

Image-based Anomaly Detection Technique: Algorithm, Implementation and Effectiveness

Seong Soo Kim and A. L. Narasimha Reddy, *Senior Member, IEEE*

Abstract— The frequent and large-scale network attacks have led to an increased need for developing techniques for analyzing network traffic. This paper 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, revealing certain kinds of attacks to the human eye. 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 “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 the effectiveness of *NetViewer* with classical detection theory based Neyman-Pearson test.

Index Terms—Network measurements; Network anomaly detection; Experimentation with real networks/Testbeds; Stochastic processes; Statistical analysis; Image processing.

I. INTRODUCTION

INCREASING malicious network traffic, such as Denial-of-service (DoS) floods, worms and other forms of attacks, have become serious threats to the network security. A number of recent studies have pointed to the need for fast detection of malicious worms to be effective in thwarting such traffic [1]. Network traffic monitoring and analysis tools are being employed to counter this threat. If efficient monitoring and analysis tools are available to network administrators, it could become possible to detect the attacks, anomalies quickly and to appropriately take action to suppress the attacks before they have had much time to propagate across the network.

A number of tools such as FlowScan [2], Cisco’s FlowAnalyzer, and AutoFocus [3], are used to study and classify traffic on the network based on usage and protocols. 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, recent studies have pointed to the need for rigorous real-time analysis for detecting and

identifying the anomalies so that mitigation action can be taken as promptly as possible [1], [24]. Some of these tools are based on the volume of traffic such as byte counts and packet counts. When links are persistently congested, attack traffic may displace normal traffic without causing a perceptible difference in the volume of the traffic on the link. Sophisticated low-rate attacks [4] and replacement attacks, which don’t give rise to noticeable variance in traffic volume, could go undetected when only traffic volume is considered. The tools that collect and process flow data may not scale to high-speed links as they focus on individual flow behavior. Certain attacks such as DDoS attacks and worm attacks are decipherable only when aggregate traffic is considered since these attack flows are not distinguishable from normal flows when only individual flow behavior is considered. In order to address these issues and improve scalability, our approach considers real-time analysis of aggregate packet header data.

Recent studies have shown that the traffic can have strong patterns of behavior over several timescales [5], and our previous work has shown the possibility of analysis of wide-sense stationary (WSS) property in network traffic [6]. Recent work in [7] has shown that Gaussian approximation could work well for aggregated traffic. Self-propagating and automated malicious codes, in general, perturb normal network traffic patterns. 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 an anomalous manner.

Our approach passively monitors packet headers of network traffic at regular intervals and analyzes the aggregate data for anomaly detection. Our approach generates images of the packet header data for both visualization and for effective processing of the collected data. During network anomalies or attacks, the usage pattern of network may change and the peculiarities could become visible in the traffic 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 work here brings techniques from image processing and video analysis to visualization and real-time analysis of traffic patterns.

In this paper, we will report on our approach to anomaly detection along with measurements conducted on real traces of traffic at three major networks. This paper will make the following significant contributions: (a) represents packet header data as images for traffic visualization, thus enabling certain kinds of attacks to become clearly visible to the human eye, (b) applies techniques from image and video processing for the analysis of network traffic, (c) demonstrates the

This work is supported by NSF grants ANI-0087372, 0223785, Texas Information Technology and Telecommunications Taskforce, Texas Higher Education Board and Intel Corp. Manuscript received August 26, 2005.

Seong Soo Kim is now working at Digital Media R&D Center of Samsung Electronics Co., Ltd. in Korea (e-mail: kimseongsoo2@hotmail.com).

A. L. Narasimha Reddy is with the Department of Electrical and Computer Engineering, Texas A&M University, 315B WERC, College Station, TX 77843-3128, USA (e-mail: reddy@ece.tamu.edu).

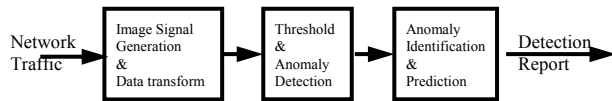


Fig. 1. The block diagram of NetViewer.

effectiveness of such measures in detecting and identifying the attacks in real-time with very small latencies (d) shows that statistical techniques can be as effective or sometimes more effective than the NP-test because of changing attack patterns and (e) demonstrates that the distributions of traffic in the underlying domains (address, port number, protocol, etc.) provide more effective signals than traffic volume alone.

II. RELATED WORK

A number of popular monitoring tools such as FlowScan, Cisco's FlowAnalyzer, and AutoFocus [3], are used as traffic analyzers. FlowSan is open source software to gather and analyze network flow data taken from NetFlow records of Cisco routers [2]. FlowScan first writes raw flow files that are later post-processed for providing information on network traffic. However, excessive backlog of flow files due to heavy-traffic or flood-based DoS attacks may make it difficult for carrying out real-time analysis of traffic. Using FlowScan, characteristics of network traffic flow anomalies can be illustrated at the flow level [8].

Recently, traffic volume has been analyzed using wavelets to detect anomalies in network traffic [5]. Work in [6] has considered correlation of addresses as a signal for analysis for anomaly detection. Recent work has considered TCP-specific signals, such as number of resets, to detect specific attacks [9]. While earlier work analyzed traffic as a time series of a single variable, our work here tries to analyze distributions over different domains of packet header data, particularly the address space and port number distributions [28], as potential signals that can be analyzed in order to detect anomalies in network traffic. Our work also brings the tools from image processing and video analysis to traffic analysis. Recent work in [27] has employed entropy for analyzing traffic distributions.

Sketch-based techniques are shown to perform close to that of per-flow methods for network traffic analysis [10]. Recent work in [11] has similarly employed 3 hash functions and LRU 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 and (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 [11] has some similarities (with significant differences in data representation and anomaly detection) to our work presented here.

While traffic analysis can be carried anywhere in the network, earlier work has focused on the edges of administrative domains [6], [25], [26] where it is easy to control any detected anomalies.

Much of the work reported here draws from the large body of work in 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 [12], color histogram differences [13], characteristic patterns in the standard deviation of pixel intensities for detection of fades [14], and color histogram of DC coefficients [15]. 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.

III. OUR APPROACH

We employ packet header data collected at a network access point for traffic analysis. This data may come from different domains such as source/destination addresses, port numbers, protocol numbers, etc. The collected data in each domain may include traffic volume in bytes, packets, number of flows 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, the image may represent traffic volume in bytes or packets going to a destination or the traffic between a (source, destination) pair. Similarly, the image may represent the port numbers seen during the sample. The image may represent the number of port numbers or flows seen between a (source, destination) pair.

The packet header domains can be large. For example, the IPv4 addresses require 2^{32} entries and the (source, destination) address pair would require 2^{64} entries. In order to reduce the memory and processing requirements of the data collection and analysis, we employ domain folding techniques. Hashing has been traditionally employed for such purposes. However, the one-way functions employed in hashing make it difficult to relate the observed anomalies back to the packet header domains. Hence, we employ simpler techniques that while reducing the memory and processing requirements allow correlating the observed phenomena back to the packet headers.

Image 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 from different domains can be represented as different colors of an image leading to uniform treatment and analysis.

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. Under some attacks (as seen with recent semi-random worm attacks), motion prediction techniques can potentially identify the patterns of attack behavior. For example, single source attacking multiple destinations will be represented by horizontal lines in the (source, destination) traffic volume image. Similarly, a Distributed DoS (DDoS) attack against a single destination would be represented by vertical lines in the

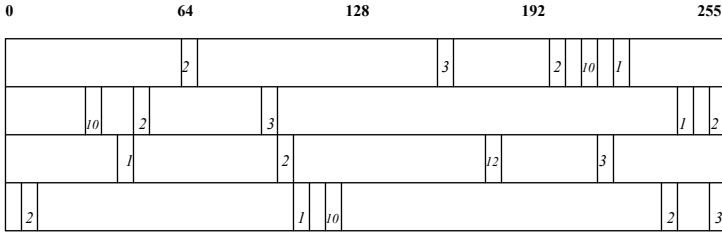


Fig. 2. Data structure for computing normalized packet count. Suppose that only five flows exist, their source (or destination) IP addresses and packet counts are as follows. IP of Flow 1 = 165. 91.212.255, Packet count of F1 = 3; IP of F2 = 64. 58.179.230, P2 = 2; IP of F3 = 216.239. 51.100, P3 = 1; IP of F4 = 211. 40.179.102, P4 = 10; IP of F5 = 203.255. 98. 2, P5 = 2

(source, destination) image. A portscan attack would be similarly visible in the port number-based images.

NetViewer consists of three major components as shown in Fig. 1. The first step consists of traffic signal generation, in which the image signal is generated from samples of network traffic. NetViewer can work with packets on the fly at a router and packet traces in libpcap (packet capture library) or NetFlow formats as described in sections IV and V.

The second stage is detection, in which transformed images are analyzed for their distributions. Various techniques from image processing and video analysis can be applied. The variance of image pixel intensities or Discrete Cosine Transform (DCT) coefficients can be used for scene change analysis as presented in section VI.

The final stage is identification and prediction, in which attackers and victims are revealed using line or edge detection algorithms. Moreover, we show that it is possible to predict the attack patterns using motion prediction algorithms in section VI.

Finally, we compare the effectiveness of NetViewer with that of well established NP test and popular Intrusion Detection System (IDS) Snort in sections VII and VIII.

A. Traces

To verify the validity of our approach, we run NetViewer on three kinds of real traffic traces.

First, we examine the tool on traces from the University of Southern California (USC) [16], which contains real network attacks in the pcap header format. Second, to inspect the sensitivity of our tool on backbone links, we examine the tool on KREONet2 traces from Oct. 12, 2003 to Oct. 26, 2003, which contain actual 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. KREONet2 trace is a collection of NetFlow trace files by the 155Mbps international ATM link. Third, we examine the tools on a live network in Texas A&M University (TAMU) campus.

IV. VISUAL MEASUREMENT OF THE NETWORK TRAFFIC

A. Data Structure for Visual Representation

We illustrate our approach with a specific example 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, the number of flows, etc. We use packet counts in the address domain here as a primary 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 (1).

$$p(m, n) = p_{mn} / \sum_m p_{mn} \tag{1}$$

We employ a simple data structure, as used in [6], 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 as shown in Fig. 2 and the normalized packet count is quantized and represented in sampling point n by (2).

$$P_{ijn} = \frac{count[i][j][n]}{\sum_{j=0}^{255} count[i][j][n]}, \quad i = 0,1,2,3, \quad j = 0, \dots, 255 \tag{2}$$

Each resultant normalized packet count represents the intensity of the corresponding pixel in the image representation of the traffic.

B. Visual Representation of Network Traffic

Each byte of the IP address has 256 entries. We arrange the normalized packet count of the 256 entries of the each byte into a 16-by-16 square, in a row-major order, for visual representation at the sampling point. The four 16-by-16 squares, which correspond to each of the 4 bytes of one IP address, are organized as a frame for the source and destination addresses respectively as in Fig. 3(a). Similarly, with 256-by-256 squares,

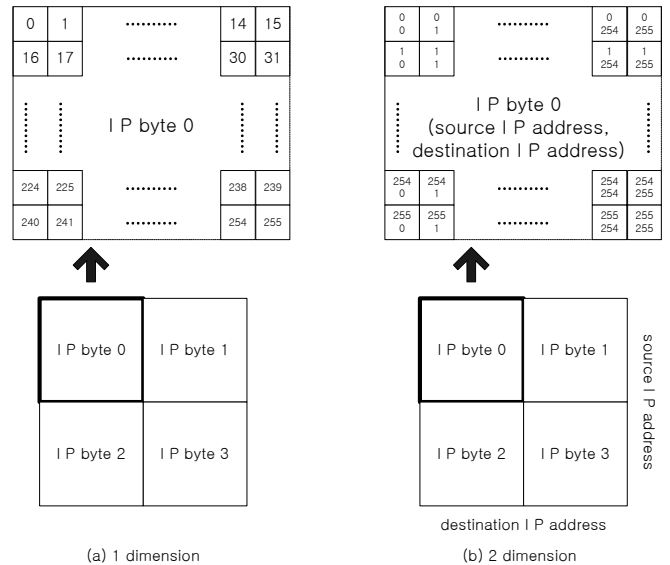


Fig. 3. The visualization of network traffic signal in IP address domain.

we can express the normalized values for the source and destination addresses simultaneously as in Fig. 3(b). Even though pixels neighboring in the image space are not necessarily neighbors in the address space, a notion of locality is not affected through maintaining the index of the data structure. Here the intensity (gray-level) of the pixel is directly proportional to the normalized packet count.

V. MODELING NETWORK TRAFFIC AS IMAGES

Self-propagating and automated malicious codes or worms 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 convergence to the destination into (i) a single target, (ii) semi-random targets (subnet and other prefix-based attacks), and (iii) random targets. Single target attack can be considered as a special case of a semi-random target case. We look at traffic, in general, as in normal behavior mode, in semi-random and in random attack mode.

A. Visual patterns in normal network traffic

Fig. 4 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 lower 3 sub-pictures, Fig. 4(c), 4(d) and 4(e) visually illustrate the normalized packet counts as outlined in Fig. 3. The aggregate traffic does not form

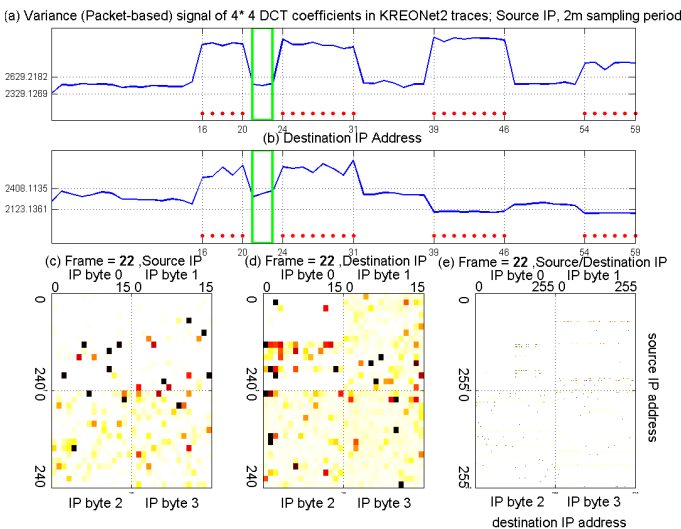


Fig. 4. Visual measurement of normal network traffic. A green rectangular time-window in (a) and (b) sub-pictures indicates the current sampling points. The bottom red dots in (a) and (b) illustrate the anomaly detection signals and the vertical dotted lines are the periods of actual anomalies. The (c) and (d) sub-pictures show the intensity of network traffic 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 of the (source, destination) pair in 2-dimension 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 unidimensional) regardless of color representation.

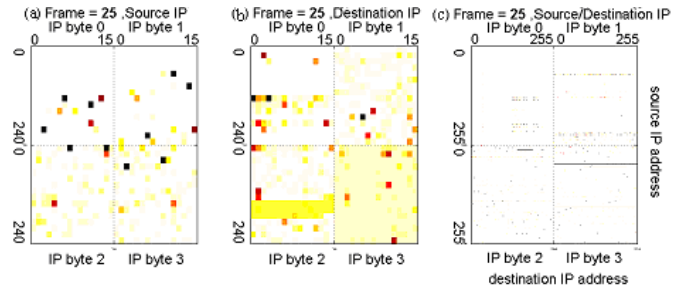
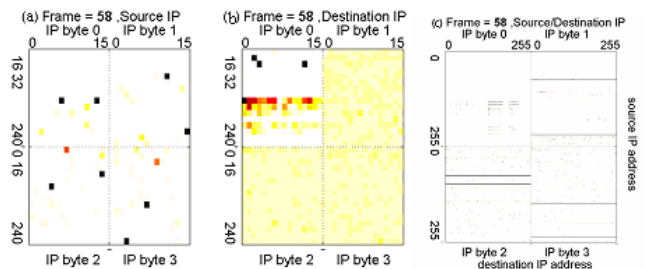


Fig. 5. Visual measurement of semi-random typed attack.

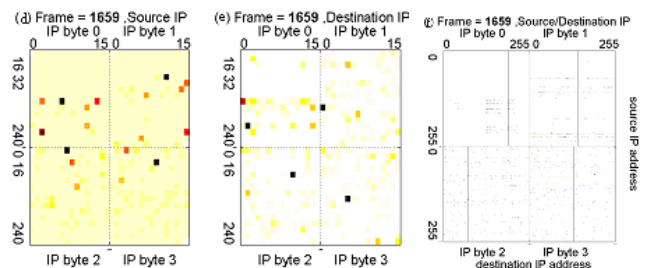
any regular shape due to dispersibility of traffic of various and numerous flows in time and space. The color and darkness of each pixel point up the intensity of traffic of corresponding IP address. Fig. 4(a) and 4(b) sub-pictures show the standard deviation of pixel intensities in real-time by (3) (or of 16 MSB DCT coefficients in postmortem by (3) after DCT) in source/destination addresses respectively. The sampled distribution signal at the current instant is shown within the green rectangles of Fig. 4(a) and 4(b). We will discuss later how these signals can facilitate anomaly detection.

B. Visual patterns in semi-random targeted attacks

Fig. 5 shows the visual measurement of P_{ijn} of the source/destination IP addresses in a semi-random targeted traffic state. From Fig. 5(b) 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 section VI.C. From the 3rd and 4th bytes of Fig. 5(c), 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



[Worm propagation type of attack traffic]



[Distributed Denial of Service (DDoS) type of attack traffic]

Fig. 6. Visual measurement of random styled attacks.

variance shows much higher values than normal traffic cases.

C. Visual patterns in random targeted attacks

Fig. 6 shows the traffic during a random attack. From Fig. 6(b), 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. 6(c), it shows that two specific sources, i.e., two attackers (visible through two black pixels in each byte quadrant in Fig. 6(a) and two horizontal lines in each quadrant in 6(c)), scan all possible destinations. Through later identification stages, we can identify 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-line 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-line 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.

D. Visual patterns in complicated attacks

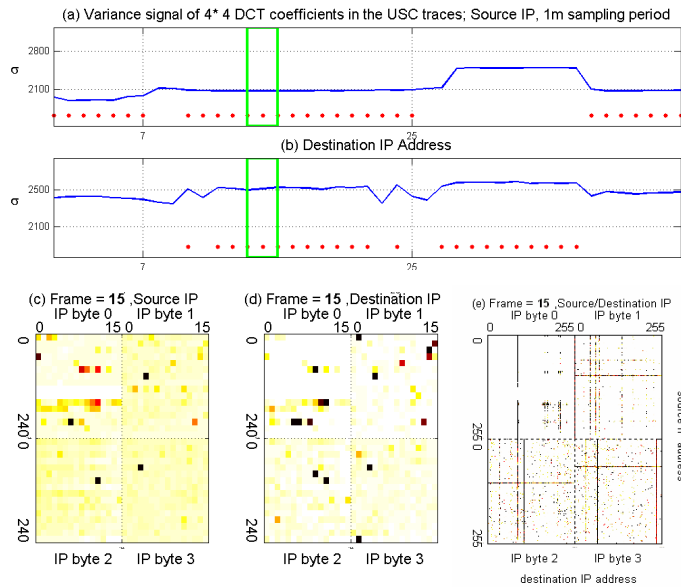


Fig. 7. Visual measurement of mixed network attack. The (c) sub-picture shows intensity of network traffic in a few of the source IP addresses. For example, the IP address 100 in 2nd byte, the 107 in 3rd byte, and the 67 in 4th byte can be considered as suspicious attack sources. From (e) sub-picture, 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.

We illustrate complicated and mixed attack patterns using USC traces in Fig. 7. Between the 7th and the 25th frames, randomly generated source addresses attack specific destination addresses. Moreover, from Fig. 7(c) and 7(e), we can infer that a few specific source addresses lead the attack. That is, the horizontal (dotted or solid) line in Fig. 7(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.

VI. ANOMALY DETECTION USING SCENE CHANGE ANALYSIS

A. Statistical Analysis

We develop a theoretical basis for deriving thresholds for analyzing traffic and anomaly detection. Recent study has shown that Gaussian approximation should work well for aggregated traffic if the level of aggregation in the number of flows and observed time scales is high enough such that individual sources are swallowed due to Central Limit Theorem [7]. Our datasets satisfy the necessary criterion for the minimal level of such an approximation. Work in [6] has shown the possibility of analysis of WSS (wide-sense stationary) 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 the sampling rate is appropriately selected for generating images, for example 1 minute, we could acquire normally distributed and stationary images [29].

Our approach to detecting anomalies envisions two kinds of detection mechanisms, real-time and postmortem. Real-time analysis exploits the variance of pixel intensities of traffic image as a signal of traffic distribution; on the other hand, postmortem analysis employs the variance of 16 DCT components of 8-by-8 DCT of traffic image for analyzing the distribution of traffic.

A.1 Real-time analysis

Real-time analysis may rely on less sophisticated analysis because of the resource demands and imminence of attacks. We employ a light-weight approach of using the variance of pixel intensities in the image for real-time analysis and anomaly detection, which is denoted by distribution signal S_{σ} . Using the variance of these image signals for deriving thresholds, we can obtain an approximation of the energy distribution of the normalized packet counts within the observation domain as follows.

$$S_{\sigma} = \left[\frac{1}{N} \sum_{k=1}^N (x_k - \bar{x})^2 \right]^{\frac{1}{2}} \tag{3}$$

, where $\begin{cases} x_k \text{ are pixel intensities, } N=1024 \text{ in real-time analysis} \\ x_k \text{ are DCT components, } N=16 \text{ in postmortem analysis} \end{cases}$

$$\text{and } \bar{x} = \frac{1}{N} \sum_{k=1}^N x_k$$

The detection signal is calculated instantaneously each

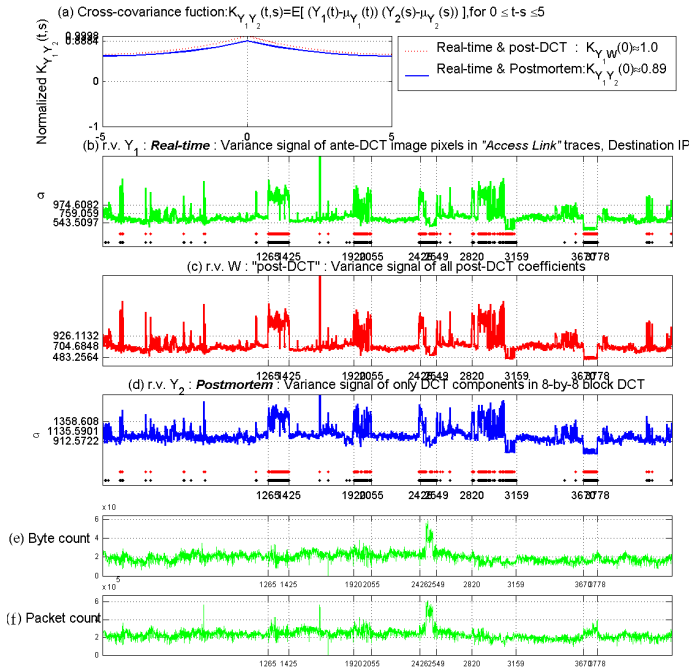


Fig. 8. The trace-driven evaluation results from address-based image signal in destination address for detecting attacks.

The (b) sub-picture 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 horizontal dotted 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) sub-picture shows the cross-correlation coefficients between these signals.

sampling instant for real-time analysis. Since our technique exploits the distribution property within each sample of traffic independently, the techniques are not impacted by the diurnal property of traffic over time. Fig. 8(b) shows the detection results based on such a signal from KREONet2 trace-driven evaluation for 8 days.

A.2 Postmortem analysis

We employ the Discrete Cosine Transform (DCT) for scene change analysis in postmortem. We employ 8-by-8 blocks for DCT. We transform the sixteen 8-by-8 blocks of Fig. 3(a) using DCT. The DCT tends to concentrate information in the upper-left corner of the DCT matrix. 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. Without losing 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.

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. The similarity of two stationary random processes can be estimated using cross-covariance function

which is the cross-correlation of mean-removed sequence as follows.

$$K_{XY}(t,s) = E\left[\left(X(t) - \overline{X(t)}\right)\left(Y(s) - \overline{Y(s)}\right)\right] \quad (4)$$

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

If all of the DCT coefficients were used in the detection, this signal, which is denoted by a random variable W , would be approximately equivalent to the variance of pixels in real-time, denoted by random variable Y_1 , (as in Parseval's Theorem) as shown in $K_{Y_1 W}(0) \approx 1.0$ in Fig. 8(a), and the variance signals in Fig. 8(b) and 8(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. For simplicity, we can use only the most significant DCT coefficient ($n=1$), called DCT component, in an 8-by-8 DCT block. It has the energy packing efficiency of about 0.8 [17]. Using the variance of only DCT components in the DCT blocks, we can obtain an approximation of the energy distribution in postmortem analysis, denoted by random variable Y_2 , as shown in Fig. 8(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 Fig. 8(a), $K_{Y_1 Y_2}(0) \approx 0.89$ and hence the approximated variance signal in Fig. 8(d) would classify nearly as well as the original signal.

A.3 Thresholds Setting through Statistical Analysis

To model the distribution of normal traffic, we select only the sampling points with ambient traffic, free of attacks, as samples and look at some statistical properties. As a quantitative example, we illustrate the distribution of normal traffic in postmortem. Fig. 9 shows the histogram and normal probability plot of the variance of 16 DCT components based on the ambient KREONet2 traces. And we verify normality of fit to a normal distribution with unspecified mean and variance. Suppose $X(t)$ (or $Y(t)$) is a stationary random process defined by standard deviation of the 16 DCT components of the source (or destination) address images. The standard deviation data (i.e., expressed in random variable X or Y) have a normal distribution at 5% significance level, namely $X \sim N(2480, 50^2)$ in source addresses and $Y \sim N(2265, 50^2)$ in destination addresses. When the random variable X possesses mean μ_x and variance σ_x^2 , and the random variable Y has mean μ_y and variance σ_y^2 , the probability density function (PDF) can be expressed as follows.

$$f_0(x) = f_0(x|H_0) = \frac{1}{\sigma_x \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x - \mu_x}{\sigma_x}\right)^2\right] \approx \frac{1}{50\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x - 2480}{50}\right)^2\right] \quad (5-1)$$

$$f_0(y) = f_0(y|H_0) = \frac{1}{\sigma_y \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y - \mu_y}{\sigma_y}\right)^2\right] \approx \frac{1}{50\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y - 2265}{50}\right)^2\right] \quad (5-2)$$

For 3σ -based statistical analysis, we set 2 kinds of thresholds,

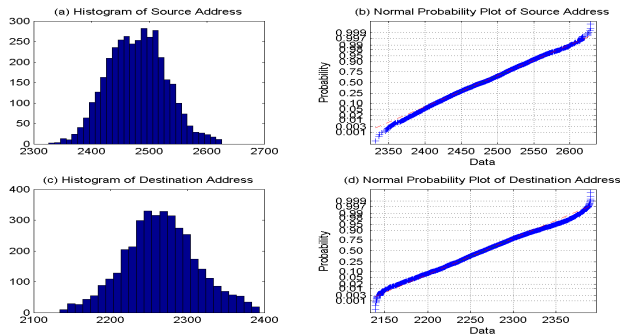


Fig. 9. The distribution of the variances of DCT components in normal network traffic in postmortem.

The data S_σ computed by (3) are respectively sampled from source and destination address-based image signals upon normal traffic free of attacks over 2 weeks of real traces. They correspond to the distribution of $f_0(x|H_0)$ and $f_0(y|H_0)$ in Fig. 13(a) and 13(b) respectively. The sampled data are displayed in a normal probability plot for graphical normality testing using a mathematical tool. If the data in (b) and (d) comes from a normal distribution, the plot will appear linear. Other probability functions will introduce curvature in the plot. The parameters for determining normality such as linear interpolation and cutoff value passed or closed to the criteria.

a high threshold T_H and a low threshold T_L . When we respectively set the T_H and T_L thresholds to $\pm 3.0\sigma$ of aforementioned traffic signal distributions in the ambient traffic, attacks can be detected with an error rate of 0.3% (if the signal is normally distributed) which can be expected as target false alarm rate by (6-1). We can judge the current traffic status by calculating the standard intensity deviation of signals in each sampling instant by (6-2).

$$X \sim N(\mu, \sigma^2) \rightarrow P(\mu - 3.0\sigma < X \leq \mu + 3.0\sigma) \approx 99.7\% \quad (6-1)$$

$$\text{traffic status} \begin{cases} \text{semi-random,} & \text{if } T_H \leq S_\sigma \\ \text{normal,} & \text{if } T_L < S_\sigma < T_H \\ \text{random,} & \text{if } S_\sigma \leq T_L \end{cases} \quad (6-2)$$

A.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 (pixel intensities in real-time processing, DCT coefficients in postmortem) 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 source address domain analysis, P is originally 2^{32} , reduced to $4 * 256$ (4 bytes of IP address * 256 values for each byte) = 1024. In 2-dimension (source, destination) images, P is reduced from 2^{64} to 256K. S is originally $32 * 32$, reduced to 1024 in real-time and 16 in postmortem. 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. 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 pcap or NetFlow type records in post-mortem, or work with more aggregate data upon packet

arrival in real-time.

B. Anomaly detection

If the variance S_σ 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 (6-2). The real-time detection requires that the analysis and the detection mechanism rely on small datasets in order to keep such on-line analysis feasible. Our detection signal can be calculated instantaneously at the sampling instants. Fig. 8(b) and 8(d) show the results from real trace-driven evaluation for 8 days in real-time and in postmortem, respectively. The major real attacks assail between the vertical lines and the resulting detection signal is shown with red dots located at 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. On the other hand, as the bottom 2 sub-pictures in Fig. 8 show, the approach using traffic volume alone itself, such as byte counts and packet counts, doesn't appropriately detect these attacks. Even when attack traffic may not induce significant overshoot in traffic volume (merely replacing existing normal traffic), these observations illustrate that anomaly detection may be feasible by studying the distributions of aggregate traffic.

C. 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 algorithm, 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. 10 illustrates the intermediate results in each sequence based on the destination IP address of the 25th frame in Fig. 5(b).

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.

$$\frac{|count[i][j][n] - count[i][j \pm 1][n]|}{count[i][j][n]} \geq 1.0 \quad (7)$$

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

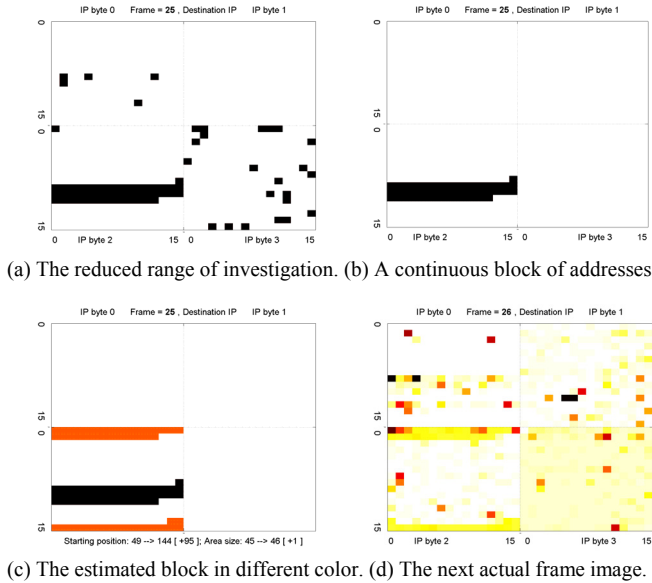


Fig. 10. An illustrative procedure showing potential attack estimation using motion prediction.

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 [18]. 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. 10. Fig. 10(a) shows the non-filtered pixels from the complexity reduction with starting pixel data and Fig. 10(b) shows the identified block of addresses. Fig. 10(c) shows the result of motion prediction (in red pixels) indicating the next set of addresses that may be a target of this attack. Fig. 10(d) is the actual traffic data for the next sampling point, validating the utility of such motion prediction techniques.

D. Identification

D.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-dimension image, we can be informed of the concentration of the attack. Through a 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 (8).

$$C_{ijn} = \frac{count[i][j][n]}{\sum_{j=0}^{255} count[i][j][n]} * \frac{count[i][j][n-1]}{\sum_{j=0}^{255} count[i][j][n-1]}, i=0,1,2,3, j=0,...,255 \quad (8)$$

We define delta as the difference of normalized packet counts by (9).

$$\Delta p_{ijn} = \frac{count[i][j][n]}{\sum_{j=0}^{255} count[i][j][n]} - \frac{count[i][j][n-1]}{\sum_{j=0}^{255} count[i][j][n-1]}, i=0,1,2,3, j=0,...,255 \quad (9)$$

Correlation is calculated by (8), possession ratio by (2), and delta by (9). Delta is remarkable at the instant of beginning and the end of attacks. Correlation of each pixel would have probability of $(1/256)^2$ in case of perfectly uniform distribution. We set 3.8% (flexible according to administrator’s criteria) as correlation thresholds, which means the corresponding IP address, on an average, successively sends 50 times as many packets as in the evenly distributed case. Once an 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 for the anomalies over the four byte-segment levels independently as shown in the upper part of Fig. 11.

```
*****
[ 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. 11. The detection report for the Fig.5 of anomaly identification. ¹

¹ For privacy, the IP addresses in Fig. 11 are appropriately sanitized.

“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.

D.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 a 4-byte (whole) IP address.

First, along with our image data representation, we employ 4 independent hash functions, h_1, h_2, h_3, h_4 , each with range $\{1, \dots, m\}$ as a Bloom filter [19]. For each IP address a_m 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. 11.

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

VII. COMPARISON WITH NP TEST

A. NP Test

In order to verify the inherent strengths of various image signals and the effectiveness of 3σ -based statistical analysis, we compare statistical analysis with the classical and well-established detection theorem, Neyman-Pearson (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. 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 a priori classified datasets for modeling the distributions of the noise and the signal. For anomaly detection purposes, NP-test would require samples of attack traffic as well as normal traffic.

A.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 [20], 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 (N), H_0 : represents the null hypothesis or the normal network traffic. The probability density under H_0 is represented by $P(X=x|H_0) = P(x|H_0)$, where X is a random variable denoting the observation.

(ii) Noise with signal (N+S), 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 $P(X=x|H_1) = P(x|H_1)$.

We can define the sample space of the two-lateral signals as 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, a solid model of normal behavior is required. We look at some statistical properties of aforesaid feasible signals in the normal mode i.e., normal traffic free of attacks. Based on the probability distribution, we assume that the short-term network traffic S_0 exhibits approximately normal distribution as shown in (5-1) and (5-2), namely H_0 : $X \sim N(\mu_{XN}, \sigma_{XN}^2)$ in source domain of image-based signals, and H_0 : $Y \sim N(\mu_{YN}, \sigma_{YN}^2)$ in destination domain. The probability density functions (PDF) of H_0 can be expressed as follows.

$$f_0(x|H_0) = \frac{1}{\sigma_{XN}\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu_{XN}}{\sigma_{XN}}\right)^2\right] \quad (10-1)$$

$$f_0(y|H_0) = \frac{1}{\sigma_{YN}\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-\mu_{YN}}{\sigma_{YN}}\right)^2\right] \quad (10-2)$$

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 source domain, the distribution of traffic under H_1 could be considered as approximately normal distribution with large variance. Source domain has unimodality as $f_1(x|H_1)$ in Fig. 13(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 $f_1(y|H_{1,H})$ in Fig. 13(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. 12(a) and 12(b) illustrate two normal probabilities of abnormal traffic in destination address-based signal Y . The histogram

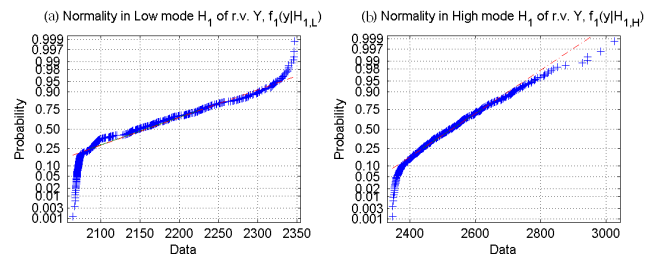


Fig. 12. The distribution of the variances of DCT components in abnormal network traffic.

The data (a) and (b) are respectively excerpted from low and high mode data of destination address-based image signals from abnormal traffic with attacks over the same periods of Fig. 9. They correspond to the distribution of $f_1(y|H_{1,L})$ and $f_1(y|H_{1,H})$ in Fig. 13(b) respectively.

Data of source address-based image signals in abnormal traffic with attacks, $f_1(x|H_1)$, in Fig. 13(a) is not shown here, but shows approximately Gaussian distribution with large variance.

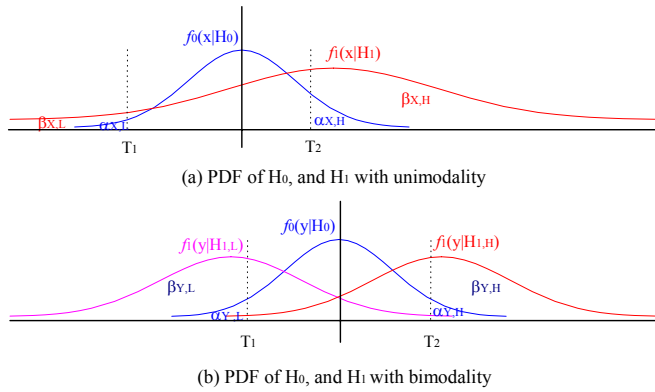


Fig. 13. Illustration of PDF and true / false positive rates.

indicates that the non-parametric data might be appropriately fitted with a mixture of two normal distributions with the different means and standard deviations as follows [21, 22].

$$f_1(y|H_1) = \varepsilon \phi_1 + (1-\varepsilon) \phi_2 = \varepsilon f_1(y|H_{1,L}) + (1-\varepsilon) f_1(y|H_{1,H}) \quad (11)$$

, where ε is mixing proportion

ϕ_1, ϕ_2 are normal PDFs with location and scale parameters $\mu_1, \sigma_1, \mu_2, \sigma_2$.

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 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. The PDF under H_1 can be expressed as follows.

$$f_1(x|H_1) = \frac{1}{\sqrt{2\pi} \sqrt{\sigma_{XN}^2 + \sigma_{XS}^2}} \exp \left[-\frac{1}{2} \frac{(x - (\mu_{XN} + \mu_{XS}))^2}{\sigma_{XN}^2 + \sigma_{XS}^2} \right] \quad (12-1)$$

$$f_1(y|H_1) = \varepsilon^* \frac{1}{\sqrt{2\pi} \sqrt{\sigma_{YN}^2 + \sigma_{YLS}^2}} \exp \left[-\frac{1}{2} \frac{(y - (\mu_{YN} + \mu_{YLS}))^2}{\sigma_{YN}^2 + \sigma_{YLS}^2} \right] + (1-\varepsilon)^* \frac{1}{\sqrt{2\pi} \sqrt{\sigma_{YN}^2 + \sigma_{YHS}^2}} \exp \left[-\frac{1}{2} \frac{(y - (\mu_{YN} + \mu_{YHS}))^2}{\sigma_{YN}^2 + \sigma_{YHS}^2} \right] \quad (12-2)$$

A.2 Bayes' Likelihood Ratio Test

The Bayesian criterion method assumes that the two outputs are governed by apriori probabilities, π_0 and π_1 , which are $P(H_0 \text{ is true})$ and $P(H_1 \text{ is true})$ respectively, and that a cost is assigned to each of the four outcomes. 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 (α), true positive (β) and false negative.

Bayesian criterion leads to likelihood ratio test [23], 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 source domain X of image-based signals as follows.

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

$$\begin{cases} \text{if } \Lambda(x) \geq \eta, \text{ announce } H_1 \\ \text{if } \Lambda(x) < \eta, \text{ announce } H_0 \end{cases} \quad (13-1)$$

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

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

$$\begin{cases} \text{if } \Lambda(y) \geq \eta, \text{ announce } H_1 \\ \text{if } \Lambda(y) < \eta, \text{ announce } H_0 \end{cases} \quad (13-2)$$

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

$$\eta = \frac{\pi_0(C_{10} - C_{00})}{\pi_1(C_{01} - C_{11})} = \frac{\pi_0}{\pi_1}, \text{ when } \alpha + \beta = 1 \quad (14)$$

Threshold η can be defined by the ratio of $P(H_0 \text{ is true})$ and $P(H_1 \text{ is true})$, 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 hence making it difficult to set a proper threshold. If we do not know an empirical distribution, we could virtually set η as 1 for eliminating the effects of the factor by logarithmic vanishing. On the other hand, the Neyman-Pearson (NP) test bypasses these factors by introducing the conditional probabilities as (15-1), (15-2), (16-1) and (16-2). 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 . The NP-test is optimal in the sense that it maximizes the probability of detection for a fixed probability of false alarm.

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 a statistical threshold of 3σ . Given the threshold, we solve the Bayesian detector in (13-1) and (13-2), and derive critical regions (Z_1) of either boundary (T_1 and T_2) in Fig. 13. These critical regions are close to those of 3σ -based method due to normality.

A.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 the likelihood ratio test detects H_1 as (15-1) and (15-2) from blue-colored PDFs of Fig. 13.

$$\alpha_X = \int_{Z_1} f_0(x|H_0) dx = \int_{-\infty}^{T_1} f_0(x|H_0) dx + \int_{T_2}^{\infty} f_0(x|H_0) dx \quad (15-1)$$

$$\alpha_Y = \int_{Z_1} f_0(y|H_0) dy = \int_{-\infty}^{T_1} f_0(y|H_0) dy + \int_{T_2}^{\infty} f_0(y|H_0) dy \quad (15-2)$$

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

And the detection rate β in 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 (16-1) from red-colored PDF of Fig. 13(a). Consequently false negative rate (type II error) is calculated as $1-\beta$. Similarly, the true positive rates in destination domain of image-based signals with bimodality can be defined as (16-2) from red/pink-colored PDFs of Fig. 13(b).

$$\beta_X = \int_{Z_1} f_1(x|H_1) dx = \int_{-\infty}^{T_1} f_1(x|H_1) dx + \int_{T_2}^{\infty} f_1(x|H_1) dx \quad (16-1)$$

$$\beta_Y = \int_{Z_1} f_1(y|H_1) dy = \varepsilon * \left[\int_{-\infty}^{T_1} f_1(y|H_{1,L}) dy + \int_{T_2}^{\infty} f_1(y|H_{1,L}) dy \right] + (1-\varepsilon) * \left[\int_{-\infty}^{T_1} f_1(y|H_{1,H}) dy + \int_{T_2}^{\infty} f_1(y|H_{1,H}) dy \right] \quad (16-2)$$

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 the derived density function and the given thresholds, we can induce the expected true positive rates and false positive rates of each feasible traffic signal.

A.4 Application of traffic signals in NP-test

For explaining how the NP-test is applied, we exemplify the real-time image-based signal of flow distribution in destination address domain in Table 1. Through analysis of image-based signals computed by (3), the distribution of H_0 shows an approximately normal distribution at 5% significance level, namely $Y \sim N(910.9, 146.5^2)$ in destination address. Similarly, 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 ratio of low mode and high mode is 0.27 and 0.73 from histogram.

Under the given threshold and PDFs, defined as (10-2) and (12-2), we solve the NP-test as (13-2) and derive critical regions of either boundary.

For destination address variable Y

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

through numerical analysis

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

$$\begin{cases} Z_0: 485 < y < 1335 \\ Z_1: y \leq 485, 1335 \leq y \end{cases} \quad (17)$$

These critical regions are close to those of 3σ -based method, $471 < y < 1350$.

We can compute the false alarm rate α (type I error) as the interval probability distribution from (15-2) as,

$$\alpha_Y \approx \int_{-\infty}^{485} \frac{1}{146.5\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-910.9}{146.5}\right)^2\right] dy + \int_{1335}^{\infty} \frac{1}{146.5\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-910.9}{146.5}\right)^2\right] dy \approx 0.0037 \quad (18)$$

Similarly, the detection rate β is calculated as a mixture of two interval probabilities from (16-2) as,

$$\beta_Y \approx 0.27 \left\{ \int_{-\infty}^{485} \frac{1}{45.0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-405.9}{45.0}\right)^2\right] dy + \int_{1335}^{\infty} \frac{1}{45.0\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-405.9}{45.0}\right)^2\right] dy \right\} + 0.73 \left\{ \int_{-\infty}^{485} \frac{1}{462.2\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-1518.4}{462.2}\right)^2\right] dy + \int_{1335}^{\infty} \frac{1}{462.2\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-1518.4}{462.2}\right)^2\right] dy \right\} \approx 0.746 \quad (19)$$

For the real-time flow-based signal in the destination address domain, the false alarm rate is about 0.37% and the detection rate is 74.6%, as shown in Table 1. With expected detection rates and false alarm rates, we can evaluate the innate power of feasible traffic signals.

B. Flow Distribution in 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 / destination port, the flood-based attacks spread flows over the destination IP addresses (or ports) in random or dictionary mode style attacks. 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 1. As shown in the Table 1, 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.

TABLE 1. RESULTS OF FLOW-BASED SIGNALS.²

| Time | D. | TP β^1 | FP α^2 | NP β^3 | NP α^4 | LR ⁵ | NLR ⁶ |
|-------------|--------------------------|------------------|------------------|-----------------|---------------|-----------------|------------------|
| Real-time | SA ⁷ | 90.3% 706/782 | 0.22% 8/3563 | <i>90.0%</i> | <i>0.14%</i> | 402.1/ 663.8 | 0.10/ 0.10 |
| | DA ⁸ | 56.5% 442/782 | 0.25% 9/3563 | <i>74.6%</i> | <i>0.37%</i> | 223.8/ 201.1 | 0.44/ 0.25 |
| | (SA, DA) ⁹ | 92.8% 726/782 | 0.42% 15/3563 | – | – | 220.5 | 0.07 |
| | Post mortem | SA | 91.6% 716/782 | 0.20% 7/3563 | <i>91.8%</i> | <i>0.14%</i> | 466.0/ 676.6 |
| Post mortem | DA | 52.0% 407/782 | 0.17% 6/3563 | <i>70.7%</i> | <i>0.40%</i> | 305.9/ 178.7 | 0.48/ 0.29 |
| | (SA, DA) | 94.8% 741/782 | 0.20% 7/3563 | – | – | 482.3 | 0.05 |

- True Positive rate (β) by 3σ -based statistical analysis
- False Positive rate (α) by 3σ -based statistical analysis
- expected true positive rate (β) by Neyman-Pearson test
- expected false positive rate (α) by Neyman-Pearson test
- Likelihood Ratio is measured by β/α , ideally infinity, in measurement by LR in 3σ / LR in NP-test
- Negative Likelihood Ratio is measured by $1-\beta/1-\alpha$, ideally zero, by NLR in 3σ / NLR in NP-test
- SA stands for Source Address.
- Destination Address.
- Source Address and Destination Address in combination.

² Statistical analysis is in non-italic type whereas NP-test is in italic type.

VIII. CONCLUSIONS

In this paper, we have presented an approach which represents traffic data as images or frames at each sampling point. Based on such a visual transformation, the door for applying techniques from image and video processing for the analysis of network traffic has been opened. 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 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 classical NP-test from detection theory to evaluate the effectiveness and quantitative evaluation of different image-based traffic signals.

REFERENCES

- [1] David Moore, Colleen Shannon, Geoffrey M. Voelker, and Stefan Savage, "Internet Quarantine: Requirements for Containing Self-Propagating Code", in *INFOCOM 2003 conference*, April, 2003.
- [2] Dave 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.
- [3] 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.
- [4] A. Kuzmanovic and E. Knightly, "Low-Rate TCP-Targeted Denial of Service Attacks", in *Proc. of ACM SIGCOMM 2003*, Karlsruhe, Germany, August 2003.
- [5] 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.
- [6] Seong Soo Kim, A. L. Narasimha Reddy and Marina Vannucci, "Detecting traffic anomalies through aggregate analysis of packet header data", in *Proc. of Networking 2004, LNCS vol. 3042, pp 1047-1059*, Athens, Greece, May 2004.
- [7] Jorma Kilpi and Ilkka Norros, "Testing the Gaussian approximation of aggregate traffic", in *Proceedings of ACM SIGCOMM Internet Measurement Workshop (IMW) 2002*, Marseille, France, November 2002.
- [8] P. Barford and D. Plonka, "Characteristics of Network Traffic Flow Anomalies", in *Proc. of ACM SIGCOMM Internet Measurement Workshop (IMW) 2001*, October, 2001.
- [9] N. Weaver, S. Staniford and V. Paxson, "Very fast containment of scanning worms", in *Proc. of Usenix Security Symposium*, Aug. 2004.
- [10] B. Krishnamurthy, S. Sen, Y. Zhang and Y. Chen, "Sketch-based Change Detection: methods, Evaluation, and Applications", in *Proceedings of ACM SIGCOMM Internet Measurement Conference (IMC) 2003*, Miami, USA, October 2003.
- [11] Hyogon Kim, Inhye Kang, and Saewoong Bahk, "Real-time Visualization of Network Attacks on High-speed Link", *IEEE Network Magazine*, Sept.-Oct. 2004.
- [12] Dan Lelescu and Dan Schonfeld, "Statistical Sequential Analysis for Real-time Video Scene Change Detection on Compressed Multimedia Bitstream", *IEEE Transactions on Multimedia*, vol. 5, issue 1, pp 106-117, 2003.
- [13] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of Full-motion Video", *Multimedia Systems*, vol. 1, no. 1, pp 10-28, 1993.
- [14] R. Lienhart, C. Kuhmunch, and W. Effelsberg, "On the Detection and Recognition of Television Commercials", in *Proc. Of the International Conference on Multimedia Computing and Systems*, pp 509-516, Ottawa, Canada, 1997.
- [15] K. Shen and E. J. Delp, "A fast Algorithm for Video Parsing Using MPEG Compressed Sequences", in *IEEE Conference on Image Processing*, pp 252-25, 1995.
- [16] A. Hussein, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks", in *Proc. of ACM SIGCOMM 2003*, Karlsruhe, Germany, August 2003.
- [17] J. D. Gibson, T. Berger, T. Lookabaugh, R. Baker, and D. Lindbergh, *Digital Compression for Multimedia*, Morgan Kaufmann Series, 1st Edition, pp. 247, 1998.
- [18] Gyaourova, A., C. Kamath, and S.-C. Cheung, "Block matching for object tracking", LLNL Technical report, October 2003. UCRL-TR-200271.
- [19] Burton Bloom, "Space/time trade-offs in hash coding with allowable errors", *Communications of ACM*, 13(7), pp 422-426, July 1970.
- [20] H. Vincent Poor, *An Introduction to Signal Detection and Estimation*, Springer Press, 2nd Edition, pp. 11, 1994
- [21] NIST/SEMATECH e-Handbook of Statistical Methods. Available : <http://www.itl.nist.gov/div898/handbook/>
- [22] Emanuel Parzen, "On Estimation of a Probability Density Function and Mode", *The Annals Mathematical Statistics*, Vol. 33, No. 3, pp 1065-1076, September 1962
- [23] Robert V. Hogg and Allen T. Craig, *Introduction to mathematical statistics*, Macmillan Company, 2nd Edition, pp. 285, 1965.
- [24] C. C. Zou, W. Gong and D. Towsley, "Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense", in *Proceedings of ACM CCS Workshop on Rapid Malcode (WORM'03)*, Washington, DC, USA, October 2003.
- [25] J. Mirkovic, G. Prier and P. Reiher, "Attacking DDoS at the Source", in *10th IEEE International Conference on Network Protocols*, Paris, France, November 2002.
- [26] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms", in *Computer Communications Review (CCR)*, Vol. 34, No. 2, April 2004.
- [27] A. Lakhina, M. Crovella and C. Diot, "Mining Anomalies Using Traffic Feature Distributions", in *Proc. of ACM SIGCOMM 2005*, Philadelphia, PA, USA, August 2005.
- [28] Seong Soo 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.
- [29] Seong Soo Kim and A. L. N. Reddy, "Modeling Network traffic as Images", in *Proc. of IEEE ICC 2005*, Seoul, Korea, May, 2005.



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, and the Ph.D. degree in computer engineering in the Department of Electrical and Computer Engineering at Texas A&M University in May 2005. 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. After 3-month Post Doc. course, he is currently working as a principal engineer at Samsung Electronics Co., Ltd. in Korea. His research interests are in computer

network security, wireless networking, 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.



A. L. Narasimha Reddy received his B.Tech degree in Electronics and Electrical engineering from the Indian Institute of Technology, Kharagpur, India, in August 1985, and the M.S. and Ph.D. degrees in Computer Engineering from the University of Illinois at Urbana-Champaign in May 1987 and August 1990, respectively. At the University of Illinois at Urbana-Champaign, he was supported by an IBM Fellowship. He is currently a professor in the Department of Electrical and Computer Engineering at Texas A & M University. He was a research staff member at IBM Almaden Research Center in San Jose from August 1990 to August 1995. Reddy's

research interests are in network security, network QoS, multimedia, I/O systems and computer architecture. Currently, he is leading projects on building scalable network security solutions and wide area storage systems. Prof. Reddy is a member of ACM SIGARCH and is a senior member of IEEE Computer Society. He received a US National Science Foundation CAREER Award in 1996. He received outstanding professor awards at Texas A & M University during 1997-1998 and 2003-2004.