_{1,L}) + (1-\varepsilon) f_1(y|H_{1,H}), \text{ where } \varepsilon \text{ is mixing proportion} \tag{5.12}$$

$$\approx 0.27 * \frac{1}{45.0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-405.9}{45.0}\right)^2\right] + 0.73 * \frac{1}{462.2\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-1518.4}{462.2}\right)^2\right]$$

Under given threshold and PDFs which is defined as (5.5b) and (5.7b), we solve the NP Test as (5.8b) and derive critical regions of either boundary.

For destination address variable Y

$$\Lambda(y) = \frac{0.27 \frac{1}{45.0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-405.9}{45.0}\right)^2\right] + 0.73 \frac{1}{462.2\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-1518.4}{462.2}\right)^2\right]}{\frac{1}{146.5\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-910.9}{146.5}\right)^2\right]} = 14.08 \tag{5.13}$$

through numerical analysis

$$y_{1,2} = 910 \pm 425$$

$$\begin{cases} Z_0 : 485 < y < 1335 \\ Z_1 : y \le 485, 1335 \le y \end{cases}$$

General optimal nonlinear test is performed as shown in Appendix A. These critical regions are close to those of 3σ-based method, $471<y<1350$ in (3.4). The 3σ method and NP detector achieve almost equivalent detection performance.

We can compute the false alarm rate α (type I error) as the interval probability distribution from (5.10b) as,

$$\alpha_Y \approx \int_{-\infty}^{485} \frac{1}{146.5\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-910.9}{146.5}\right)^2\right] dy + \int_{1335}^{\infty} \frac{1}{146.5\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-910.9}{146.5}\right)^2\right] dy \approx 0.0037 \tag{5.14}$$

Similarly, the detection rate β is calculated as a mixture of two interval probabilities from (5.11b) as,

$$\begin{aligned}\beta_Y \approx\ & 0.27\left\{\int_{-\infty}^{485} \frac{1}{45.0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-405.9}{45.0}\right)^2\right] dy + \int_{1335}^{\infty} \frac{1}{45.0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-405.9}{45.0}\right)^2\right] dy\right\} + \\ & 0.73\left\{\int_{-\infty}^{485} \frac{1}{462.2\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-1518.4}{462.2}\right)^2\right] dy + \int_{1335}^{\infty} \frac{1}{462.2\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-1518.4}{462.2}\right)^2\right] dy\right\} \\ \approx\ & 0.746\end{aligned} \tag{5.15}$$

In flow-based signal of the Table 12, in case of destination address domain in real-time, the false alarm rate is about 0.37% and the detection rate is 74.6%. With expected detection rates and false alarm rates, we can evaluate the power of feasible traffic signals.

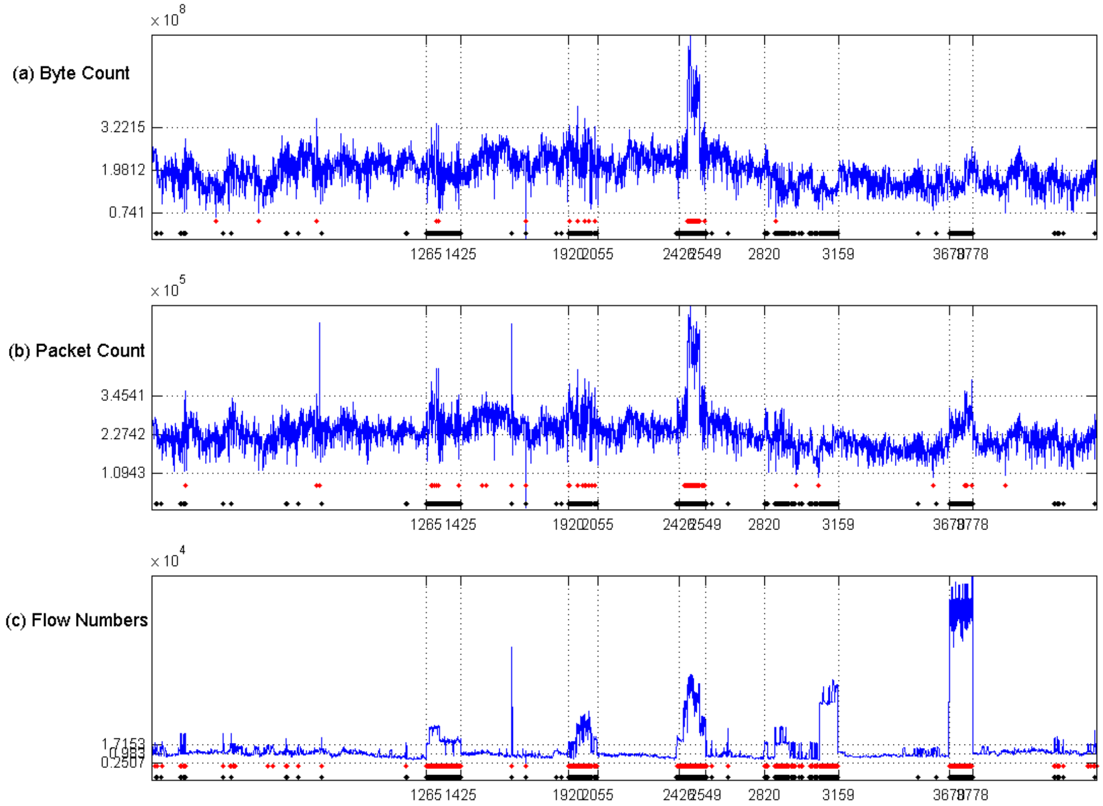


Fig. 44. Trace-driven detection results in scalar signals.
The red dots located on the top are marked when traffic volume-based signals declare anomalies. The black dots located on the bottom show actual anomalous traffic. The dotted vertical lines mean 5 major attacks. The dotted horizontal lines respectively show the T_H, mean and T_L based on 3σ method of ambient signals.

D. Evaluation of Signals

1. Byte Counting and Packet Counting

Fig. 44 and Table 9 shows the results of employing scalar signals on the KREONet2 trace for anomaly detection.

Fig. 44(a), 44(b) and the 2nd and 3rd column of the Table 9 demonstrate the detection strength of the traffic volume-based signals. In KREONet2 traces, there are 5

TABLE 9

RESULTS OF SCALAR SIGNALS [5]

Signals	**T.P. β** [a]	**F.P. α** [b]	***NP β*** [c]	***NP α*** [d]	**LR** [e]	**NLR** [f]
Byte count	11.0% 86/782	0.11% 4/3563	*35.3%*	*0.15%*	98.0/ *241.8*	0.89/ *0.65*
Packet count	18.3% 143/782	0.25% 9/3563	*32.4%*	*0.16%*	72.4/ *206.8*	0.82/ *0.68*
Flow number	95.1% 744/782	0.73% 26/3563	*91.7%*	*0.24%*	130.4/ *384.6*	0.05/ *0.08*

a. True Positive rate by 3σ-based statistical analysis

b. False Positive rate by 3σ-based statistical analysis

c. expected true positive rate by Neyman-Pearson test

d. expected false positive rate by Neyman-Pearson test

e. Likelihood Ratio in measurement by 3σ / *LR in NP test*

f. Negative Likelihood Ratio by 3σ / *NLR in NP test*

major different styled attacks marked by dotted vertical lines and a few instantaneous probe attacks. As the pictures show, except the 3rd attack, the remaining 4 attacks did not set off any distinguishable variance in traffic volume. SQL Slammer worm located in the 5th attack, for instance, spawned a large number of traffic connections with a single packet of relatively small size of 404 bytes. Compared to large elephant flows, these flows have insignificant prominence in byte and packet counts.

The results show that byte count and packet count signals with statistical thresholds achieved detection rates of 11.0% and 18.3% respectively. This is compared to feasible detection rates of 35.3% and 32.4% with the NP Test-based analysis of the same signals. The byte count signal is slightly better; however, the difference between the two measurements is marginal. These results show that the traffic volume signals may not be adequate for providing reliable signals for anomaly detection. Moreover, even when attack traffic may not induce significant overshoot in traffic volume (merely replacing

---

[5] Statistical analysis is in non-italic type whereas NP-Test is in italic type.

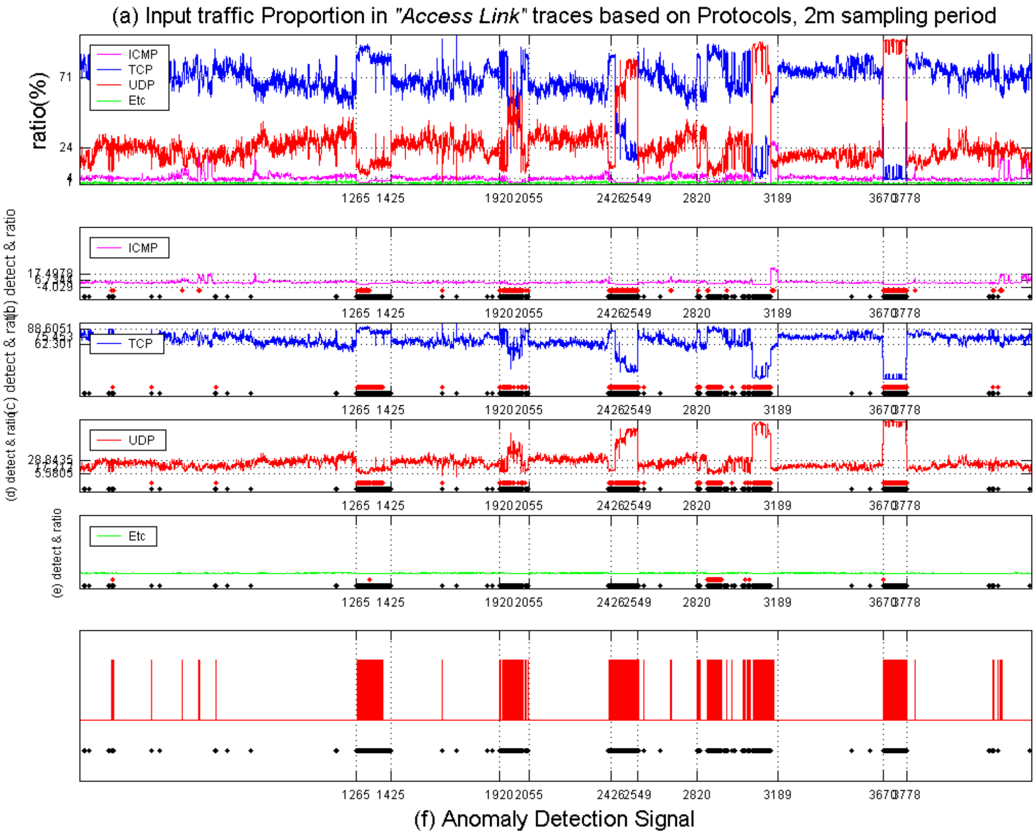


Fig. 45. Detection results based on protocol composition.
The (a) sub-picture shows the proportions of all protocols. The (b) shows the variation of ICMP protocol with time, (c) shows that of TCP, (d) shows that of UDP and (e) shows that of remnant protocols. The above red-colored dots located on the bottom in each sub-picture are marked when each protocol declares anomalies. The below black-colored dots located on the bottom show real anomalies. The dotted horizontal lines respectively show the average proportion of each protocol. The figure (f) shows the generated composite attack detection signal in red which is combined from the attack detection of each protocol. The black dots located on the bottom show actual anomalous traffic.

existing normal traffic), these observations illustrate that anomaly detection should be feasible by studying the distributions of aggregate traffic.

2. Flow Counting

Flow counting shows significantly better performance in detecting anomalous traffic as shown in Fig. 44(c) and the 4th column of the Table 9. Flow counting with statistical

TABLE 10
RESULTS OF PROTOCOL COMPOSITION SIGNALS

Signals	**T.P. β**	**F.P. α**	***NP β***	***NP α***	**LR**	**NLR**
ICMP	72.9% 570/782	1.94% 69/3563	*72.4%*	*0.37%*	37.6/ *195.7*	0.28/ *0.27*
TCP	81.0% 633/782	0.42% 15/3563	*83.7%*	*0.31%*	192.3/ *270.0*	0.19/ *0.16*
UDP	77.5% 606/782	0.39% 14/3563	*78.5%*	*0.26%*	197.2/ *302.2*	0.23/ *0.22*
ETC.	31.7% 248/782	0.00% 0/3563	*76.8%*	*0.81%*	∞ / *94.8*	0.68/ *0.23*
Composite signal	80.8% 632/782	0.28% 10/3563	–[1]	–	288.0	0.19

[1] Multi-component signal is hard to model in NP Test

thresholds achieved a detection rate of 95.1% at a false alarm rate of 0.73%. Flow counting is clearly superior to both byte counting and packet counting signals. Flow counting with NP Test-based analysis provided a detection rate of 91.7% with an accompanying false alarm rate of 0.24%. NP Test-based analysis achieved a smaller false alarm rate with an accompanying loss in detection rate compared to the statistical approach.

We study the flow counting signal further in section E.1 because of its promise based on the results in Table 9.

3. Protocol Composition

We employ 2 kinds of thresholds, a high threshold T_H that indicates that the fraction of volume of one of the network protocols increases abnormally and a low threshold T_L indicating that the fraction of the traffic volume of the network protocol decreases inordinately. When each protocol proportion in current input traffic is larger (or lower) than the 3σ of normal distribution for individual protocol, the detector declares

TABLE 11
RESULTS OF ADDRESS-BASED SIGNALS [1]

Time	**D.**	**TP β**	**FP α**	***NP β***	***NP α***	**LR**	**NLR**
Real-time	SA [2]	81.5% 637/782	0.06% 2/3563	*76.3%*	*0.15%*	1451.2/ *508.7*	0.19/ *0.24*
	DA [3]	87.1% 681/782	0.42% 15/3563	*88.4%*	*0.15%*	206.9/ *589.3*	0.13/ *0.12*
	(SA, DA)[4]	94.2% 737/782	0.48% 17/3563	–	–	197.5	0.06
Post mortem	SA	88.6% 693/782	0.06% 2/3563	*92.0%*	*0.05%*	1578.7/ *1840.0*	0.11/ *0.08*
	DA	80.2% 627/782	0.14% 5/3563	*84.1%*	*0.14%*	571.4/ *600.7*	0.20/ *0.16*
	(SA, DA)	95.9% 750/782	0.20% 7/3563	–	–	488.2	0.06

1. Measured in forgetting factor of 0.5 unless mentioned.
2. SA stands for Source Address.
3. Destination Address.
4. Source Address and Destination Address in combination.

anomalies. Fig. 45(f) shows the generated composite attack detection signal which declares an anomaly when 2 out of 4 individual protocol signals declare an anomaly.

Fig. 45 and Table 10 show the measurement results for protocol composition. Protocol composition could achieve a detection rate of 80.8% at a false alarm rate of 0.28% as shown in Table 10. This shows that protocol composition could be a useful signal. Protocol composition signal-based detector has lower complexity than the flow counting signal.

E. Evaluation of Image-based Signals

1. Packet Distribution in Address Domain

Usually the packet counts at the source addresses of the outbound aggregate traffic can illustrate the distributed properties of the network traffic usage. Meanwhile, the analysis at the destination address of the outgoing traffic can show the concentration of the flow target.

The results of the address-based signals are shown in Fig. 33 and Table 11. The Fig. 33(b) illustrates that the true positive rate is 87.1 % (681 detected out of 782) and the false positive rate is 0.42% (15 falsely detected out of 3563) for real-time detection based on destination address. And the Fig. 33(d) shows that detection rate is 80.2 % (627 detected) and the false alarm rate is 0.14% (5 false detections) for postmortem analysis, again based on destination address.

NP Test generally shows a little higher performance than statistical analysis with higher LRs and lower NLRs excepting source address in real-time.

The results indicate that the different signals exhibit different strengths in anomaly detection. The destination address-based signal performed better in real-time and the source address-based signal performed better in postmortem when compared with each other. In both of real-time and postmortem analyses, the (source, destination) based signal performed significantly better than the single dimensional signals. However, this signal has a higher storage and processing cost.

2. Flow Distribution in Address Domain

An analysis of the distribution of flows gives an idea what peer to peer transmissions consist of and how they are distributed over the address domain. This flow-based signal deals with not only the variation of the number of flows, but also with the changes in the distribution of flows.

An analysis of the flow-based image could be effective for revealing flood types of attacks. When a flow is defined as the triple of source address, destination address and destination port, the flood-based attacks spread flows over the destination IP addresses (or ports) in random or dictionary mode style attacks. Work in [35] has studied that the

TABLE 12
RESULTS OF FLOW-BASED SIGNALS

Time	**D.**	**TP β**	**FP α**	***NP β***	***NP α***	**LR**	**NLR**
Real-time	SA	90.3% 706/782	0.22% 8/3563	*90.0%*	*0.14%*	402.1/ *663.8*	0.10/ *0.10*
	DA	56.5% 442/782	0.25% 9/3563	*74.6%*	*0.37%*	223.8/ *201.1*	0.44/ *0.25*
	(SA, DA)	92.8% 726/782	0.42% 15/3563	–	–	220.5	0.07
Post mortem	SA	91.6% 716/782	0.20% 7/3563	*91.8%*	*0.14%*	466.0/ *676.6*	0.08/ *0.08*
	DA	52.0% 407/782	0.17% 6/3563	*70.7%*	*0.40%*	305.9/ *178.7*	0.48/ *0.29*
	(SA, DA)	94.8% 741/782	0.20% 7/3563	–	–	482.3	0.05

changing ratios (i.e., the rate of decrease) between the flow numbers of neighboring specific bit-prefix aggregate flows can be used for detecting peculiarities. The abnormality is apparent in sequential address/port generation scheme. The distribution of the number of flows in address space would then be expected to be much different from its normal and historical distribution.

Results of analysis of flow-based images are shown in Table 12. As shown in the Table 12, the source address-based images/signals generally exhibit higher confidence than the destination address-based images/signal for detecting traffic anomalies due to bimodality of destination addresses. If source and destination address signals are jointly adopted, we can expect higher confidence in detection rate with a consequent deterioration of the false alarm rate.

The results from Table 9 and Table 12 can be compared to understand the relative strengths of scalar flow counting signal and the flow-based image signal. It is observed that flow-based images could reduce the false alarm rates. However, it is observed that flow-based images did not improve the detection rates when compared to the scalar flow

TABLE 13
RESULTS OF PORT-BASED SIGNALS

Time	D.	TP β	FP α	*NP β*	*NP α*	LR	NLR
Real-time	SP[1]	83.4% 652/782	0.14% 5/3563	*94.9%*	*0.07%*	594.1/ *1428.8*	0.17/ *0.05*
	DP[2]	96.2% 752/782	0.17% 6/3563	*90.5%*	*0.14%*	571.1/ *630.4*	0.04/ *0.09*
	(SP, DP)[3]	96.8% 757/782	0.25% 9/3563	–	–	383.2	0.03
Post mortem	SP	93.9% 734/782	0.11% 4/3563	*94.2%*	*0.08%*	836.1/ *1183.9*	0.06/ *0.06*
	DP	95.7% 748/782	0.14% 5/3563	*87.1%*	*0.21%*	681.6/ *406.1*	0.04/ *0.13*
	(SP, DP)	96.0% 751/782	0.17% 6/3563	–	–	570.3	0.04

1. SP stands for Source Port
2. Destination Port
3. Source Port and Destination Port in combination

counting signal. From these results, the significant additional storage and processing cost entailed in flow-based images may not be warranted unless the reduction of the false alarm rate is paramount.

In our experiments, the destination flow-based signals failed to identify the 3rd attack in the KREONet2 trace. This attack consists of a concurrent host scan aimed at specific destinations (high threshold), and the SQL Slammer worm which targeted random machines (low threshold). These two simultaneous conflicting attacks complicate the detection by offsetting the address distribution characteristics of each other. As a result, it shows that composite attacks may require multiple signals for analysis. The multidimensional signal is motivated from the grounds.

3. Packet Distribution in Port Domain

Besides address domain, we could analyze and visualize the packet header information in port number domain. An analysis of the port number-based image can

TABLE 14
RESULTS OF MULTIDIMENSIONAL SIGNALS

Signals	T.P. β	F.P. α	LR	NLR
Real-time (S,D) [1]	97.1% 759/782	0.62% 22/3563	157.2	0.03
Postmortem (S,D)	97.4% 762/782	0.34% 12/3563	289.3	0.03

1. Source Domains and Destination Domains in combination

reveal portscan types of attacks. When a machine is the target of a portscan, the distribution of the exploited port numbers would deviate from its normal distribution.

The results in Table 13 indicate that port-based signal could be a powerful signal for anomaly detection achieving detection rates of up to 96% with very low false alarm rates. The statistical thresholds have occasionally outperformed the NP Test-based analysis. For example, the destination port-based postmortem signal achieves a detection rate of 95.7% at a false alarm rate of 0.14% compared to the 87.1% detection rate and 0.21% false alarm rate of the NP Test-based analysis. The difference could be verified in terms of LR (681.6 vs. 406.1) and NLR (0.04 vs. 0.13).

The simultaneous improvement in both the measures is a result of the nature of attacks which probe accessible ports in random or dictionary fashion for infiltration. It also demonstrates the complexity of correct NP Test modeling due to multimodality.

4. Multidimensional Signal

With three distinct image-based signals, we can analyze the traffic properties of each IP address and port number from multiple and diverse viewpoints. We develop a multi-component image-based analysis of traffic signal. With three distinct traffic signals, that are address-based, flow-based and port-based signals, can we improve the rate of detection of the anomalous traffic? We study this issue for both real-time and postmortem analysis. The results are shown in Table 14.

The results from Table 14 suggest that it is possible to improve the detection rates considerably by considering multidimensional signals with an accompanying higher rate of false alarms compared to the individual components of the signal. Using various aspects of packet header data simultaneously facilitates analysis of different strategies of attacks and the effect of composite anomalous traffic. Even though sophisticated attacks could go undetected when only one-component signal is investigated, it could become possible to detect the anomalies employing other signals.

F. Analytical Results

1. Sensitivity of Signals to Thresholds

In order to evaluate the effectiveness of employing different thresholds, we compare the detection results of schemes employing the image-based analysis with a scalar signal, especially the number of flows. The anomaly detection measurements in combination of source and destination domains are shown in Fig 46. At medium confidence levels (3σ), the four kinds of image-based analyses (group 2 to group 5) do not offer significant advantage in detection rates over a scalar signal of flow counting. However, the image-based signals offer significant advantage in false alarm rates. When higher confidence levels ($3.5\sigma \sim 4.0\sigma$) are considered (for decreasing the false alarm rates further), the image-based signals provide significantly better detection results than the flow counting approach. This clearly shows that the vector signal offers more significant improvement in the detection of anomalies than scalar signal.

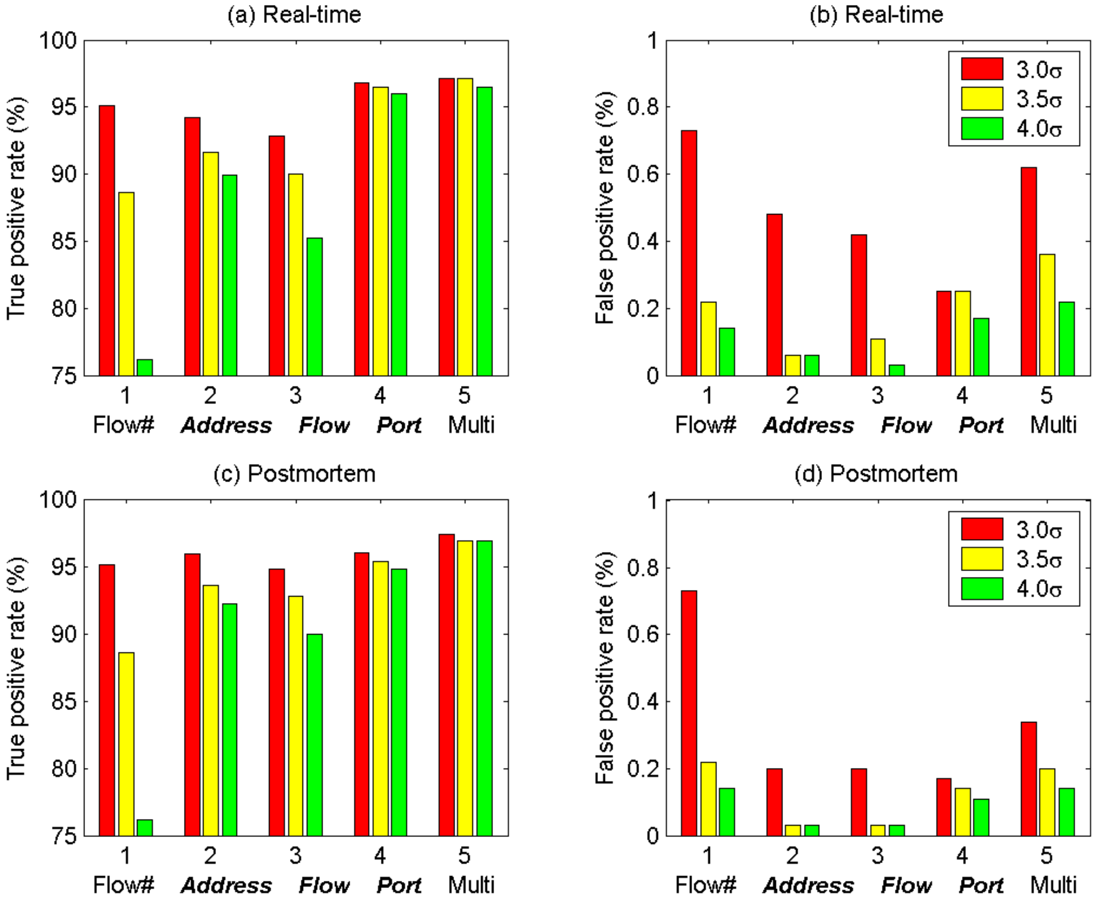


Fig. 46. The relationship between measurements and thresholds.

2. Real-time and Postmortem Analysis

With the measurement results of three forms of image-based signals and multidimensional signals, we observe that the performance of the postmortem analysis is comparable to that of real-time analysis as shown in Fig. 46 even though postmortem analysis uses only small number of DCT coefficients.

The detection rates in on-line analysis are nearly equivalent to true positive rates in off-line analysis as shown in Fig. 46(a) and 46(c). Similarly, the false alarm rates in real-time are close to those of postmortem analysis as shown in Fig. 46(b) and 46(d).

3. General Discussion of Results

The statistical analysis and NP Test show that the performance of the destination address domain slightly degrades due to the bimodality of distribution over the destination address space. These analyses show that the source address-based images/signals generally exhibit higher confidence than the destination address-based images/signals for detecting traffic anomalies especially in false alarm rates and likelihood ratios.

In most experiments, the results of statistical 3σ bounds match those of the NP detector. And these two analyses illustrate consistent evaluation on the power of various signals. Based on these two compatible analyses, we can judge which signals are more effective in detecting traffic anomalies. Our evaluations indicate that the vector signals, which track variations of distributions of the underlying traffic header domains, provide more reliable signals than the scalar signals of traffic volume. Our evaluations also indicate that flow counting is superior to volume counting in scalar signals. Our evaluations show that flow counting can approach the detection rates of vector signals but suffers from higher false alarm rates.

Between the two approaches employed, 3σ approach does not require the analysis of the distribution of H_1 and can be more easily implemented. On the other hand, the NP Test requires PDFs of the H_0 and H_1 to be known and be parametrizable. If the NP Test uses insufficient observation samples S_1 for analyzing anomalous traffic H_1, it could result in a biased modeling, for example inaccurate histogram of the multimodality. While the NP Test is optimal and useful for understanding the inherent strengths of

TABLE 15
PORT-BASED MEASUREMENT RESULTS OF CAMPUS (TAMU) TRACES

Time	**D.**	**TP β**	**FP α**	***NP β***	***NP α***	**LR**	**NLR**
Real-time	SP	93.5% 58 / 62	0.15% 2 / 1313	*98.4%*	*0.28%*	614.1/ *352.9*	0.06/ *0.02*
	DP	91.9% 57 / 62	0.08% 1 / 1313	*95.7%*	*0.18%*	1207.1/ *542.0*	0.08/ *0.04*
	(SP, DP)	95.2% 59 / 62	0.15% 2 / 1313	–	–	634.7	0.05
Post mortem	SP	93.5% 58 / 62	0.08% 1 / 1313	*97.4%*	*0.08%*	1228.3/ 1188.9	0.06/ *0.03*
	DP	91.9% 57 / 62	0.08% 1 / 1313	*94.0%*	*0.12%*	1207.1/ *803.4*	0.08/ *0.06*
	(SP, DP)	93.5% 58 / 62	0.08% 1 / 1313	–	–	1228.3	0.06

TABLE 16
RESULTS OF MULTIDIMENSIONAL SIGNALS OF CAMPUS (TAMU) TRACES

Signals	**T.P. β**	**F.P. α**	**LR**	**NLR**
Real-time (S,D)	96.8% 60 / 62	0.15% 2 / 1313	635.3	0.03
Postmortem (S,D)	98.4% 61 / 62	0.08% 1 / 1313	1291.8	0.02

various signals as used here, statistical approaches may be more effective for anomaly detection in practice (due to unknown attacks).

4. Consideration Using Other Traces

Results from ISP (USC) and Campus (TAMU) traces are shown in the Table 15 through Table 18. Overall, the performances of traffic volume-based signals and image-based signals show consistency regardless of traffic types exploited.

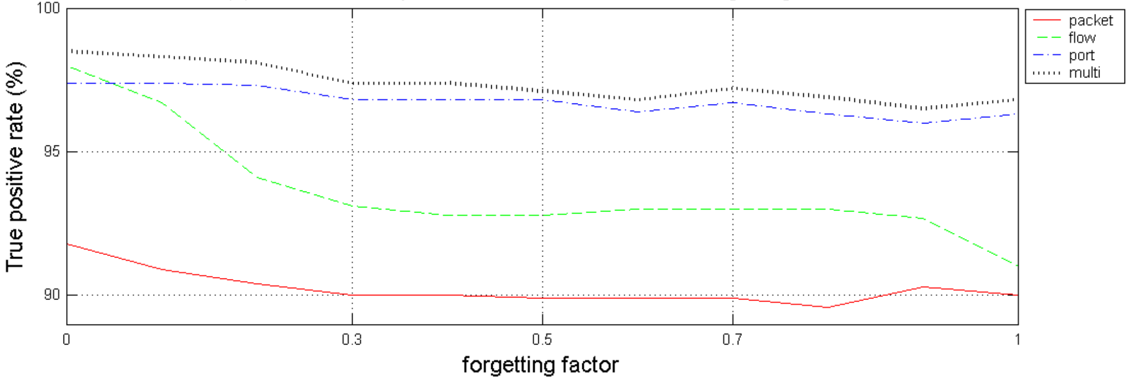
TABLE 17
ADDRESS-BASED MEASUREMENT RESULTS OF ISP (USC) TRACES

Time	**D.**	**TP β**	**FP α**	***NP β***	***NP α***	**LR**	**NLR**
Real-time	SA	84.6% 11 / 13	3.85% 1 / 26	*65.6%*	*0.14%*	22.0/ *468.6*	0.16/ *0.34*
	DA	92.3% 12 / 13	0.00% 0 / 26	*78.5%*	*0.14%*	∞ / *560.7*	0.08/ *0.22*
	(SA, DA)	100.0% 13 / 13	3.85% 1 / 26	–	–	26.0	0.00
Post mortem	SA	84.6% 11 / 13	3.85% 1 / 26	*64.4%*	*0.15%*	22.0/ 429.3	0.16/ *0.36*
	DA	84.6% 11 / 13	0.00% 0 / 26	*75.8%*	*0.14%*	∞ / *541.4*	0.01/ *0.24*
	(SA, DA)	84.6% 11 / 13	3.85% 1 / 26	–	–	22.0	0.16

TABLE 18
RESULTS OF MULTIDIMENSIONAL SIGNALS OF ISP (USC) TRACES

Signals	**T.P. β**	**F.P. α**	**LR**	**NLR**
Real-time (S,D)	100.0% 13 / 13	3.85% 1 / 26	26.0	0.00
Postmortem (S,D)	92.3% 12 / 13	3.85% 1 / 26	24.0	0.08

(a) The relationship between Detection rate and Forgetting factor



(a) The relationship between False alarm rate and Forgetting factor

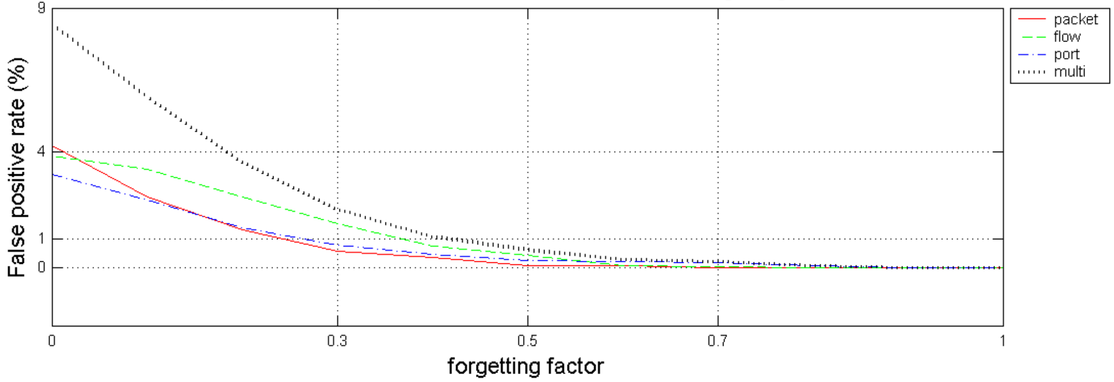
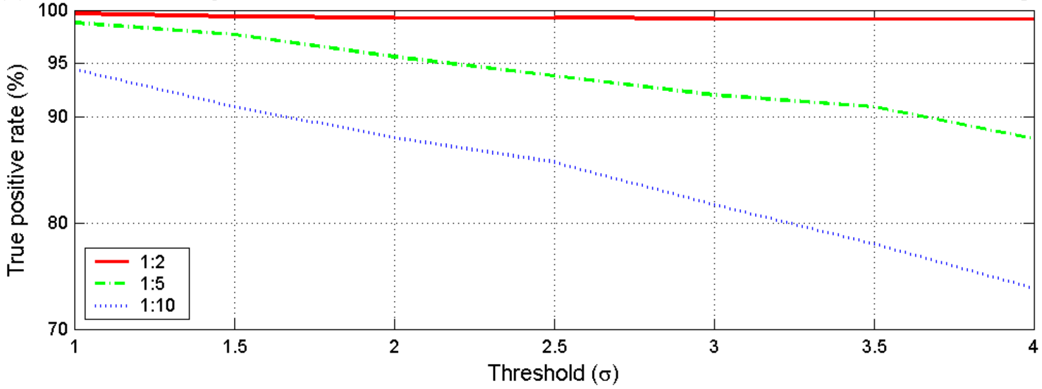


Fig. 47. The relationship between performance and forgetting factor.

5. Effect of Exponentially Weighted Moving Average

To statistically analyze network traffic in real-time, we import the exponentially weighted moving average (EWMA) to fit into the dynamics of traffic. The average of the image-based signals for time *t* is the weighted average of the previous average and the newly observed sample at the time *t*. By using moving average we can filter out short-term noises. We can define the moving-average time series as follows.


(a) The Detectionabilty of the various mixture ratios of simulated attacks in address-based real-time Image signal

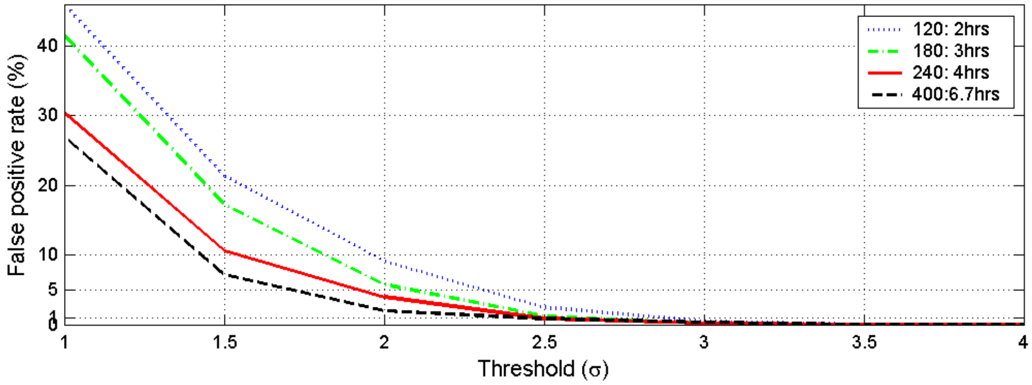

(b) The False alarm rates of the various window sizes in moving window

Fig. 48. The relationship between performance and attack volume.

$$\overline{S(t)} = \alpha \cdot S(t) + (1-\alpha) \cdot \overline{S(t-1)} \quad \text{, where } \overline{S(1)} = S(1) \tag{5.16}$$

Fig 47 shows the relationship between true/false positive rates and forgetting factor. As shown in Fig. 47(a), as the forgetting factor increases, the true positive rates slightly degrade, especially noticeable in the flow-based signal. However, the false positive rates also continue to decrease as the forgetting factor increases. We can carry the trade-off between detection rates and false alarm rates by varying the forgetting factor.

TABLE 19
THE DETECTION LATENCY OF THE VARIOUS MIXTURE RATIOS OF SIMULATED ATTACKS IN REAL-TIME MODE

	conf. level	*mix. ratio*	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	*true positive* [b]	*false positive* [b]
1.0σ	68 %	1 : 2	0[a]	0	0	0	0	0	0	0	0	99.7 % (724/ 726)	30.45 % (1091/ 3583)
		1 : 5	1	0	0	1	0	0	1	0	0	98.8 % (717/ 726)	30.45 %
		1:10	1	0	0	1	0	1	1	0	0	94.4 % (685/ 726)	30.45 %
1.5σ	86 %	1 : 2	0	0	0	0	0	0	0	0	0	99.4 % (722/ 726)	10.66 % (382/ 3583)
		1 : 5	1	0	0	1	0	0	1	0	0	97.7 % (709/ 726)	10.66 %
		1:10	2	0	0	1	0	2	2	0	0	90.9 % (660/ 726)	10.66 %
2.0σ	95.5 %	1 : 2	0	0	0	0	0	0	0	0	0	99.3 % (721/ 726)	3.96 % (142/ 3583)
		1 : 5	1	0	0	1	0	1	2	0	0	95.6 % (694/ 726)	3.96 %
		1:10	2	0	0	1	1	3	4	0	0	88.0 % (639/ 726)	3.96 %
2.5σ	98.5 %	1 : 2	0	0	0	0	0	0	1	0	0	99.3 % (721/ 726)	0.87 % (31/ 3583)
		1 : 5	1	0	0	1	0	2	2	0	0	93.8 % (681/ 726)	0.87 %
		1:10	2	0	0	1	2	4	5	0	0	85.7 % (622/ 726)	0.87 %
3.0σ	99.7 %	1 : 2	0	0	0	1	0	0	1	0	0	99.2 % (720/ 726)	0.20 % (7/ 3583)
		1 : 5	1	0	0	1	0	2	2	0	0	92.0 % (668/ 726)	0.20 %
		1:10	2	0	0	1	2	4	8	0	0	81.7 % (593/ 726)	0.20 %
3.5σ	99.95 %	1 : 2	0	0	0	1	0	0	2	0	0	99.2 % (720/ 726)	0.03 % (1/ 3583)
		1 : 5	1	0	0	1	1	2	4	0	0	90.9 % (660/ 726)	0.03 %
		1:10	2	0	0	1	2	4	29	0	0	78.0 % (566/ 726)	0.03 %
4.0σ	99.99 %	1 : 2	1	0	0	1	0	0	2	0	0	99.2 % (720/ 726)	0.00 % (0/ 3583)
		1 : 5	1	0	6	1	1	2	5	0	1	87.9 % (638/ 726)	0.00 %
		1:10	2	0	6	1	2	4	29	0	2	73.8 % (536/ 726)	0.00 %

a. Latency is measured by minute unit

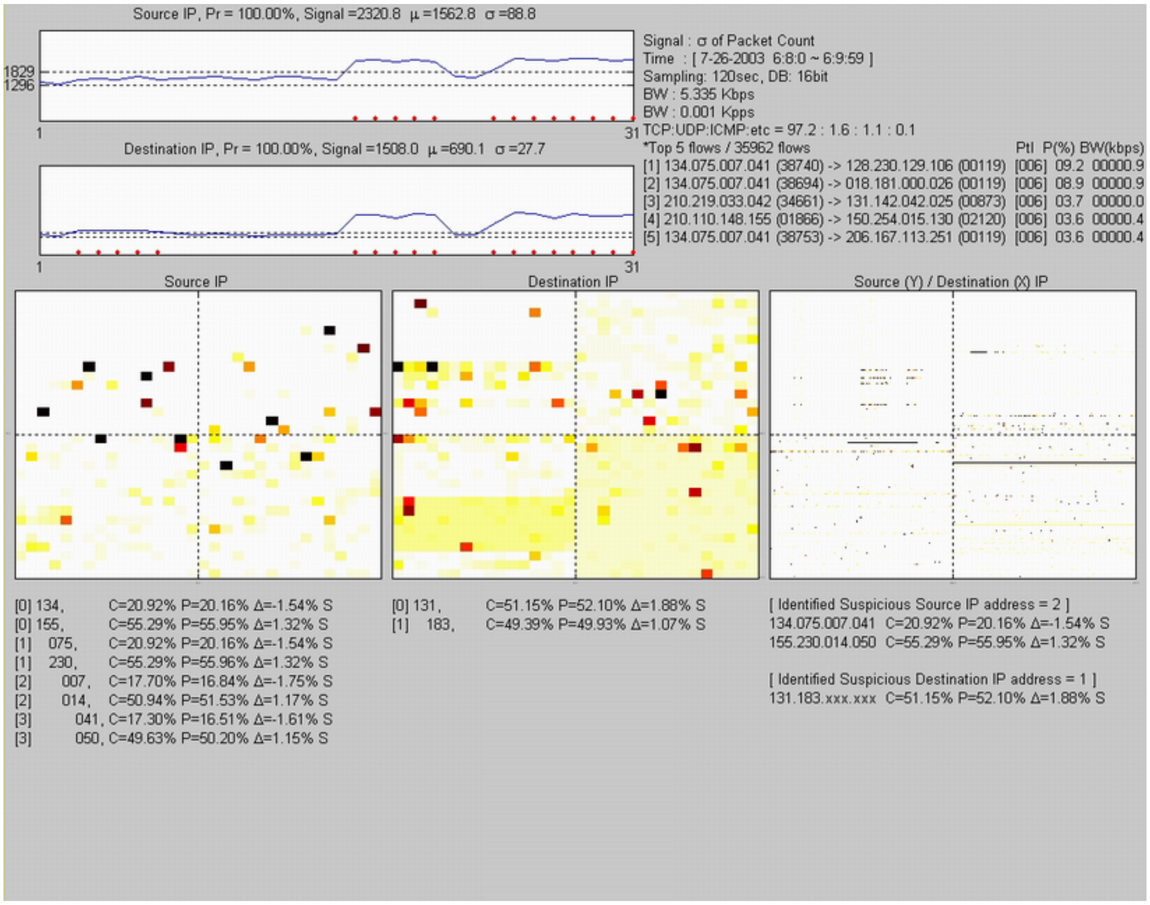
b. Rate in percentile (%) and (true, false) detection / true (anomaly, normal) traffic

6. Sensitivity of Signals to Attack Volume

The trace-driven analysis dealt with preexisting anomalies in the network traffic. In order to study the effectiveness of proposed schemes against attacks that may employ low volume of traffic, we conducted experiments with simulated attacks. The sensitivity experimentation against attack volume goes as follows. First we inject prescribed synthesized attacks, as described in Table 3, into the ambient NLANR traffic at various mixture ratios. The mixture ratios are varied from 1:2, to 1:5 to 1:10 in the ratio of attack traffic to normal traffic. Next the resultant contaminated traffic runs through our detector. Finally, we evaluate the attack detection report by our monitor with the attack specifications. By default the forgetting factor is set as 0.5, the window size is 4 hours, and sampling period is 1 minute.

Fig. 48(a) and true positive rate column in the Table 19 show the sensitivity of the various attack strengths in the default setting. The x-axis in the Fig. 48(a) corresponds to the regulated thresholds, and the y-axis represents the true positive rates. As the threshold increases, the detection rates degrade, especially noticeable in the low-rate attacks. However, even though the detector did not necessarily detect the attack at every sampling point, the detection rates would be adequate to perceive the attack. As shown in Table 19, for example, our detector achieves 81.7% detection rate at the 3.0σ threshold and the 1:10 traffic. It means that the monitor can detect 49 anomalies in 1-hour duration attack in average. Table 19 also presents the delay or latency in detection in minutes (or sampling points). Except for the 7th attack, our detector exhibits fast detection even at low attack volumes. The 7th simulated attack consists of a single source staging an attack on a single destination. At low volumes, such an attack appears as a large flow and cannot be effectively distinguished from normal traffic.

Since the injected attack is superimposed only during attack duration, the mixture ratios do not have an effect on the false positive rate as shown in the Table 19. Fig. 48(b) shows the relative error of the selected window sizes for real-time processing. In case of 2.5σ and over, the false alarm rates are tolerable regardless of window size. The results show that the image-based signal is effective over a wide rage of attack traffic volumes.


Source IP, Pr = 100.00%, Signal =2320.8 μ =1562.8 σ =88.8
1829
1296
1
31
Destination IP, Pr = 100.00%, Signal =1508.0 μ =690.1 σ =27.7
1
31
Source IP
Destination IP
Source (Y) / Destination (X) IP

```
Signal : σ of Packet Count
Time  : [ 7-26-2003  6:8:0 ~ 6:9:59 ]
Sampling: 120sec, DB: 16bit
BW : 5.335 Kbps
BW : 0.001 Kpps
TCP:UDP:ICMP:etc = 97.2 : 1.6 : 1.1 : 0.1
*Top 5 flows / 35962 flows                                      Ptl  P(%) BW(kbps)
[1] 134.075.007.041 (38740) -> 128.230.129.106 (00119)  [006]  09.2  00000.9
[2] 134.075.007.041 (38694) -> 018.181.000.026 (00119)  [006]  08.9  00000.9
[3] 210.219.033.042 (34661) -> 131.142.042.025 (00873)  [006]  03.7  00000.0
[4] 210.110.148.155 (01866) -> 150.254.015.130 (02120)  [006]  03.6  00000.4
[5] 134.075.007.041 (38753) -> 206.167.113.251 (00119)  [006]  03.6  00000.4
```

```
[0] 134,        C=20.92% P=20.16% Δ=-1.54% S
[0] 155,        C=55.29% P=55.95% Δ=1.32% S
[1]   075,      C=20.92% P=20.16% Δ=-1.54% S
[1]   230,      C=55.29% P=55.96% Δ=1.32% S
[2]     007,    C=17.70% P=16.84% Δ=-1.75% S
[2]     014,    C=50.94% P=51.53% Δ=1.17% S
[3]       041,  C=17.30% P=16.51% Δ=-1.61% S
[3]       050,  C=49.63% P=50.20% Δ=1.15% S
```

```
[0] 131,        C=51.15% P=52.10% Δ=1.88% S
[1]   183,      C=49.39% P=49.93% Δ=1.07% S
```

```
[ Identified Suspicious Source IP address = 2 ]
134.075.007.041  C=20.92% P=20.16% Δ=-1.54% S
155.230.014.050  C=55.29% P=55.95% Δ=1.32% S

[ Identified Suspicious Destination IP address = 1 ]
131.183.xxx.xxx  C=51.15% P=52.10% Δ=1.88% S
```

Fig. 49. The running example of NetViewer.

G. Tool Development

We are developing NetViewer, a network monitoring tool that can simultaneously detect, identify and visualize attacks and anomalous traffic in real-time by passively monitoring packet headers. We plan to release our tool to general public.

1. Frame of NetViewer

NetViewer consist of traffic distribution signals, current traffic images and relevant traffic information. Fig. 48 shows the snapshot (or frame) of NetViewer's output.

The above left two sub-pictures illustrate traffic distribution over the latest predefined (and adjustable) time-window. The two pictures map to source domain and destination domain respectively. The captions above each picture explain the current traffic distribution. The Source IP term means that this signal is originated from packet counts in the source IP address domain. This term can be changed to Source FLOW which analyzes the number of flows over the source IP address domain, or to Source PORT which analyzes packet counts in the source port domain, or to Source MULTIDIMENSIONAL which analyzes multiple components of different distributions according to tool user's selection. The Pr term estimates the anomalous probability of current traffic distribution assuming Gaussian distribution. The probability is computed from the following distribution parameters, the signal term, μ and σ, through the normal distribution table. The Signal term is computed by (4.5). The "μ" and "σ" terms mean the mean value and the standard deviation of distribution signal using EWMA. The left two figures and dotted vertical lines illustrate the $\pm 3.0\sigma$ levels. The red dots located on the bottom of the each sub-picture are marked when 3σ-based statistical analysis detects anomalies. The detection signal automatically triggers to identify the IP addresses of sources and destinations and can be used to alert traffic anomalies to the network operator.

The above right text shows the general information of current network traffic. The display includes the signal type, the actual traffic time, the sampling period, the size of the data structure and the bandwidth in Kbps (bits per second) and Kpps (packets per second). The next line illustrates the proportion occupied by each traffic protocol. It can be possible to determine whether the current traffic is behaving normally through correlating it to that of previous states of traffic as explained in chapter V. The Top 5 flows term shows the topmost 5 flows in selected traffic volume based on the traffic signal, which is packet count or the number of flows. It is expressed as a tuple of source IP address/port number and destination IP address/port number, exploited protocol, occupied proportion and bandwidth.

The bottom three sub-pictures illustrate image-based traffic in the source/destination IP address domain and the 2-dimensional domain. Once anomalies are detected, the identified IP addresses of sources and destinations are revealed in byte-segment level and 4-byte whole structure simultaneously. These identified IP addresses are quantitatively investigated on the basis of statistical measurements using the correlation (C), possession ratio (P), delta (Δ) between consecutive frames and the black list (S).

2. Future Version of NetViewer

We are developing a network processor-based system to test the practical feasibility at line speeds. We plan to release our tool to general public. Measurement-based anomaly detection could be combined with IDS approach to enable more effective monitoring of network traffic. We will combine our scheme with an IDS tool such as Snort in the future.

H. Summary

In this chapter, we have evaluated a number of signals proposed for detecting traffic anomalies. We evaluated the signals on three different traces using two different methods. We proposed classical detection theory based on NP Test for detecting anomalies in traffic. We also employed statistical techniques for unknown attack detection. Both the statistical techniques and the NP Test-based evaluations provide similar conclusions. Our evaluations indicate that the vector signals, which track variations of distributions of the underlying traffic header domains, provide more reliable signals than the scalar signals of traffic volume. Our evaluations also demonstrated that statistical techniques can be simpler than and as effective as the NP Test-based analysis.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

We built a robust multidimensional tool that analyzes addresses, port numbers, traffic volume and other data for the detection of anomalies over several timescales. We studied containment approaches along the multiple dimensions of addresses, port numbers, protocols and other such header data (through rate throttling) based on such a detection tool.

We studied the feasibility of analyzing packet header data through wavelet analysis for detecting traffic anomalies. Specifically, we proposed the use of correlation of destination IP addresses and port numbers in the outgoing traffic at an egress router. Our results show that statistical analysis of aggregate traffic header data may provide an effective mechanism for the detection of anomalies within a campus or edge network. We studied the effectiveness of our approach in postmortem and real-time analysis of network traffic. The results of our analysis are encouraging and point to a number of interesting directions for future research.

In this dissertation, we also presented an approach which represents traffic data as images or frames at each sampling point. First, we studied the impact of various design parameters for representing network traffic as images. Such an approach enabled us to view traffic data as a sequence of frames or video and allowed us to apply various image processing and video analysis techniques for studying traffic patterns. We demonstrated our approach through an analysis of traffic traces obtained at three major networks. Our results showed that our approach leads to useful traffic visualization and analysis. Also we studied detection and identification approaches along multiple dimensions of IP packet header data such as addresses, port numbers, and the number of flows.

We compared our statistical approach with classical NP Test from detection theory to evaluate the effectiveness of different traffic signals. We also compared our approach with a signature-based IDS system. Our results indicate that measurement-based statistical approaches can be simple and effective and could be combined with IDS approaches to enable more effective monitoring of network traffic.

We studied the effectiveness and quantitative evaluation of the image-based analysis of traffic with different packet header data and in diverse networks. In this dissertation, we evaluated a number of signals proposed for detecting traffic anomalies. We evaluated the signals on three different traces using two different methods. Both the statistical techniques and the NP Test-based evaluations provide similar conclusions. Our evaluations indicate that the vector signals, which track variations of distributions of the underlying traffic header domains, provide more reliable signals than the scalar signals of traffic volume. Our evaluations also indicate that flow counting is superior to volume counting. Our evaluations show that flow counting can approach the detection rates of vector signals but suffers from higher false alarm rates.

Our evaluations also demonstrated that statistical techniques can be simpler and as effective (and sometimes better) as the NP Test-based analysis. We also showed that it is possible to employ multidimensional signals consisting of multiple components. Our results also showed that a small number of DCT coefficients may be sufficient to represent traffic images for postmortem analysis.

We are developing a network processor-based system to test the practical feasibility at line speeds. We plan to release our tool to general public. Measurement-based anomaly detection could be combined with IDS approach to enable more effective monitoring of network traffic. We will combine our scheme with an IDS tool such as Snort in the future.

To improve the usefulness of our mechanism, further work is recommended as follows.

1) Study our approach against different types of traces such as IPv6 and DNS (Domain Name System). IPv6 expands the available address space and the nature of the address. This not only impacts our data collection, but also the nature of our analysis. It is necessary to study this impact and develop suitable algorithm. Domain space reduction approaches are more important in IPv6 because of the larger address spaces. Moreover, IPv6 allows users to pick different addresses over different times for the same machine. This would further complicate the analysis. Additionally it has been shown that DNS servers can be used to speed up attack propagation in IPv6 networks where addresses may be sparsely distributed in the address domain [70]. This raises the need for studying the traffic to the DNS servers in order to identify anomalies quickly.
2) We need to develop different traffic signals for different types of traffic. We need to analyze if our approach is sensitive to the type of the attacks, (a) intermittent versus persistent, (b) random, semi-random versus sequential scans, and (c) volume of attack traffic as a fraction of link capacity.
3) Techniques and measures from steganalysis can be used to characterize attacks that cannot be easily detected by the proposed mechanisms. In image-based analysis of network traffic, from an attacker's perspective, to keep an attack undetectable, the attack traffic has to be hidden within the traffic distributions (or image representations). This is similar to the body of work done in steganalysis or steganography. Steganalysis studies techniques for hiding information within images or video and techniques for detecting that information. Representing traffic as images allows us to borrow techniques

from steganaysis to understand and characterize the types of attacks that may not be easily detected.

REFERENCES

[1] S. Staniford, V. Paxson, and N. Weaver, “How to 0wn the Internet in Your Spare Time”, in *Proc. of the 11th USENIX Security Symposium (Security ’02)*, San Francisco, CA, August 5-9, 2002.

[2] D. Moore, C. Shannon, G. M. Voelker, and S. Savage, “Internet Quarantine: Requirements for Containing Self-Propagating Code”, in *Proc. of IEEE INFOCOM 2003*, April, 2003.

[3] E. Mitchell, “Introduction to Computer Security”, pp. 9, in *Proc. of Texas Workshop on Security of Information Systems*, College Station, TX, April, 2003. http://net.tamu.edu/~ellenm/papers/ics.ppt

[4] P. Barford, J. Kline, D. Plonka and A. Ron, “A Signal Analysis of Network Traffic Anomalies,” in *Proc. of ACM SIGCOMM Internet Measurement Workshop (IMW) 2002*, Marseille, France, November, 2002. pp. 71-82.

[5] S. Kim, A. L. N. Reddy and M. Vannucci, “Detecting traffic anomalies through aggregate analysis of packet header data”, in *Proc. of Networking 2004, LNCS* vol. 3042, Athens, Greece, May, 2004, pp. 1047-1059.

[6] C. Estan, S. Savage and G. Varghese, "Automatically Inferring Patterns of Resource Consumption in Network Traffic", in *Proc. of ACM SIGCOMM 2003*, Karlsruhe, Germany, August, 2003.

[7] D. Plonka, “FlowScan: A Network Traffic Flow Reporting and Visualization Tool”, in *Proc. of USENIX 14th System Administration Conference (LISA) 2000*, New Orleans, LA, December, 2000.

[8] A. Kuzmanovic and E. Knightly, "Low-Rate TCP-Targeted Denial of Service Attacks", in *Proc. of ACM SIGCOMM 2003*, Karlsruhe, Germany, August, 2003.

[9] A. Ramanathan, “WADeS: A Tool for Distributed Denial of Service Attack Detection”, M.S. thesis, Texas A&M University, College Station, August, 2002.

Available: http://dropzone.tamu.edu/techpubs/2002/thesis_ramanathan.pdf.

[10] A. Feldmann, A. Gilbert, P. Huang and W. Willinger, "Dynamics of IP traffic: A Study of the Role of Variability and the Impact of Control", *Computer Communication Review,* vol. 29, no. 4 (Proc. of the ACM Sigcomm '99, Cambridge, MA), pp. 301-313, 1999.

[11] J. Kilpi and I. Norros, "Testing the Gaussian approximation of aggregate traffic", in *Proc. of ACM SIGCOMM Internet Measurement Workshop (IMW) 2002*, Marseille, France, November, 2002.

[12] S. Floyd and V. Jacobson, "Random Early Detection Gateway for Congestion Avoidance", *IEEE/ACM Transactions on Networking,* vol. 1, pp. 397-413, August, 1993.

[13] D. Tong and A. L. N. Reddy, "QoS Enhancement with Partial State", in *International Workshop on QoS*, London, England, June, 1999, pp. 87-96.

[14] Smitha and A. L. N. Reddy, "LRU-RED: An Active Queue Management Scheme to Identify High Bandwidth Flows at a Congested Router", in *Proc. of IEEE Globecomm*, San Antonio, TX, November, 2001.

[15] P. Achanta and A. L. N. Reddy, "Design and Evaluation of a Partial State Router", in *Proc. of IEEE ICC*, Paris, France, June, 2004.

[16] T. J. Ott, T. V. Lakshman and L. H. Wong, "SRED: Stabilized RED", in *Proc. of IEEE INFOCOM '99*, vol. 3, 1999, pp. 1346-1355.

[17] M. Roesch, "Snort-Lightweight Intrusion Detection for Networks", in *Proc. of USENIX 13th Systems Administration Conference (LISA) 1999*, Seattle, Washington, November, 1999.

[18] Snort, June, 2004. Available: http://www.snort.org/.

[19] J. Mirkovic, G. Prier and P. Reiher, “Attacking DDoS at the Source”, in *10th IEEE International Conference on Network Protocols*, Paris, France, November, 2002. pp. 312-321.

[20] C. Wong, C. Wang, D. Song, S. Bielski and G. R. Ganger, “Dynamic Quarantine of Internet Worms”, *The International Conference on Dependable Systems and Networks (DSN-2004),* Florence, Italy, June 28-July 01, 2004, pp. 62-71.

[21] A. Garg and A. L. N. Reddy, "Mitigation of DoS attacks through QoS regulation", in *Proc. of IWQOS Workshop*, Miami Beach, FL, May, 2002.

[22] J. Ioannidis and S. M. Bellovin, “Implementing Pushback: Router-Based Defense against DDoS Attacks”, in *Proc. of Network and Distributed System Security Symposium*, San Diego, CA, February, 2002.

[23] Y. Zhang, L. Breslau, V. Paxson and S. Shenker, “On the Characteristics and Origins of Internet Flow Rates”, in *ACM SIGCOMM 2002*, Pittsburgh, PA, August, 2002.

[24] Smitha, I. Kim and A. L. N. Reddy, "Identifying long term high rate flows at a router", in *Proc. of High Performance Computing*, Hyderabad, India, December, 2001.

[25] I. Kim, “Analyzing Network Traces To Identify Long-Term High Rate Flows”, M.S. thesis, Texas A&M University, College Station, May, 2001.
Available: http://dropzone.tamu.edu/techpubs/2001/thesis_ben9.pdf.

[26] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson and S. Shenker, “Controlling High Bandwidth Aggregates in the Network (Extended Version)”, in *ACM SIGCOMM Computer Communication Review,* vol. 32, no. 3, July 2002.

[27] C. Estan and G. Varghese, “New Directions in Traffic Measurement and Accounting”, in *ACM SIGCOMM 2002*, Pittsburgh, PA, August, 2002.

[28] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya and C. Diot, "Traffic Matrix Estimation: Existing Techniques and New Directions", in *ACM SIGCOMM 2002*, Pittsburgh, PA, August, 2002.

[29] Packeteer, "PacketShaper Express", white paper, 2003.
Available: http://www.packeteer.com/resources/prod-sol/Xpress_Whitepaper.pdf

[30] S. Floyd, S. Bellovin, J. Ioannidis, K. Kompella, R. Mahajan and V. Paxson, "Pushback Messages for Controlling Aggregates in the Network", *IETF Internet-draft, work in progress*, July, 2001. Expired draft.

[31] S. Savage, D. Whetherall, A. Karlin and T. Anderson, "Practical network support for IP traceback", in *Proc. of ACM SIGCOMM*, 2000.

[32] C. Cheng, H. T. Kung and K. Tan, "Use of spectral analysis in defense against DoS attacks", in *Proc. of IEEE Globecom*, 2002.

[33] I. H. Witten, A.Moffat and T. C. Bell, *Managing Gigabytes – Compressing and Indexing Documents and Images*, 2nd ed., San Francisco, CA: Morgan Kaufmann, 1999, pp. 129–141.

[34] T. M. Gil and M. Poletto, "MULTOPS: A Data-Structure for Bandwidth Attack Detection", in *Proc. of the 10th USENIX Security Symposium*, Washington, D.C., August, 2001. pp. 23-38.

[35] E. Kohler, J. Li, V. Paxson and S. Shenker, "Observed Structure of Addresses in IP Traffic, in *Proc. of ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November, 2002. pp. 253-266.

[36] P. Barford and D. Plonka, "Characteristics of Network Traffic Flow Anomalies", in *Proc. of ACM SIGCOMM Internet Measurement Workshop (IMW) 2001*, October, 2001.

[37] S. Kim and A. L. N. Reddy, "A Study of Analyzing Network traffic as Images in Real-Time", in *Proc. of IEEE INFOCOM 2005*, Miami, FL, March, 2005.

[38] B. Krishnamurthy, S. Sen, Y. Zhang and Y. Chen, "Sketch-based Change Detection: Methods, Evaluation, and Applications", in *Proc. of ACM SIGCOMM Internet Measurement Conference (IMC) 2003*, Miami, FL, October 2003.

[39] B. Bloom, "Space/time trade-offs in hash coding with allowable errors", *Communications of ACM,* vol. 13, no. 7, pp. 422-426, July, 1970.

[40] H. Kim, I. Kang, and S. Bahk, "Real-time Visualization of Network Attacks on High-speed Link", *IEEE Network Magazine*, September-October, 2004.

[41] D. Lelescu and D. Schonfeld, "Statistical Sequential Analysis for Real-time Video Scene Change Detection on Compressed Multimedia Bitstream", *IEEE Transactions on Multimedia,* vol. 5, no. 1, pp. 106-117, 2003.

[42] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic Partitioning of Full-motion Video", *Multimedia Systems*, vol. 1, no. 1, pp. 10-28, 1993.

[43] R. Lienhart, C. Kuhmunch, and W. Effelsberg, "On the Detection and Recognition of Television Commercials", in *Proc. of the International Confernce on Multimedia Computing and Systems,* Ottawa, Canada, 1997, pp. 509-516.

[44] K. Shen and E. J. Delp, "A Fast Algorithm for Video Parsing Using MPEG Compressed Sequences", in *IEEE Conference on Image Processing,* vol. 2, 1995, pp. 252-255.

[45] CERT Coordination Center (CERT/CC), "CERT Advisory CA-2003-04 MS-SQL Server Worm", January, 2003.
Available: http://www.cert.org/advisories/CA-2003-04.html.

[46] National Laboratory for Applied Network Research (NLANR), Measurement and Operations Analysis Team, "NLANR Network Traffic Packet Header Traces", August, 2002. Available: http://pma.nlanr.net/Traces/.

[47] KREONet2 (Korea Research Environment Open NETwork2).
Available: http://www.kreonet2.net.

[48] D. Moore, V. Paxon, S. Savage, C. Shannon, S. Staniford and N. Weaver, "*The Spread of the Sapphire/Slammer Worm*", the technical report of CAIDA. Available: http://www.caida.org/outreach/papers/2003/sapphire/sapphire.html.

[49] S. Kim, A. L. N. Reddy and M. Vannucci, "Detecting traffic anomalies using discrete wavelet transform", in *Proc. of International Conference on Information Networking (ICOIN) 2004, LNCS* vol. 3090, Busan, Korea, Feburary, 2004, pp. 951-961.

[50] C. S. Burrus, R. A. Gopinath and H. Guo, *Introduction to Wavelets and Wavelet Transforms*, Upper Saddle River, NJ: Prentice Hall, 1998.

[51] I. Daubechie, "Ten Lectures on Wavelets", *vol. 61, CBMS-NSF Regional Conference Series in Applied Mathematics*, Philadelphia: Society for Industrial and Applied Mathematics, 1992.

[52] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp 674-693, 1989

[53] G. W. Wornell, *Signal Processing with Fractals: A Wavelet Based Approach*, Upper Saddle River, NJ: Prentice Hall, 1996.

[54] P. Huang, A. Feldmann and W. Willinger, "A non-intrusive, wavelet-based approach to detecting network performance problems", in *Proc. of Internet Measurement Workshop (IMW)*, Nov. 2001.

[55] D. B. Percival and A. T. Walden, *Wavelet Methods for Time Series Analysis*, chapter 4, Cambridge, UK: Cambridge University Press, 2000.

[56] The MathWorks. Inc., *MatLab software*, ver 6.1.0.450 Release 12.1, May 2001.

[57] E. R. Dougherty, *Random Processes for Image and Signal Processing*, New York: SPIE/IEEE Press, 1999, pp. 61.

[58] S. Kim and A. L. N. Reddy, "Modeling Network traffic as Images", in *Proc. of IEEE ICC 2005*, Seoul, Korea, May, 2005.

[59] A. Gyaourova, C. Kamath, and S.-C. Cheung, "Block Matching for Object Tracking", LLNL Technical Report, UCRL-TR-200271, October, 2003.

[60] Analysis Console for Intrusion Databases (ACID), June, 2004.
Available: http://www.cert.org/kb/acid.

[61] A. Hussein, J. Heidemann, and C. Papadopoulus, “A framework for classifying denial of service attacks”, in *Proc. of ACM SIGCOMM 2003*, Karlsruhe, Germany, August, 2003.

[62] A. Lakhina, M. Crovella and C. Diot “Diagnosing Network-wide Traffic Anomalies”, in *Proc. of ACM SIGCOMM 2004*, September, 2004.

[63] K. C. Claffy, H. Braunn and G. C. Polyzos, “A Parameterizable Methodology for Internet Traffic Flow Profiling”, in *IEEE Journal on Selected Areas in Communications*, vol.13, no. 8, pp. 1481-1494, October 1995.

[64] M. Durand and P. Flajolet, “Loglog Counting of Large Cardinalities”, in *"Engineering and Applications Track" of the 11th Annual European Symposium on Algorithms (ESA03), LNCS* vol. 2832, September, 2003, pp. 605-617.

[65] MathWorld: The web’s most extensive mathematics resource, July, 2004.
Available: http://mathworld.wolfram.com/LikelihoodRatio.html.

[66] H. V. Poor, *An Introduction to Signal Detection and Estimation*, 2nd ed., New York: Springer Press, 1994, pp. 11.

[67] NIST/SEMATECH *e-Handbook of Statistical Methods,* May, 2004.
Available: http://www.itl.nist.gov/div898/handbook/eda/section3/histogr5.htm.

[68] E. Parzen, “On Estimation of a Probability Density Function and Mode”, *The Annals Mathematical Statistics*, vol. 33, no. 3, pp. 1065-1076, September, 1962.

[69] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*, New York: Macmillan Company, 2nd ed., 1965, pp. 285.

[70] A. Kamra, H. Feng, V. Misra and A. D. Keromytis, "The Effect of DNS Delays on Worm Propagation in an IPv6 Internet", in *Proc. of IEEE INFOCOM 2005*, Miami, FL, March, 2005.

[71] CERT@ Coordination Center, "*CERT@ Advisory CA-1997-28 IP Denial-of-Service Attacks*", December, 1997.
Available: http://www.cert.org/advisories/CA-1997-28.html.

[72] D. Bruschi and E. Rosti, "Disarming offense to facilitate defense", in *Proc. of the 2000 Workshop on New Security Paradigms,* Ballycotton, County Cork, Ireland, 2000, pp. 69-75.

[73] H. Wang, D. Zhang and K. G. Shin, "Detecting SYN Flooding Attacks", in *Proc. of IEEE INFOCOM 2002*, New York City, NY, 2002.

[74] D. Moore, G. Voelker and S. Savage, "Inferring Internet Denial of Service Activity", in *Proc. of USENIX Security Symposium 2001*, August, 2001.

[75] S. Kim and A. L. N. Reddy, "The Real-time Detection and Containment of Network Attacks Using QoS Regulation", in *Proc. of IEEE ICC 2005*, Seoul, Korea, May, 2005.

APPENDIX

A. Optimal Test

$$\mathrm{H}_0 : \mathrm{X} \sim \mathrm{N}(\mu_N, \sigma_N{}^2)$$

$$\mathrm{H}_1 : \mathrm{X} \sim \mathrm{N}(\mu_N + \mu_S, \sigma_N{}^2 + \sigma_S{}^2)$$

$$\Lambda(x) = \frac{f_1(x|H_1)}{f_0(x|H_0)} = \frac{\dfrac{1}{\sqrt{2\pi}\sqrt{\sigma_N{}^2+\sigma_S{}^2}} \exp\left[-\dfrac{1}{2}\dfrac{\left(x-(\mu_N+\mu_S)\right)^2}{\sigma_N{}^2+\sigma_S{}^2}\right]}{\dfrac{1}{\sqrt{2\pi}\sigma_N} \exp\left[-\dfrac{1}{2}\dfrac{\left(x-\mu_N\right)^2}{\sigma_N{}^2}\right]} \mathop{\gtrless}_{H_0}^{H_1} \eta$$

$$\exp\left[-\frac{\left(x-(\mu_N+\mu_S)\right)^2}{2(\sigma_N{}^2+\sigma_S{}^2)} + \frac{\left(x-\mu_N\right)^2}{2\sigma_N{}^2}\right] \gtrless \frac{\sqrt{\sigma_N{}^2+\sigma_S{}^2}}{\sigma_N}\eta$$

by taking *ln*

$$-\frac{\left(x-(\mu_N+\mu_S)\right)^2}{2(\sigma_N{}^2+\sigma_S{}^2)} + \frac{\left(x-\mu_N\right)^2}{2\sigma_N{}^2} \gtrless \ln\frac{\sqrt{\sigma_N{}^2+\sigma_S{}^2}}{\sigma_N} + \ln\eta$$

by setting ratio $\frac{\sigma_N}{\sigma_S} = R_\sigma$, and new threshold $\tilde{\eta}$ as

$$\mu_S{}^2 \mathrm{R}_\sigma{}^4 + \left(\mu_S{}^2 + 2(\sigma_N{}^2+\sigma_S{}^2)(\ln\frac{\sqrt{\sigma_N{}^2+\sigma_S{}^2}}{\sigma_N} + \ln\eta)\right)\mathrm{R}_\sigma{}^2 + \left(\frac{1}{\sigma_S{}^2} - 1\right)\mu_N{}^2 = \tilde{\eta}$$

$$\left[x + \left(\mu_S \mathrm{R}_\sigma{}^2 - \mu_N\right)\right]^2 \mathop{\gtrless}_{H_0}^{H_1} \tilde{\eta} \tag{A.1}$$

B. Protocol Composition by Traffic Fraction Thresholds

Typical network attacks are the TCP SYN flood, ICMP directed broadcast, Smurf, Ping of Death [48], [71], [72], [73]. It has been shown that more than 90% of the DoS

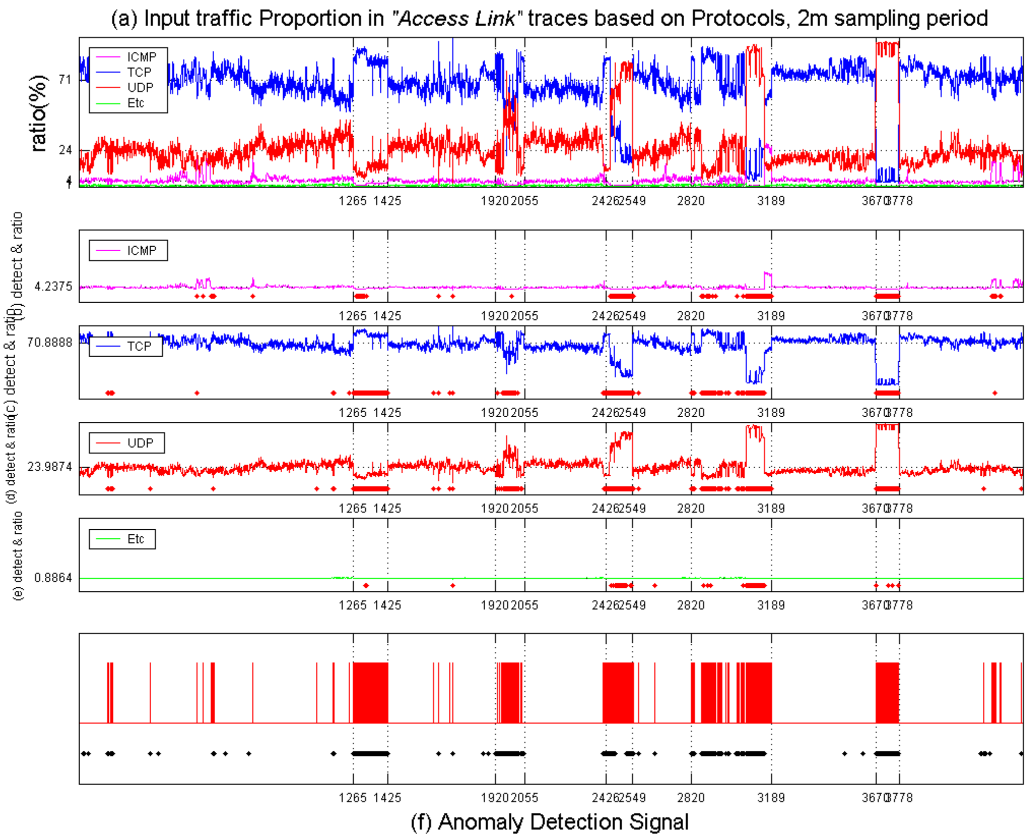


Fig. 50. Detection results based on protocol composition.
The (a) sub-picture shows the proportions of all protocols. The (b) shows the variation of ICMP protocol with time, (c) shows that of TCP, (d) shows that of UDP and (e) shows that of remnant protocols. The red dots located on the bottom are marked when each protocol declares anomalies. The dotted horizontal lines respectively show the average proportion of each protocol. The figure (f) shows the generated attack detection signal in red which is combined from the attack detection of each protocol. The black dots located on the bottom show actual anomalous traffic.

attacks use TCP [74]. We consider an approach based on observing the composition of different protocols in the network traffic. We employ 2 kinds of thresholds, a high threshold T_H that indicates that the fraction of volume of one of the network protocols increases abnormally and a low threshold T_L indicating that the fraction of the traffic

TABLE 20

RESULTS OF PROTOCOL COMPOSITION SIGNALS

Signals	**T.P. β**	**F.P. α**	***NP β***	***NP α***	**LR**	**NLR**
Protocol composition	89.8% 702/782	2.30% 82/3563	–[1]	–	39.0	0.10

1. Multi-component signal is hard to model in NP test

volume of the network protocol decreases inordinately [75]. We set the high threshold for detecting abnormal increase as (A-2).

$$T_H(p,t)=1/\sqrt{r(p,t)} \tag{A.2}$$

, where p is individual protocol

$r(p,t)$ is the current proportion of the protocol

The higher the protocol's proportion of normal traffic is, the lower the setting for its high threshold is. This improves sensitivity. Additionally we employ another threshold, which detects decreases in traffic of that protocol. Because the proportion of each protocol in traffic is closely interrelated to each other, the increase of a proportion of one protocol makes the proportions of other protocols to decrease. By using this reciprocal property we are able to detect anomalies of other protocols. For example, when we observe the abrupt decrease of proportion of traffic other than TCP, we are able to notice TCP-based traffic anomalies. And we set the low thresholds by inversing high thresholds as $T_L(p,t) = 1/T_H(p,t)$.

When each protocol proportion in current input traffic is larger (or lower) than the product of average proportion and the high (or low) threshold for individual protocol as in (A.3), the detector declares anomalies. It indicates the balance of traffic by protocol changed abruptly.

$$\begin{array}{ll} \text{If } r(p,t)\begin{cases} > \overline{r(p,t-1)} * \mathrm{T_H}(p,t) \\ < \overline{r(p,t-1)} * \mathrm{T_L}(p,t) \end{cases}, & \Lambda(p,t) \text{ is attack} \\ \text{Otherwise} & , \ \Lambda(p,t) \text{ is normal} \end{array} \tag{A.3}$$

Fig. 49 and Table 19 show the measurement results for protocol composition. Protocol composition could achieve a detection rate of 89.8% at a false alarm rate of 2.30%. This shows that protocol composition could be a useful signal with a slightly high false alarm rate. Protocol composition signal-based detector has lower complexity than the flow counting signal. As a result, rate control as in [20], window control and weighted fair queuing can play a significant role in protecting servers from anomalous traffic.

VITA

Seong Soo Kim received his B.S. and M.S. degrees in electrical engineering from Yonsei University, Seoul, Korea, in February 1989 and February 1991, respectively. During his MS degree at Yonsei, he was supported by an LG Fellowship.

He worked in the areas of analog/digital consumer electronics and home networking as a research engineer at LG Electronics Co., Ltd. of Korea from January 1991 to August 2001. He received the Ph.D. degree in computer engineering in the Department of Electrical Engineering at Texas A&M University in May 2005. His research interests are in computer network security, multimedia including image and signal processing, and stochastic processing.

He received an "outstanding research engineer" award at LG in 1995 and received a "Patent-Technology" award from the national patent officer in 1996. He has 31 domestic (registered or pending) patents and 5 international patents.

His permanent address is Geoyeo 1-danzi Apt. 104-621, Geoyeo-Dong, Songpa-Gu, Seoul, Korea, Republic of, (postal code) 138-112, and his personal e-mail address is kimseongsoo2@hotmail.com.