A Study of Analyzing Network traffic as Images in Real-Time

Seong Soo Kim and A. L. Narasimha Reddy
Department of Electrical Engineering
Texas A&M University
College Station, TX 77843-3128, USA
{skim, reddy}@ee.tamu.edu

Abstract—**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. 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 NetViewer with classical detection theory based Neyman-Pearson test and an IDS tool.**

Keywords-Network measurements; Experimentation with real networks/Testbeds; Stochastic processes; Statistics.

I. INTRODUCTION

Increasing malicious network traffic, such as denial-of-service (DoS) floods, worms and other forms, have become serious threats to the network security. Network traffic monitoring and analysis tools are being employed to counter this threat. If efficient visual and analysis tools are available to network administrators, it could become possible to detect the attacks, anomalies and to appropriately take action to suppress the attacks before they have had much time to propagate across the network. We study the possibilities of traffic-analysis based visual mechanism for attack detection and identification.

To study and classify traffic on the network based on usage and protocols, a number of tools such as FlowScan [5], Cisco's FlowAnalyzer, and AutoFocus [1], 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 post-mortem 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 giving further information about possible changes in network traffic. Sophisticated low-rate attacks [2] 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. 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 passively monitors packet headers of network traffic at regular intervals and generates image of this packet header data. These 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 [3], and our previous work has shown the possibility of analysis of wide-sense stationary (WSS) property in network traffic [4]. Recent work in [7] 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 an anomalous manner. 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 visual 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.

In this paper, we will report our measurements conducted on real traces of traffic at three major networks. This paper will make the following significant contributions: (a) employing packet header data as images for traffic visualization, (b) employing image processing and compression techniques for efficiently storing and processing such visual data and (c) on the effectiveness of such measures in detecting and identifying the attacks in real-time with very small latencies.

II. RELATED WORK

A number of popular monitoring tools such as FlowScan, Cisco's FlowAnalyzer, and AutoFocus [1], are used as traffic analyzers. FlowSan is open source software to gather and analyze network flow data taken from NetFlow records of Cisco routers [5]. In the FlowScan, cflowd writes raw flow files that wait to be post-processed by flowscan for providing against heavy-traffic or flood-based DoS attacks. However, excessive backlog of flow files may cause to be placed in difficult real-time analysis. Using FlowScan, characteristics of network traffic flow anomalies are illustrated at flow level [6].

Recently, traffic volume has been analyzed using wavelets to detect anomalies in network traffic [3]. Our earlier work has considered correlation of addresses as a signal for analysis for anomaly detection [4]. While earlier work analyzed traffic as a

This work is supported by an NSF grant ANI-0087372, Texas Higher Education Board, Texas Information Technology and Telecommunications Taskforce and Intel Corp.

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 space. 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 [8]. Recent work in [23] 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 (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 [23] has some similarities (with significant differences in data representation and anomaly detection) to our work presented here.

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 [9], color histogram differences [10], characteristic patterns in the standard deviation of pixel intensities for detection of fades [11], and color histogram of DC coefficients [12]. 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 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, 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.

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 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 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.

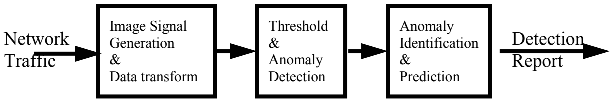

Network Traffic
Image Signal Generation & Data transform
Threshold & Anomaly Detection
Anomaly Identification & Prediction
Detection Report

Figure. 1. The block diagram of our tool.

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. Packet header traces or NetFlow records may be used as described in sections 4, 5 and 7.

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 Discrete Cosine Transform (DCT) coefficients and absolute difference of image pixels can be used for scene change analysis as presented in section 4 and 5.

The final stage is identification and prediction, in which attackers and victims are revealed using line or edge detection algorithms. Moreover, we estimate the movement of attack patterns using motion prediction algorithms as shown in section 6.

Finally, we compare NetViewer with well established NP test and popular Intrusion Detection System (IDS) Snort in sections 8 and 9.

3.1 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) [14], which contains real network attacks in the pcap header format. Additionally 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, to compare NetViewer with Snort (an IDS tool), we examine the tools on a live network in Texas A&M University (TAMU) campus.

IV. VISUAL MEASUREMENT OF THE NETWORK TRAFFIC

4.1 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).

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

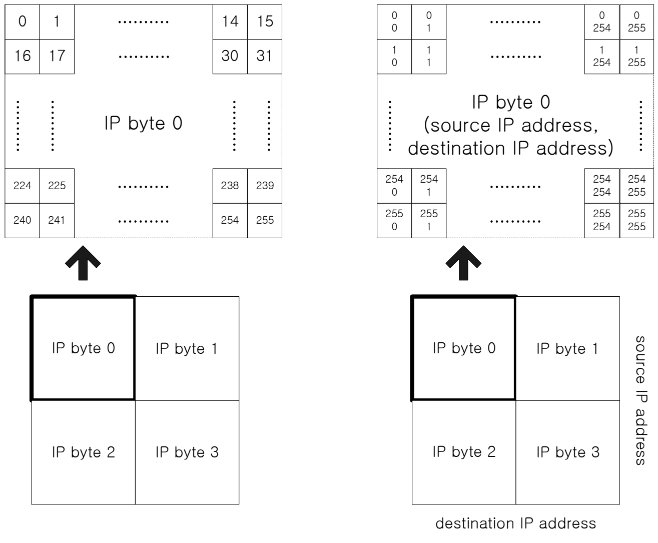
We employ a simpler alternative data structure as used in [4] for reducing the storage and computation complexity over 2^{32} discontinuous address space from O(n) to O(lgn). This data structure consists of 4 arrays "*count[4]*". Each array expresses one of the 4 bytes in an IP address structure. A location *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 (2).

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

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

4.2 DCT

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*16 square for visual representation at the sampling point. The 16*16 squares of each of the 4 bytes of IP address are organized as a frame for the source and destination addresses respectively as in Fig. 2(a). Similarly, with 256*256 squares, we can express the normalized values for the source and destination addresses simultaneously as in Fig. 2(b). Here the intensity (gray-level) of the pixel is directly proportional to the normalized packet count.



(a) 1 dimension

(b) 2 dimension

Figure 2. The visualization of network traffic signal in IP address

Each frame with four 16*16 squares is treated as one 32*32 block for DCT. We transformed the 32*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*32 DCT coefficients, we select only 4*4 coefficients in the upper-left corner. These coefficients can represent a good approximation of the energy in a sequence.

Using the variations of these 4*4 DCT coefficients for deriving thresholds, we can obtain an approximation of the energy distribution of the normalized packet counts within IP address domain as follows.

$$\sigma = \left[\frac{1}{16} \sum_{k=1}^{16} (x_k - \bar{x})^2 \right]^{\frac{1}{2}} \tag{3}$$

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

A study of required parameters for presenting network traffic as images like the number of DCT coefficients retained and sampling rate will be evaluated in the future work.

4.3 Thresholds Setting through 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 traffic 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 [4] 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.

To model the distribution of traffic, we select only ambient trace, free of attacks, as samples and look at some statistical properties. Fig. 3 shows the histogram and normal probability plot of the variations of 4*4 DCT coefficients based on the ambient KREONet2 traces. And we verify normality of these 2-week data through the Lilliefors test for goodness of fit to a normal distribution with unspecified mean and variance. Suppose X(t) (Y(t)) is a random process defined by standard deviation of the 4*4 DCT coefficients of the source (destination) address images. The variance data have a normal distribution at 5% significance level, namely X~N(2480, 50^2) in source addresses and Y~N(2265, 50^2) in destination addresses. When random variable X(t) possesses mean μ and

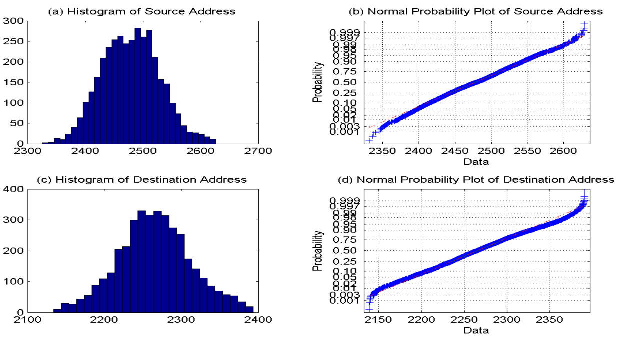


Figure 3. The distribution of the variances of 4*4 DCT coefficients in normal network traffic. If the data does come from a normal distribution, the plot will appear linear in (b) and (d). Other probability density functions will introduce curvature in the plot.

variance σ^2, the probability density function (pdf) can be expressed as follows.

$$f_0(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \approx \frac{1}{50\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-2480}{50}\right)^2\right] \qquad (4-1)$$

$$f_0(y) \approx \frac{1}{50\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2265}{50}\right)^2\right] \qquad (4-2)$$

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 4*4 DCT coefficients of each frame as (5).

$$\textit{traffic status} \begin{cases} \textit{semi-random}, & \text{if } \sigma > T_H \\ \textit{normal}, & \text{if } T_L \le \sigma \le T_H \\ \textit{random}, & \text{if } \sigma < T_L \end{cases} \qquad (5)$$

When we set the T_H and T_L thresholds to $\pm 3.0\sigma$ respectively, these figures of about 2330 and 2630 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.

$$P(\mu - 3.0\sigma < X \le \mu + 3.0\sigma) \approx 99.7\% \qquad (6)$$

This interval matches 99.7% confidence level by (6). With such thresholds, we can detect attacks with error rate of 0.3%, which can be expected as target false alarm rates.

V. ANOMALY DETECTION USING SCENE CHANGE ANALYSIS

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 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 generally at traffic as in normal behavior mode, in semi-random and in random attack modes.

5.1.1 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 Fig. 4(a) and 4(b) sub-pictures show the standard deviation of 4*4 DCT coefficients (by (3)) after DCT in source/destination addresses respectively.

The lower 3 sub-pictures visually illustrate the normalized packet counts as outlined in Fig. 2. 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 DCT of traffic frames maintain the middle level between the two anomalous cases in the Fig. 4(a) and 4(b) as set in (5).

5.1.2 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(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

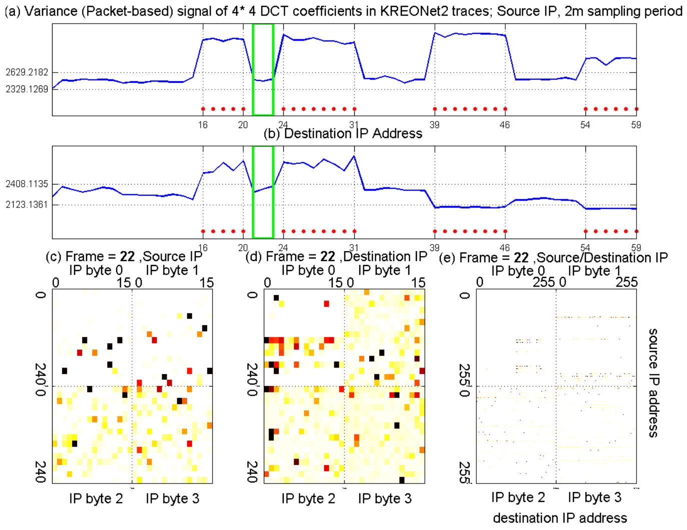


Figure 4. Visual measurement of normal network traffic.
A green rectangular time-window in (a) and (b) sub-pictures indicates the current sampling points. The bottom red dots in (a) and (b) illustrate the anomaly detection signals and the vertical lines are the periods of actual anomalies. The (c) and (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-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 unidimensional) regardless of color representation.

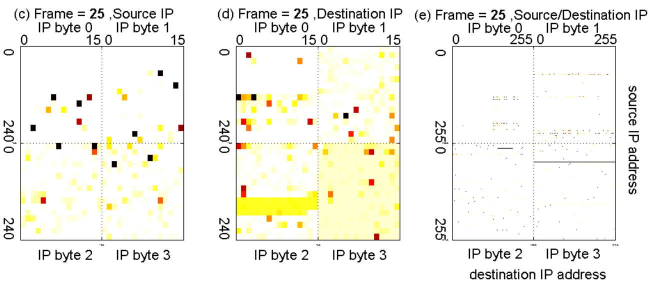


Figure 5. Visual measurement of semi-random typed attack

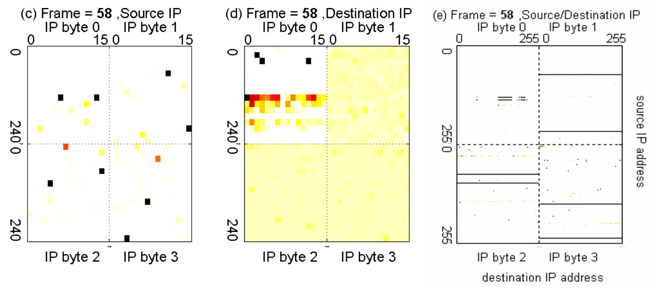


Figure 6. Visual measurement of (horizontal) random styled attack

the next potential attack using "motion prediction" in section 5.3. From the 3rd and 4th bytes of Fig. 5(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.

5.1.3 Visual patterns in random targeted attacks

Fig. 6 shows the traffic during a (horizontal) random attack. From Fig. 6(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. 6(e), it shows that two specific sources, i.e., two attackers (visible through black pixels in Fig. 6(c) and horizontal lines in 6(e)), scan all possible destinations. 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.

5.1.4 Visual patterns in complicated attacks

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.

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.

5.2 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 (5). 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. Results from real trace-driven evaluation for 8 days are shown in the top 2 sub-pictures of the Fig. 8. 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

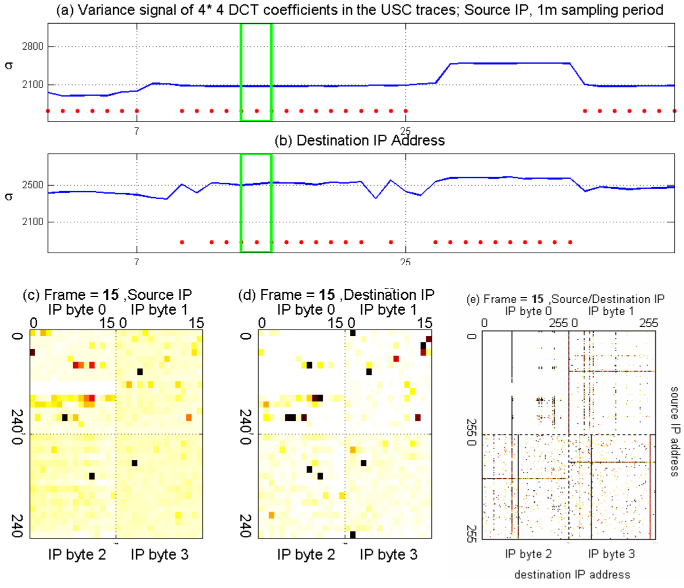


Figure 7. Visual measurement of mixed network attack.
The (c) sub-picture shows intensity of network traffic in a few of the source IP addresses. For example, the IP address 100 in 2nd byte, the 107 in 3rd byte, and the 67 in 4th byte can be considered as suspicious attack sources. From (e) sub-picture,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

confirm the existence of these attacks. On the other hand, as the bottom 2 sub-pictures show, the approach using traffic volume alone itself, such as byte count and packet count, 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.

5.3 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. 9 illustrates the intermediate results in each sequence based on the destination IP address of the 25th frame in Fig. 5(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.

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

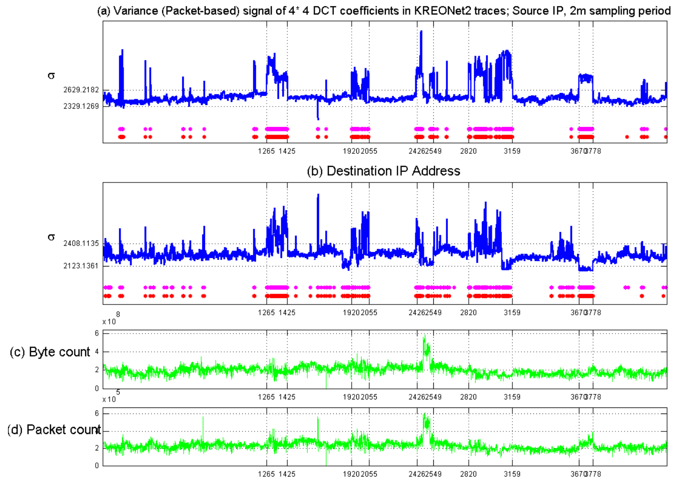


Figure 8. The results from trace-driven evaluation for detecting attacks. The magenta dots located on the top are marked when NP test declares anomalies. The Red dots located on the bottom show NetViewer's detection of abnormalities. The horizontal dotted lines in (a) and (b) show the T_H and T_L thresholds based on 3σ method.

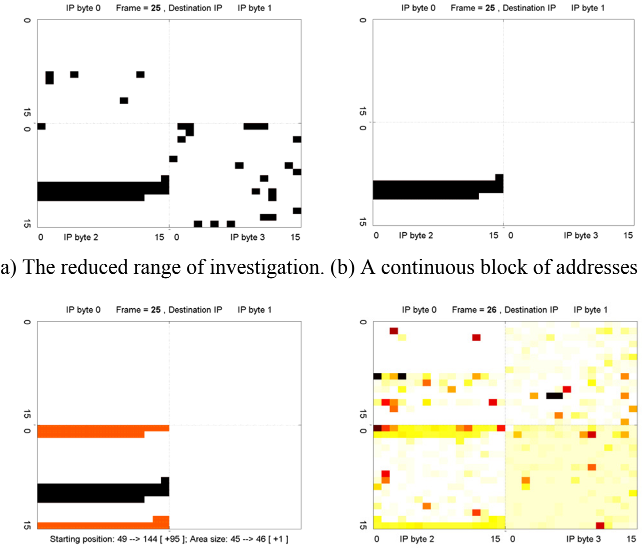
In the 2nd stage, to find a block of addresses, a continuity check is carried out. For improving the continuity, a few non-continuous pixels between continuous pixels, which results from the aperture problem, are considered as a portion of the continuous block. The aperture problem appears in situations 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 [13]. 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. 9. Fig. 9(a) shows the non-filtered pixels from the complexity reduction with starting pixel data and Fig. 9(b) shows the identified block of addresses. Fig. 9(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. 9(d) is the actual traffic data for the next sampling point, validating the utility of such motion prediction techniques.

5.4 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 (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



(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.

Figure 9. An illustrative procedure showing potential attack estimation using motion prediction

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 4*4 = 16. 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 pcap or NetFlow type records in post-mortem, or work with more aggregate data upon packet arrival in real-time.

VI. IDENTIFICATION

6.1 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 (8).

$$C_{ijn} = \frac{count[i][j][n]}{\sum_{j=0}^{255} count[i][j][n]} * \frac{count[i][j][n-1]}{\sum_{j=0}^{255} count[i][j][n-1]}, \begin{array}{l} i = 0,1,2,3 \\ j = 0,..,255 \end{array} \tag{8}$$

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

$$\Delta p_{ijn} = \frac{count[i][j][n]}{\sum_{j=0}^{255} count[i][j][n]} - \frac{count[i][j][n-1]}{\sum_{j=0}^{255} count[i][j][n-1]}, \begin{array}{l} i = 0,1,2,3 \\ j = 0,..,255 \end{array} \tag{9}$$

Correlation is calculated by (8), possession ratio by (2), and delta by (9). 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 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. 10.

"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.

6.2 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 as shown in Appendix A-2.

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,…, m} as a Bloom filter [22]. 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. 10.

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

```
***************************************************************
[ 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%
***************************************************************
```

Figure 10. The detection report for the Fig.5 of anomaly identification.[1]

[1] For privacy, the IP addresses in Fig. 10 are appropriately sanitized.

VII. MULTIDIMENSIONAL VISUALIZATION

Up to now, we have focused on normalized packet counts in the address domain. Besides normalized packet counts, we can use the number of flows and the correlation of the normalized packet counts over the address space for analyzing a variety of aspects of traffic, independently or jointly.[2] The various pieces of 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. The normalized distribution of the number of flows would be much different from its normal activity.

An analysis of the correlation based image gives an idea of continuance over specific address space. Correlation by (8) 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. 11 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 our previous work [4]. With the three distinct traffic signals, we can analyze the traffic properties of each IP address from diverse viewpoints.

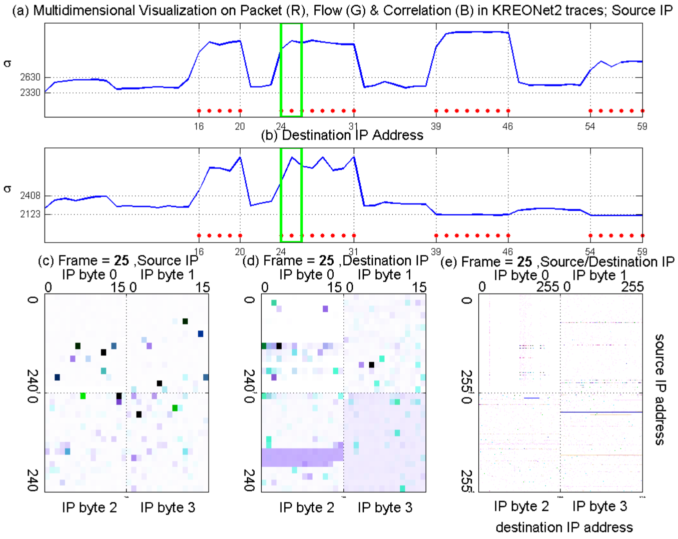


Figure 11. Multidimensional Visualization of semi-random attack.
The normalized packet count in IP address space is represented as the red, the number of flow as the green, and the correlation of the packet count as the blue components of the corresponding pixel image in (c), (d) and (e). The average 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.

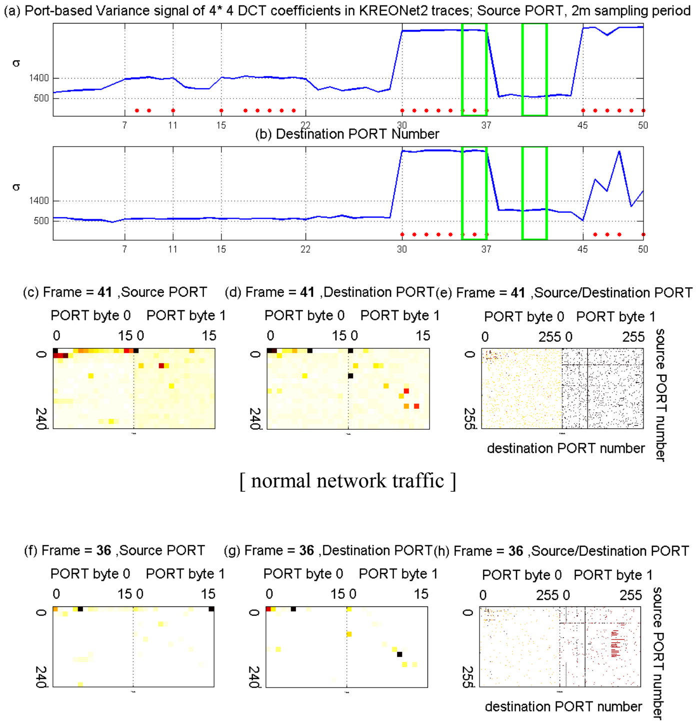
7.1 Visual patterns in port number domain

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. 12 shows the visual measurement of P_{ijn} in the source/destination port number domain for detecting portscans. Fig. 12(c) through 12(e) illustrate normal network traffic, where destination port #80 (visible through vertical line in Fig. 12(e)) occupies a large portion of traffic generally. On the other hand, Fig. 12(f) and 12(g) visualize that traffic concentrates from #1295 of source port to #1434 of destination port in SQL Slammer worm. Fig 12(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.



[normal network traffic]

[attack traffic: SQL Slammer worm]

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).

Figure 12. Port-based visual measurement

[2] 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.

VIII. COMPARISON WITH NP TEST

8.1 Distribution of H_0 and H_1

To verify our statistical thresholds approach, we compare it with classical and well-established mathematical detection theorem, Neyman-Pearson (NP) test in detection and estimation theory. In detection theory, data in observed interval represent either noise or noise with signal. Thus there are two statistical hypotheses [15].

(i) Noise only, H_0: represents the null hypothesis or the normal network traffic. The probability density under H_0 is represented by
$P(X = x|H_0) = P(x|H_0)$.

(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
$P(X = x|H_1) = P(x|H_1)$.

We want to develop an algorithm to choose between H_0 and H_1 and minimize the probability of error.

8.2 Probability density function of H_0 and H_1

The pdf of H_0 can exploit the (4-1) in source IP address and (4-2) in destination as follows.

$$f_0(x|H_0) = f_0(x) \qquad (10\text{-}1)$$

$$f_0(y|H_0) = f_0(y) \qquad (10\text{-}2)$$

To model the distribution of abnormal traffic, we excerpt only trace with attacks as samples and investigate the statistical measures. In the case of source address, the distribution of traffic under H_1 could be considered as approximately normal distribution with heavy-tail. In the case of destination addresses, however, the pdf shows shape close to a bimodal distribution. These two separated modes are located in the outlier of the normal distribution of H_0 as shown in data distributions of Fig. 3(d) and Appendix A-1. Each of the modes can be locally modeled to have a rough normal distributed component as shown in Appendix A-1. This non-parametric regression process might be appropriately fit with a mixture of two normal distributions with the different locations and standard deviations [16, 17]. The mixing proportion (between 0 and 1) can be fit using either least squares or maximum likelihood. We estimate the contamination in same probability from the histogram.

Through analysis of samples, we simplify the distribution of H_1, namely $X \sim N(2950,\ 195^2)$ in source addresses, and $Y_L \sim N(2090,\ 22^2)$ and $Y_H \sim N(2550,\ 122^2)$ in destination. The pdf under H_1 can be expressed as follows,

$$f_1(x|H_1) \approx \frac{1}{195\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-2950}{195}\right)^2\right] \qquad (11\text{-}1)$$

$$f_1(y|H_1) = \varepsilon f_1(y|H_{1,L}) + (1-\varepsilon) f_1(y|H_{1,H}), \quad \text{where } \varepsilon \text{ is contaminat ion}$$

$$\approx \frac{1}{2} * \frac{1}{22\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2090}{22}\right)^2\right] + \frac{1}{2} * \frac{1}{122\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2550}{122}\right)^2\right] \qquad (11\text{-}2)$$

8.3 Bayes' likelihood ratio test

The notion of using the magnitude of the ratio of two probability density functions as the basis of a best test or of a uniformly most powerful (UMP) test will help to provide an intuitively appealing method of constructing a test of a null hypothesis against an alternative hypothesis. This method leads to a test that is called likelihood ratio test [18] which is defined as,

$$\Lambda(x) = \frac{f_1(x|H_1)}{f_0(x|H_0)} = \frac{\frac{1}{195\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-2950}{195}\right)^2\right]}{\frac{1}{50\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-2480}{50}\right)^2\right]} \overset{H_1}{\underset{H_0}{\gtrless}} \eta \qquad (12)$$

For a practical system, given a particular 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. For example, when false alarm rate is constrained to 5%, the threshold is derived as 0.3 and corresponding detection rate reaches 97.4%. General optimal nonlinear test is performed as shown in Appendix A-3.

To evaluate our approach against NP-test, we calculate the false alarm rate and the detection rate based on a given threshold. A 6.0 as threshold of NP test corresponds to statistical thresholds of 3σ. Given the threshold, we solve the NP-test and derive critical regions of either boundary.

For source address variable X

$$\Lambda(x) = \frac{f_1(x|H_1)}{f_0(x|H_0)} = \frac{\frac{1}{195\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-2950}{195}\right)^2\right]}{\frac{1}{50\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-2480}{50}\right)^2\right]} = \eta = 6.0$$

$$\exp\left[-\frac{1}{2}\left\{\left(\frac{x-2950}{195}\right)^2 - \left(\frac{x-2480}{50}\right)^2\right\}\right] = 23.4$$

by taking ln

$$-\frac{1}{2}\left\{\left(\frac{x-2950}{195}\right)^2 - \left(\frac{x-2480}{50}\right)^2\right\} = \ln 23.4$$

by expanding

$$x^2 - 4894x + 5953928 = 0$$

$$x_{1,2} = 2447 \pm 184 \qquad (13\text{-}1)$$

$$\begin{cases} \text{if } 2263 \le x \le 2631, & \text{then} \quad H_0 \\ \text{else} & \qquad\quad\ H_1 \end{cases}$$

For destination address variableY

$$\Lambda(y) = \frac{\frac{1}{2}\frac{1}{22\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2090}{22}\right)^2\right] + \frac{1}{2}\frac{1}{122\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2550}{122}\right)^2\right]}{\frac{1}{50\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2265}{50}\right)^2\right]} = 6.0$$

through numerical analysis

$$y_{1,2} = 2270 \pm 138 \qquad (13\text{-}2)$$

$$\begin{cases} \text{if } 2132 \le y \le 2408, & \text{then} \quad H_0 \\ \text{else} & \qquad\quad\ H_1 \end{cases}$$

These critical regions are close to those of 3σ in (6). Results from real trace-driven evaluation for 8 days are shown in the top 2 sub-pictures of the Fig. 8. The magenta dots located on the top are marked when the NP detector declares anomalies. The 3σ method and NP detector achieve almost equivalent detection performance as shown in Fig. 8.

8.4 False alarm rate and detection rate

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,

$$\alpha_X = \int_{-\infty}^{T1} f_0(x|H_0)\,dx + \int_{T2}^{\infty} f_0(x|H_0)\,dx \qquad (14\text{-}1)$$
$$\approx \int_{-\infty}^{x1} \frac{1}{50\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-2480}{50}\right)^2\right] dx + \int_{x2}^{\infty} \frac{1}{50\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-2480}{50}\right)^2\right] dx$$

$$\alpha_Y \approx \int_{-\infty}^{y1} \frac{1}{50\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2265}{50}\right)^2\right] dy + \int_{y2}^{\infty} \frac{1}{50\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2265}{50}\right)^2\right] dy \quad (14\text{-}2)$$

Similarly, the detection rate β 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. Consequently false negative rate (type II error) is calculated as 1-β.

$$\beta_X = \int_{-\infty}^{T1} f_1(x|H_1)\,dx + \int_{T2}^{\infty} f_1(x|H_1)\,dx \qquad (15\text{-}1)$$
$$\approx \int_{-\infty}^{x1} \frac{1}{195\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-2950}{195}\right)^2\right] dx + \int_{x2}^{\infty} \frac{1}{195\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-2950}{195}\right)^2\right] dx$$

$$\beta_Y = \frac{1}{2}\left[\int_{-\infty}^{T1} f_1(y|H_{1,L})\,dy + \int_{T2}^{\infty} f_1(y|H_{1,L})\,dy\right] + \frac{1}{2}\left[\int_{-\infty}^{T1} f_1(y|H_{1,H})\,dy + \int_{T2}^{\infty} f_1(y|H_{1,H})\,dy\right]$$
$$\approx \frac{1}{2}\left\{\int_{-\infty}^{y1} \frac{1}{22\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2090}{22}\right)^2\right] dy + \int_{y2}^{\infty} \frac{1}{22\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2090}{22}\right)^2\right] dy\right\} +$$
$$\frac{1}{2}\left\{\int_{-\infty}^{y1} \frac{1}{122\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2550}{122}\right)^2\right] dy + \int_{y2}^{\infty} \frac{1}{122\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-2550}{122}\right)^2\right] dy\right\} (15\text{-}2)$$

According to the NP test, in the case of source IP addresses, the false alarm rate is about 0.2% and the detection rate is 94.9%. For destination addresses, the false alarm rate is about 0.6% and the detection rate is 92.5%. Due to the bimodality of destination addresses, the performance slightly degrades. This analysis shows that the source address based images/signal exhibits higher confidence than the destination address based images/signal for detecting traffic anomalies in our traces.

Actually, through image-based 3σ evaluation of source IP addresses in real-time, we achieve about 0.30% false alarm rate (11 false alarms out of 3616 sampling periods) and realize about 92.3% detection rate (detects 673 out of 729 suspicious symptoms). For destination addresses, the false positive rate is about 0.75% (27 out of 3616) and the true positive rate is 87.2% (636 out of 729).

Overall, the performances of 3σ bound can match those of the NP detector. However, 3σ approach does not require the analysis of the distribution of H_1 and can be more easily implemented. It means that the 3σ method is practical for monitoring traffic, especially online, when traffic is normally distributed in normal/attack scenarios. On the other hand, if traffic is not normally distributed, NP detector may give more quantitative and better results than the 3σ method. However, if the current attack's distribution of H_1 does not match previously observed distributions, the NP detector may not give correct results.

IX. COMPARISON WITH IDS

9.1 Intrusion detection system

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.

9.2 Snort

We review Snort as representative IDS [19, 20], 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) [21].

9.3 Overall results of Snort and NetViewer

Snort system reported 13,257 alerts distributed over the experiment time period as shown in Appendix A-4. Results from NetViewer based on normalized packet counts are shown in the top 2 sub-pictures in Appendix A-5. In the trace, it is apparent that there are continuous anomalies over almost the entire time period. This detection result agrees with that of Snort.

9.4 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.

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, 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 [20]. 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.

X. FUTURE WORK AND CONCLUSIONS

In this paper, 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 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 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 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 develop a network processor based system for testing the practical feasibility of our approach, plan to release our tool to general public and combine it with an IDS tool such as Snort in the future.

ACKNOWLEDGMENT

We are very grateful to Prof. Don R. Halverson for his comments and reviews on statistical analysis, to Roger Heiniluoma for his assistance on the Snort experiment, to Man Hee Lee and Dr. Okhwan Byeon at KISTI for their help in accessing KREONet2 traces. Thanks to Alefia Hussain, John Heidemann and Christos Papadopoulos at USC for sharing USC trace data.

REFERENCES

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

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

[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] 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.

[9] 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.

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

[11] 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, pp 509-516,* Ottawa, Canada, 1997.

[12] 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.

[13] Gyaourova, A., C. Kamath, and S.-C. Cheung, "Block matching for object tracking", LLNL Technical report, October 2003. UCRL-TR-200271.

[14] 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.

[15] H. Vincent Poor, An Introduction to Signal Detection and Estimation, Springer Press, 2nd Edition, *pp. 11,* 1994

[16] NIST/SEMATECH e-Handbook of Statistical Methods. Available : http://www.itl.nist.gov/div898/handbook/

[17] Emanuel Parzen, “On Estimation of a Probability Density Function and Mode”, *The Annals Mathematical Statistics, Vol. 33, No. 3, pp 1065-1076*, September 1962

[18] Robert V. Hogg and Allen T. Craig, Introduction to mathematical statistics, Macmillan Company, 2nd Edition, *pp. 285*, 1965.

[19] Martin Roesch, “Snort-Lightweight Intrusion Detection for Networks”, in *Proc. of USENIX 13th Systems Administration Conference (LISA) 1999*, Seattle, Washington, USA, November 1999.

[20] Snort. Available : http://www.snort.org/

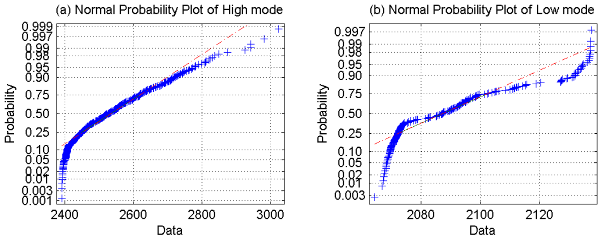
[21] Analysis Console for Intrusion Databases (ACID). Available : http://www.cert.org/kb/acid

[22] Burton Bloom, “Space/time trade-offs in hash coding with allowable errors”, *Communications of ACM, 13(7), pp 422-426*, July 1970.

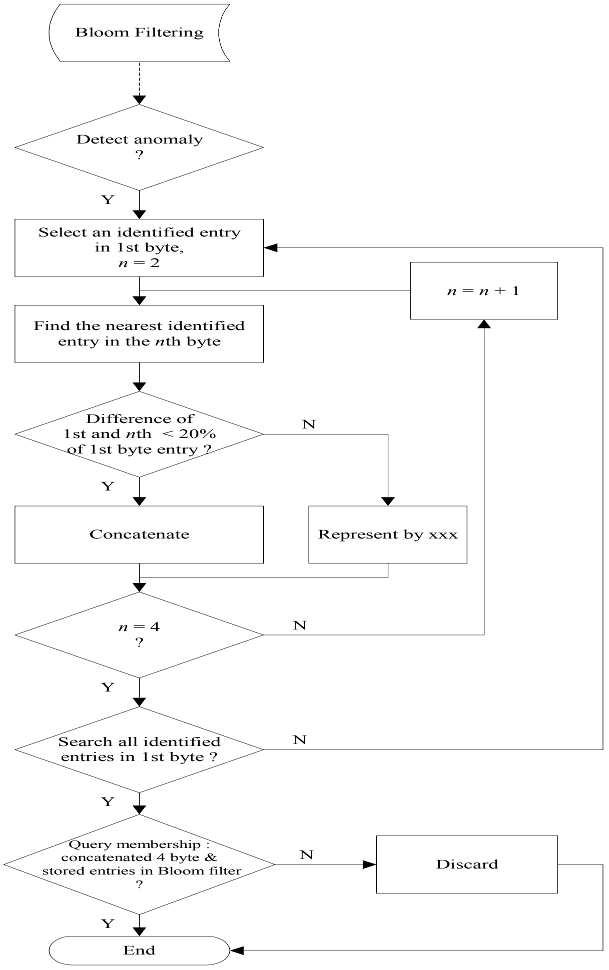
[23] Hyogon Kim, Inhye Kang, and Saewoong Bahk, “Real-time Visualization of Network Attacks on High-speed Link”, *IEEE Network Magazine*, Sept.-Oct. 2004.

APPENDIX

A-1. Normality of two modes in destination bimodal.



A-2. The Flowchart in concatenation of identification


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

A-3. Optimal test

$H_0 : X \sim N(\mu_N, \sigma_N^2)$

$H_1 : X \sim N(\mu_N + \mu_s, \sigma_N^2 + \sigma_S^2)$

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

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

by taking ln

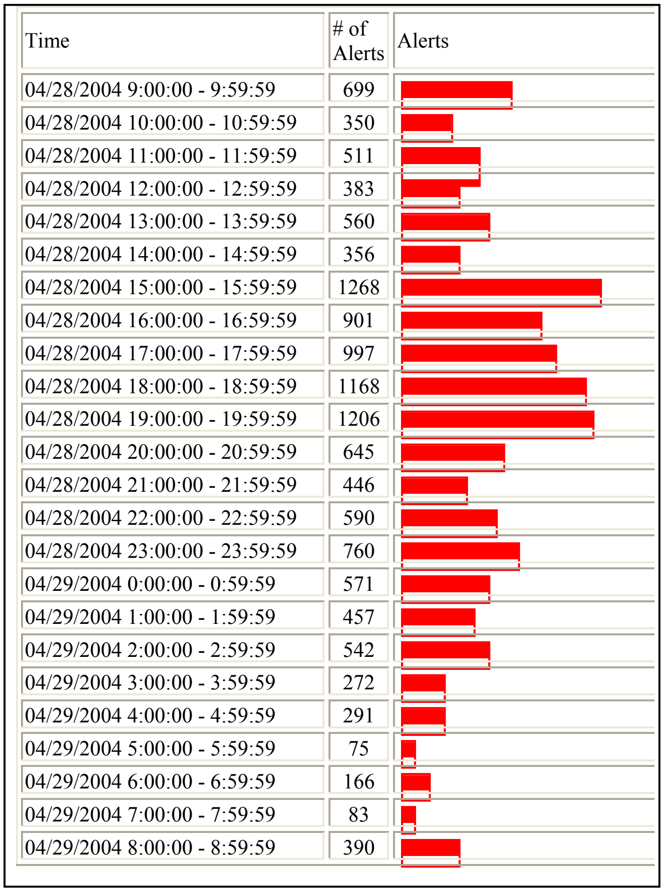
$$-\frac{(x-(\mu_N+\mu_S))^2}{2(\sigma_N^2+\sigma_S^2)} + \frac{(x-\mu_N)^2}{2\sigma_N^2} \gtrless \ln\frac{\sqrt{\sigma_N^2+\sigma_S^2}}{\sigma_N} + \ln\eta$$

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

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

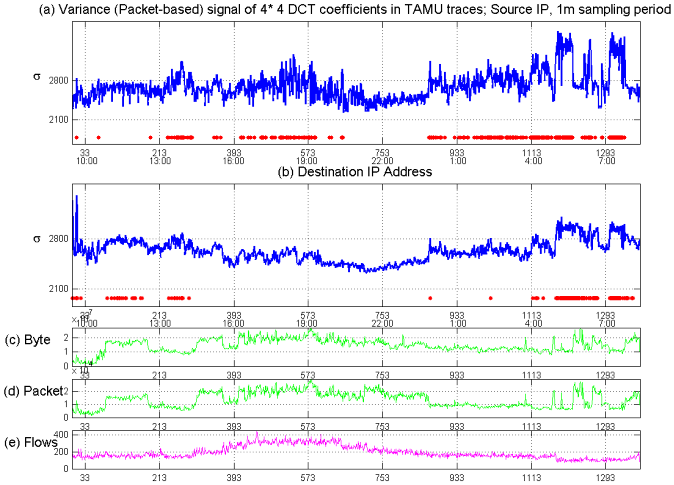
$$\left[x + \left(\mu_S R_\sigma^2 - \mu_N\right)\right]^2 \overset{H_1}{\underset{H_0}{\gtrless}} \tilde{\eta}$$

A-4. Snort report during 24 hours, Apr 28 & 29,2004



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	

A-5. NetViewer detection for Apr 28 & 29, 2004


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

(b) Destination IP Address