# Mutual Information Applied to Anomaly Detection

Yuliya Kopylova, Duncan A. Buell, Chin-Tser Huang, and Jeff Janies

*Abstract:* **Anomaly detection systems play a significant role in protection mechanism against attacks launched on a network. The greatest challenge in designing systems detecting anomalous exploits is defining what to measure. Effective yet simple, Shannon entropy metrics have been successfully used to detect specific types of malicious traffic in a number of commercially available IDS's. We believe that Renyi entropy measures can also adequately describe the characteristics of a network as a whole as well as detect abnormal traces in the observed traffic. In addition, Renyi entropy metrics might boost sensitivity of the methods when disambiguating certain anomalous patterns. In this paper we describe our efforts to understand how Renyi mutual information can be applied to anomaly detection as an offline computation. An initial analysis has been performed to determine how well fast spreading worms (Slammer, Code Red, and Welchia) can be detected using our technique. We use both synthetic and real data audits to illustrate the potentials of our method and provide a tentative explanation of the results.**

*Index Terms:* **Fast spreading worms, network anomaly detection, Renyi mutual information.**

## I. INTRODUCTION

Network monitoring systems have become a vital component in security infrastructures because they allow the detection of policy violations ranging from attempts to gain unauthorized access from the outside to wormlike activities emanating from within the network. Proliferation of computers and internet services combined with the rising complexity of modern networks lead to more elaborate attack patterns. Detecting such patterns is a necessary step in taking proactive corrective actions. Most intrusion monitoring systems, as shown in surveys [1], [3], [8], employ statistical techniques to establish profiles of "normal" system activities in order to identify anomalous behavior through observing significant deviations from the normal system activities. The major challenge of this approach is in defining parameters that accurately describe "normal" behavior without imposing constraints hindering adaptivity. That is, user behavior can change too rapidly for the learning system to adapt to this new behavior, and such deviations could result in false positive alarms. Another widely accepted alternative relies on data mining for signature-based anomalous patterns to single out known threats. The major handicap of this approach is inability to detect previously unseen attacks with different signatures. Worse yet, as discussed in [1], the proprietary nature of the signatures for most commercial ID systems makes a detailed discussion of their accuracy and adequacy difficult. Aside from the quality of the signatures, the timeliness of response in releasing detection signatures when new attack mechanisms have been identified is a major limitation of signature-based intrusion detection systems since there is a conflict in addressing the two goals of timeliness and quality.

It has been suggested in [6] and [11] that an information theoretic approach has a very strong potential for determining the state of a network as well as capturing the dynamics of the network traffic. Specifically, a set of metrics based on generalized Rényi entropies reflects the patterns of average topological order and disorder of a network. An abnormal variance of these entropy functions is hypothesized to reflect anomalous topological structural changes, which are indicative of a possible attack, of malicious activities, or of system failure.

The main objective of this paper is to demonstrate the feasibility of using Rényi entropy functions in the context of an intrusion detection environment and to validate the working hypothesis stated above. In this paper, we will briefly summarize what we consider to be the key events in understanding how Rényi mutual information can be utilized to signal that a system under observation is undergoing a significant change. We will also address specifics of the data simulation process and the expected tendencies based on the underlying theory. Following that, we will discuss possible interpretations of the experimental results based on analysis of traffic due to three fast spreading worms (Slammer, Code Red and Welchia). Finally, we describe what we think is a realistic scenario for future work.

## II. THEORETICAL BACKGROUND

In this section, we will discuss a few information-theoretic metrics used in our experiments and explain how they characterize regularities (that is, patterns) in audit and simulated network traffic data.

Generalized Shannon-Rényi entropy measures the uncertainty, variability or complexity of a collection of data items.

*Definition 1:* For a system $X$ with a finite set of $M$ possible states $\{x_1, x_2, \ldots, x_M\}$ the Shannon entropy of $X$ is defined as

$$H(X) = -\sum_{i=1}^{x_M} P(x_i) \log P(x_i) \tag{1}$$

where $P(x_i)$ is the probability that the system $X$ is in state $x_i$.

Shannon entropy is typically interpreted as the average information content of the data source, that is, it allows one to determine the expected capacity of a channel required to transmit the encoding of how evenly the states of $X$ are distributed. The entropy value is smaller when the data distribution is skewed, that is, when the data exhibits a pattern. The entropy value is larger when the data distribution is more symmetric, that is, when the data exhibits higher randomness [11].

*Definition 2:* For a system $X$ with a finite set of $M$ possible states $\{x_1, x_2, \cdots, x_M\}$ and system $Y$ with a finite set of $N$ pos-

Manuscript received March 8, 2006; approved for publication by James Won-Ki Hong, Division III Editor, July 20, 2007.

The authors are with the University of South Carolina, USA, email: {kopylova, buell, huangct, janies}@cse.sc.edu.

1229-2370/08/$10.00 © 2008 KICS

sible states $\{y_1, y_2, \cdots, y_N\}$, the conditional entropy of $X$ given $Y$ is in state $y_j$ is defined as

$$H(X|Y = y_j) = -\sum_{i=1}^{M} P(x_i, Y = y_j) \times \log P(x_i|Y = y_j). \qquad (2)$$

The conditional entropy of $X$ given $Y$ is the average over $Y$ of the conditional entropy of $X$ given $y$

$$H(X|Y) = -\sum_{i=1,j=1}^{M,N} P(x_i, y_j) \log P(x_i|y_j). \qquad (3)$$

The conditional entropy measures how much entropy (or uncertainty) is eliminated in $X$ by observing $Y$. In other words, if the combined system can be described with $H(X, Y)$ bits of information, and we learn how many bits are required to encode $X$, then upon the removal of $Y$, encoding of the resulting system will require the remaining $H(X|Y)$ bits of information.

*Definition 3:* For a system $X$ with a finite set of $M$ possible states $\{x_1, x_2, \cdots, x_M\}$ the generalized Rényi entropy of order $q$, for $q > 0$, is defined as

$$H_q(X) = \frac{1}{1-q} \log \sum_{i=1}^{M} P(x_i)^q \qquad (4)$$

where $P(x_i)$ is the probability that $X$ is in state $x_i$.

The Rényi entropies constitute a more general family of information measures than the Shannon entropy, which is $\lim_{q \to 1} H_q(X) = H(X)$. For the Shannon entropy measure, data items with high or low probability do not contribute much to the resultant value. With the generalized Rényi entropy for $q > 1$, higher probability events contribute more to the value than lower probability events do. The larger the $q$, the greater the weighting in the sum from probabilities having higher values.

*Definition 4:* For a system $X$ with a finite set of $M$ possible states $\{x_1, x_2, \ldots, x_M\}$ and system $Y$ with a finite set of $N$ possible states $\{y_1, y_2, \ldots, y_N\}$, the mutual information of $X$ relative to $Y$ is defined as

$$I(X;Y) = \sum_{i=1,j=1}^{M,N} P(x_i, y_j) \log \frac{P(x_i, y_j)}{P(x_i)P(y_j)}. \qquad (5)$$

Mutual Information represents the average amount of information about $X$ that can be gained by observing $Y$; that is, it measures the amount of reduction of uncertainty in $X$ after $Y$ is known. There is a relationship between entropy, conditional entropy, and mutual information: $I(X;Y) = H(X) - H(X|Y)$. Since $H(X|Y)$ represents the amount of information shared between $X$ and $Y$, $I(X;Y)$ corresponds to the intersection of the information in $X$ with the information in $Y$.

## III. PROPOSED MODEL

This section outlines how we apply elements of information theory to data processing in an attempt to capture network traffic abnormalities and details the process of obtaining parameters we use to make inferences. This model was suggested by Gudkov et al. in [6], [7] and our discussions outside of the scope of this paper related to a separate joint project.

A network in this model is viewed as a set of $n$ nodes, some pairs of which are connected with a representative non-negative weight indicating the strength of connection. A state of the network at a given time slice $t_i$ can be represented by a connectivity matrix $C_{ij}$ whose values are $m$ if a connection is established (initiated from node $i$ with node $j$ as destination) and 0 otherwise. The value of $m$ indicates the strength of the connection, either in number of connections made or number of bytes transferred in a given time slice.

To apply probability concepts, the entropy calculations start by normalizing the connectivity matrices $C$ obtained by processing the transaction log data:

$$\sum_{i,j} C_{ij} = 1. \qquad (6)$$

The total probability of the established connections initiated from the node $i$ is

$$P_i = \sum_j C_{ij}. \qquad (7)$$

The corresponding entropy characterizing uncertainty of the row connections, namely, the uncertainty that a given node in the network has established active connections originated *from* that node in a given time slice

$$H(row) = -\sum_j P_i \log P_i. \qquad (8)$$

Similarly, uncertainty that some connections have been established *to* a given node is characterized by entropy

$$H(col) = -\sum_i P_j \log P_j. \qquad (9)$$

The mutual information obtained from the connectivity matrix describes the state of the network at a given time slice as

$$\begin{aligned} I(C) &= H(row) + H(col) - H(col|row) \\ &= -\sum_{i,j} C_{ij} \log \frac{C_{ij}}{P_j P_j}. \end{aligned} \qquad (10)$$

Finally, in the general case, the mutual information is computed using the generalized Rényi entropy as

$$\begin{aligned} MI &= H_q(row) + H_q(col) - H_q(col|row) \\ &= \frac{1}{1-q} \frac{\log(\sum_j P_j^q) \log(\sum_i P_i^q)}{\log(\sum_{i,j} C_{ij}^q)}. \end{aligned} \qquad (11)$$

In this paper, we describe our efforts to understand how mutual information based on generalized Rényi entropy can be applied to anomaly detection as an offline computation.

There are three major differences separating our method from recent emerging information-theory-based anomaly detection methods:

- We take into consideration row and column entropies; this allows for greater flexibility in determining patterns (signatures). In conventional methods the overall entropy is calculated describing the aggregate configuration of the network (that is, seeking transformation of one time frame into another in its entirety, versus seeking transformation of row configurations into column configurations within the same frame and from frame to frame in our method). In other words, we believe it is possible to detect the presence of anomalous behavior (using frame to frame dynamics), as well as to identify the type of anomalous behavior (within frame patterns).
- We use Rényi information, which appears to be more sensitive than Shannon information due to its asymmetry. Unlike the Shannon measure, the average logarithm is replaced by the average of powers. This change results in the shift of relative contributions of an individual event to the total sum. We believe this property is highly desirable when disambiguating certain types of patterns (for example, scenario 1: Many connections from the host A to host B within the same time frame; versus scenario 2: Numerous unique connections from host A to a range of hosts).
- We use a diluting technique to alleviate deterioration in the sensitivity of the method due to repeated occurrence of patterns of the same type. This technique addresses the subtlety, associated with the severity of infection (the greater the number of infected hosts, the less profound the patterns become). The idea is to introduce a "reference" point into each sliding window by diluting the data in the current window with the data from a window characterized by low entropy.

## IV. THE EXPERIMENTS

In this section we describe results obtained from experiments and their relation regarding the theoretical definitions discussed earlier. From here on, the data refers to pre-processed timestamped audit records consisting of a set of features (IP address, origination, and destination ports, etc.)

### A. Data

The data of interest is a complete set of logical connections (traffic) in the network under observation made within a specified time. There are two main approaches to obtaining such data: Generating synthetic data and obtaining the data from an operational network using a packet capturing program.

We focused our research efforts exclusively on fast spreading worms. Their rapid propagation leaves no chance for a prompt human-mediated response, thus making them an ideal case study for the feasibility analysis. However, we can certainly hypothesize how our methods would behave if applied to the slow spreading worms:

- Aggressive, but slower spreading worms will still exhibit exponential growth pattern. Thus, adjusting (increasing) the window size discussed in the subsequent Section IV-B should suffice to catch up with the worm's speed. This assumption should hold for the slower spreading worms that spike outbound communication patterns and/or affect the volume of outbound connections.
- Truly stealthy worms evoke no communication patterns by hiding their scanning activities in normal traffic. It is highly doubtful our technique would be useful for these types of worms.

Our first case study (the Slammer worm) is based on synthetic data. To generate representative Slammer traffic, we superimposed the generated Slammer worm traffic onto the baseline background traffic to obtain a 5GB log file of raw data resembling a TCP header dump. Our second case study (the Code Red worm) is based on the real data collected from a sniffer located on the network segment behind the Internet router and in front of the firewall. The data is recorded before and during the Code Red worm outbreak (spanning 48 hours) and is filtered to reflect all attempts from external hosts requesting connections to port 80 on random IP addresses in class B network behind the firewall. Our third case study (the Welchia worm) is based on real data collected directly from the router of an attacked network. It corresponds to the audit file reflecting all activities over a 24-hour period on the day of a confirmed Welchia worm outbreak.

We point out that there are problems in using either real or synthetic data. With synthetic data, we run the risk of examining phenomena that are not sufficiently "real". With real data, the phenomena are real, but we cannot control the characteristics exhibited in the data. We use both types of data in our experiments in order to compensate the limitation of both types.

#### A.1 Slammer

Due to its extreme speed of infection, Slammer provides an adequate lower bound for sensitivity analysis of the proposed method. Regardless of normal traffic patterns, the presence of this worm in a network increases the traffic load to maximum capacity and quickly infects all susceptible nodes. Therefore, the worm presents itself as an anomaly with the characteristics of extreme bandwidth consumption to the point of inflicting a Denial of Service (DoS) attack on the infected network.

When infected with the Slammer worm, machines transmit 376-byte UDP packets to random IP addresses to propagate the infection. Since UDP protocol does not require acknowledgement, there is no penalty incurred by the scanning host if a randomly selected IP address does not exist. With the above medium, an infection propagates in less than 1 millisecond in modern sub-networks. The total propagation time is calculated by adding together the transmission delay and propagation delay of a 376-byte UDP packet in a 100 Mbps network. The transmission delay is usually defined as the ratio of the total number of bits in a packet to the rate of the network. The propagation delay is considered negligible and does not factor into determination of infection rate [10].

In simulating the Slammer infection, several assumptions are made about the nature of the network and the behavior of the worm:

- *The window of recorded traffic is at most 30 minutes:* By limiting the time of the infection, two advantages are gained. First, many variants in network activity can be limited. For instance, network workloads vary greatly through the course of a given day or week, but traffic for a finite amount of time can be considered constant, if the window of time is small. Second, the size of the generated log files is far less than logs of greater durations of time. Both aid in analysis for accuracy and storage requirements.

- *The probability of infecting is constant:* The probability that an outside infection hits a random IP address that is active in the sub-network being monitored is considered to be constant and independent of time. Though this is not entirely accurate, it is arguable that in using short windows of data collection, the rates do not vary drastically enough to affect the results negatively.
- *A certain amount of network traffic is not malicious:* Though the infectious traffic will eventually dominate the networks traffic load, it is assumed that a certain percentage of the traffic recorded is not malicious.
- *The network size is greater than 100 nodes and all network traffic is visible to the IDS:* If all routers and switches report relayed packets to the IDS, then it is conceivable that the IDS has a global view of the network.

The simulation package is divided into three components: background generation, worm generation, and merging. The background generator has little variance in operation; it produces a log of transactions according to a predetermined parameter list. These parameters include density of the network, the number of nodes in the network, the percentage of communications from this network to the outside world, and the duration of the traffic. The log produced by this generator is random non-malicious activity native to the network's normal operations.

The worm traffic generator evaluates the network according to current time slices (1 millisecond for the purposes of this experiment) and user specified parameters. At a given time slice all infected computers or hosts are evaluated according to their state and user specified parameters and appropriate actions are taken. Evaluation of this nature provides for the possibility of several hosts propagating the infection in the same time slice, a characteristic that is entirely possible in large networks. The additional parameters specify infection rate, miss penalties, and propagation delay. The probability of infection is indicative of the rate at which an external host accurately predicts an IP address within the network. The number of infections from external hosts in a given time slice is the result of a Poisson distribution with probability equal to the probability of infection. When an internal node is attempting to propagate an infection, the probability of success and the intended destination are determined by the result of a Poisson distribution. The choice of Poisson distribution is consistent with the nature of infections and is also made in related works such as [13]. There are three possibilities: the node can infect an internal host, infect an external host, or miss (select an IP address not in use). If the Poisson distribution indicates that the infection will succeed, a random number generator is used to generate an infecting IP address. If the IP address exists within the subnet and can be infected, the destination host becomes infected. Otherwise the IP address is assumed to be external. In both cases the host waits for the propagation delay value number of time steps. Otherwise, the host waits for the miss penalty value number of time steps.

By adjusting the infection rate, miss penalties, and propagation delay values, the lower bound of sensitivity can adequately be assessed. At present the infection rate is set to .005; with the current random number generator, the first malicious packet appears after 53 milliseconds. Due to the nature of the medium used to transfer the packet, the miss penalty is set to 0 milliseconds, and the propagation delay is set to 1 milliseconds. Though the miss penalty and propagation delay are assumed to be constant, we choose to make these values adjustable for future testing.

Finally, the merger is used to combine the logs generated by the background generator and the worm generator.

### A.2 Code Red

The first incarnation of the Code Red worm was launched on July 13th, 2001. It exploited a common vulnerability of the Microsoft IIS web servers known as buffer overflow. Once a host gets infected, the worm spreads by initializing 99 threads generating random IP addresses, and then probing those addresses to see if it can connect to tcp/80 and compromise the hosts using the same vulnerability. The bug in random subnet generation routine in the original version of the worm (static seed) was later fixed to prevent re-infecting exactly the same systems multiple times.

The data set we obtained from a second party contains filtered traffic collected on the network segment behind the Internet router and in front of the firewall protecting a class B network. The filter was set to reflect the flow of the traffic destined to port 80 and rejected by the firewall since no designated web servers resided outside of DMZ. The log consists of initial SYN packets without acknowledgements before and on the day of the attack (July 18th through 19th, corresponding to the Code Red v2 worm outbreak).

### A.3 Welchia

Similar to the Slammer worm, the Welchia worm spread throughout the world in a matter of hours. However, unlike Slammer, this worm uses TCP connections to propagate, which require a three way handshake for establishing a connection between two hosts. Since TCP is a connection-oriented protocol and the packet size is larger, the spread of this worm is slower, than that of Slammer. Furthermore, a miss in scanning results in a TCP timeout delay in which the initiating host must wait until the expiration of an interval of time before determining that the destination address is not in use.

The data set provided to us by the second party is approximately 3600 seconds in duration. However, the only data that is recorded is from packets entering and leaving the network. Therefore, intra-network traffic is lost. This may result in a skewed or limited view of the network as a whole but accurate to real world deployment strategies for an IDS.

## *B. Data Processing with respect to Sliding Window*

In reading the data, the transaction log is partitioned into sliding windows with overlap. From each window a connectivity matrix is derived representing the state of the network at time frame $t_i$. Once the whole transaction log file is processed, a

| Data description | Slammer | Welchia | Code Red |
|---|---|---|---|
| Type of traffic | Simulated | Real (all) | Real (rejected) |
| Simulation Parameters | Infection Rate 0.005<br>Miss Penalty 0 ms<br>Propag. Delay 1 ms | N/A | N/A |
| Duration of log file | 30 min | 24 h | 24 h (of 48 h) |
| Sampling rate | 1 ms | 5 sec | 5 sec |
| Window size | 1 ms | 1 min | 30 sec |

Fig. 1. Summary of parameters and data description for the three experiments

unique index is assigned to each network node participating in transactions. The index representing a node is static and does not change from window to window; this allows us to pinpoint a particular node and examine its activity in relation to the system as a whole. Given a connectivity matrix, the entropy measures are calculated using the procedures outlined in Section III.

One of the important tunable parameters is the size of the sliding window; this was empirically determined and greatly influenced by the density of connections observed per window and the sampling rate of the data. The premise is quite simple: we can detect common behavior from window to window if their content is identical or structurally similar, that is if the window size is small enough, common behavior can be isolated despite the variation in the overall structure of the connectivity matrix. A great source discussing theoretical foundation of determining the appropriate sliding window size to analyze sequences of events in order to discover recurrent episodes can be found in [14]. However, it emphasizes the following point: "... to be considered interesting, the events of an episode must occur close enough in time. The user defines how close is close enough by giving the width of the time window." In other words, determining if a recurrent event falls into the sliding window can be computationally confirmed, whereas the exact window size is determined through a tedious and subjective process of trial and errors. For the Slammer worm, the size of the sliding window was set to 1 millisecond, whereas it was set to 1 minute for the Welchia worm traffic, reflecting a much slower propagation rate. For the Code Red worm, the sliding window was set to 30 seconds to examine the overall dynamics; we used a smaller window size (5 seconds) to zoom into the incipient phase. Too large of a window results in denser traffic that leads to deterioration of the resolution of the proposed metrics. As the sampling period increases, the data becomes more homogeneous because more data both from the past and the future are included, making mutual information look more deterministic (it increases in value and smoothes out). Too small of a window, on the other hand, make random uncorrelated patterns stand out.

## *C. Expected Tendencies*

### C.1 Shannon Entropy

In the context of anomaly detection, entropy can be used in a similar way to measure the regularity of audit data. Assigning a set of attributes (IP address, origination port, and destination port) to an entry in a data log is sufficient to "label" a transaction. It is clear that the same labels will be assigned to a number of records that differ in their timestamps but are otherwise indistinguishable. Entropy grows when distinctions lose their meaning and the system spreads into every possible configuration. Entropy is reduced when only one of many possibilities is prevalent. That is, the smaller the entropy, the fewer the number of different records (the higher the redundancy), indicative of more regularity (the presence of patterns) in the audit log. In other words, a few hyperactive nodes spanning many connections within a short period of time or hitting the same ports over and over again will cause the overall entropy value to drop.

### C.2 Rényi Entropy

Generalized Rényi entropy can be applied to anomaly detection under the same scenarios for which Shannon entropy is believed to be useful. However, the Rényi entropy might be more sensitive when disambiguating certain types of patterns. Because the relative contribution of events to the total sum is changed to favor events with higher probability, it should be easier to pick up repeated transactions in the audit data since those transactions receive larger weights.

### C.3 Mutual Information

Mutual information, interpreted as a measure of uncertainty, can be thought of as the reduction in the uncertainty of $X$ (rows) due to the knowledge of $Y$ (columns). Applied to the context of our model, we can consider entries of a connectivity matrix as outcomes of a random variable when operating within the same time-slice (same connectivity matrix). Joint entropy and mutual information characterizing two different time slices will change with every geometrical alignment of entries in the connectivity matrix. Rows and columns with common values are regarded as small information sources, whereas uncommon-valued entries are appraised as more information. This suggests that the mutual information should be sensitive to certain symmetries in the connectivity matrix. For example, the simultaneous presence of a defined row and column in the connectivity matrix will result in decrease of the mutual information parameter.

## *D. Experimental Results*

### D.1 Slammer Worm

Slammer was one of the fastest computer worms in history. Activity of this worm is readily identifiable on a network by the presence of 376-byte UDP packets. These packets will appear to be originating from seemingly random IP addresses and destined for port 1434/udp.

The synthetic log file we examined in our first case study contained nine instances of Slammer infection. When analyzing the mutual information plot, Slammer's particular behavior (scanning without acknowledgments) corresponds to a change in levels of the mutual information (MI). There are a few plateaus on the plot, each corresponding to introduction of a new infected host.

As shown in Fig. 2, the very first anomaly is clearly identifiable as a sudden dive at window 1800. The smaller MI value suggests that a new pattern has been detected. Moreover, this new pattern is drastically different from what have been seen before. The second instance of infection is at window 2050,

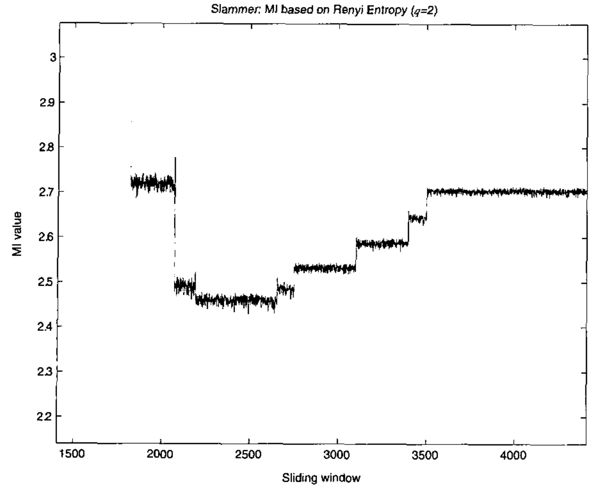


Fig. 2. Slammer: MI Plot for the incipient stages of infection. The plateaus on the plot correspond to introduction of new infected hosts. The initial drop in the MI value can be ascribed to within-frame dynamics associated with particular nodes' alignments, whereas the rise following it can be explained by the frame-to-frame dynamics reflecting a persisting pattern.

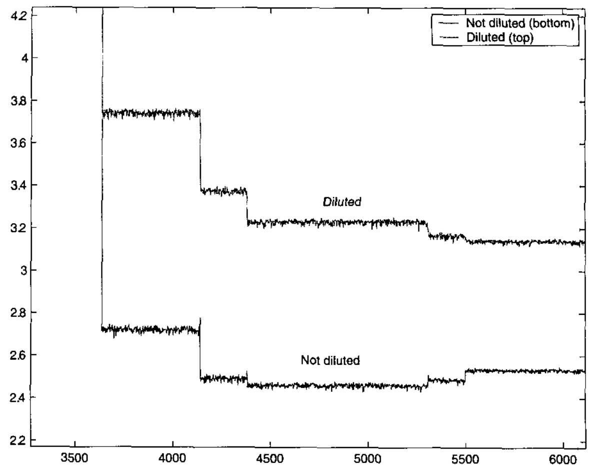


Fig. 3. Slammer: Effect of diluting.

when MI decreases, thus indicating the presence of another pattern. However, the value change in this case is less pronounced because both patterns are similar.

The initial drop in the MI value can be ascribed to within-frame dynamics associated with particular nodes' alignments, whereas the rise following it can be explained by the frame-to-frame dynamics. The further infection progresses, the more information is embedded in the previous time frame. In other words, with more and more nodes involved in abnormal activities, the MI value starts changing in the opposite direction. This is because what was earlier perceived as an extraordinary event (that is, in disagreement with what have been seen) is now perceived as a part of a consistent trend. In other words, introducing new abnormalities *of the same type* loses its "shock" value as the overall pattern becomes apparent.

### D.2 Diluting Technique

This subtlety, associated with the severity of infection (the greater the number of infected hosts, the less profound patterns become), is explained by the concave shape of the entropy function. There exists some minimum point, beyond which the mutual information will increase in value either due to introduction of a new pattern of the same type of abnormal activity (as the abnormal activity now becomes the majority and hence entropy decreases) or due to removal of an existing pattern (which signifies the trend moving toward normal status). Therefore, at such a point it is difficult to disambiguate the behavior of the mutual information (increase or decline of abnormal activities) without taking into consideration the history of the previous trends.

To address this subtlety, we propose to use a diluting technique. The idea of diluting is to introduce a "reference" point into each sliding window by diluting the data in the current window with the data from a window corresponding to a baseline in which there is no abnormal traffic. In this case, although the introduction of a new pattern of the same type adds to the overall proportion of the abnormal activity within the window, the abnormal activity remains in the minority because the calculation is diluted by the baseline traffic. Therefore, the aforementioned ambiguity is removed, because introduction of a new pattern will increase the mutual information whereas removal of an existing pattern will decrease the mutual information. Although a window corresponding to the baseline traffic is not readily available, the diluting can still be partially but effectively achieved by sampling both a previous window and the current window; in our experiments, we took every second record in both windows. The effect of applying the diluting technique is shown in Fig. 3, where it is evident in the shift of the point of concavity. When the diluting technique is not applied, the MI curve (the bottom curve) starts climbing after reaching a minimum point and results in an ambiguity. We disambiguate by applying the diluting technique that results in the top curve of the graph, and we observe that the MI curve continues to decrease. Thus we can deduce that the ambiguity is due to the introduction of a new pattern of the same type, rather than the removal of an existing pattern. In summary, by using both the diluted and the original data we can make inferences on the presence of anomalies and their severity and tell if the increase of the MI value can be ascribed to the elevation of abnormal activities or to their decline.

It can be argued that in realistic scenarios it is extremely hard to hypothesize what would constitute the "baseline" behavior (time of the day (load), presence of a server(s), specific connections characteristic of this particular network, etc). However, assuming the overall model is valid, low entropy values (but not necessarily high MI values) must capture the essence of the normal behavior. In other words, the reference point can be done not with respect to the hypothetical ordinary traffic, but with respect to the portions of the traffic characterized by low entropy (i.e., likely to be "ordinary" mathematically speaking).

Another way to dilute traffic is by using an adaptive reference frame that is equivalent to remembering the direction of the last change. This approach allows to redefine the baseline behavior as the system evolves and reduce the "fatigue" from the patterns of the same type by continuously shifting the concave point.

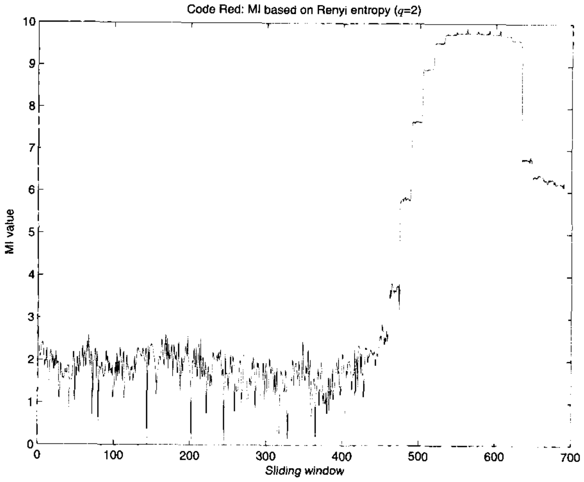


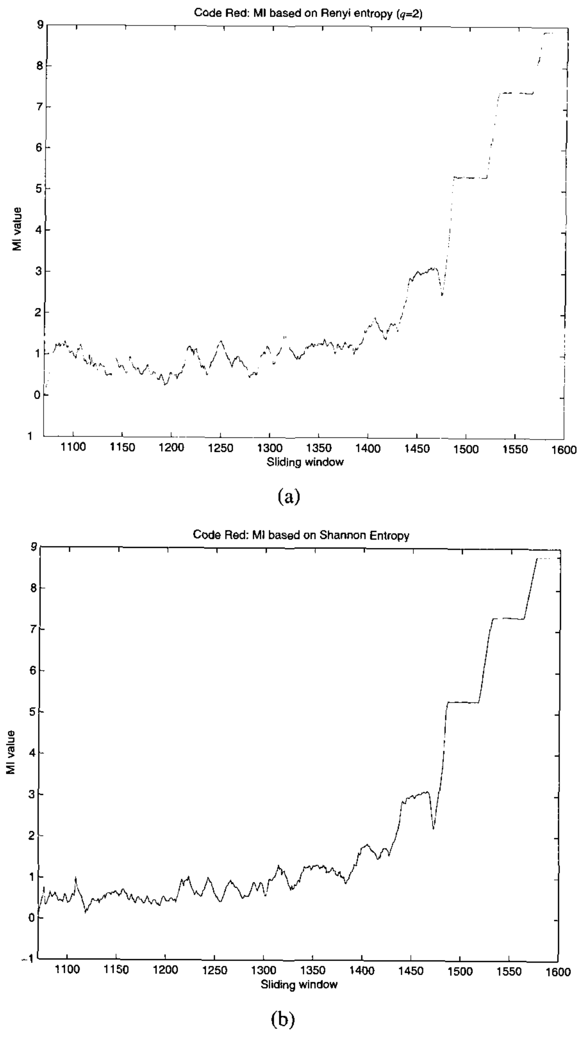
Fig. 4. Code red: MI Plot. Three phases are observable: Random probing activities (up to window 400) followed by acute worm outbreak (manifesting itself in high MI values) concluded by decline of activities (MI begins to subside).

### D.3 Code Red

Code Red (v2) was another virulent worm that wrought havoc and infected more than 350,000 computers overnight. The propagation of this worm is identifiable by the presence of packets destined to a multitude of hosts on port tcp/80. Our second case study was based on real traffic captured by a sniffer placed at the entry point of a class B network protected by a firewall. The data we examined consist of initial SYN packets to port 80 rejected by the firewall over the period of 48 hours.

It is not the case that all of the packets were generated by the worm; some of them were contributed by probing and hacking activities. Nonetheless, the patterns detected are undoubtedly due to the worm propagation once the number of connections starts to skyrocket. The outbreak of Code Red is evident through observing changes in MI values shown in Fig. 4. The starting point of infection obtained from the original log file translates to window 400 on the plot. This means that patterns prior to window 400 describe the background noise consisting of random probing activities. Once anomalous activities establish themselves as persistent patterns, the MI value starts to climb. Once again, the further the infection progresses, the more information is embedded in the previous time frame, which, in terms of frame-to-frame dynamics, is characterized by higher MI values. The last portion of the plot indicates decline in activities after containment measures were put in place.

By window 500 the worm had been spreading for five hours. It would be interesting to investigate, whether there is some within-frame dynamics that could suggest anomalous activities before it is too late. To "zoom in", we adjusted (lowered) the sliding window size to further investigate the portion of the traffic corresponding to windows 400–500 in Fig. 4, hoping to bring to focus some tendencies lost when examined on a bigger scale. Fig.5 compares Shannon and Rényi MI behavior immediately after the first signs of infection.



(a)

(b)

Fig. 5. Code Red: Comparison of Rényi (left) vs. Shannon (right) MI plots immediately after infection.

Similar to the Slammer experiment (but not quite as pronounced), we observed a decline in Rényi MI values prior to their increase (windows 1,070–1,200 corresponding to 15 minutes of traffic in real time). Shannon MI, though very similar in general shape, stayed relatively flat for the duration of the windows of interest. We hypothesize that this behavior might be ascribed to the higher sensitivity of Rényi entropy to particular connectivity matrix configurations. In this case, the peculiarity of configuration can be thought of as distinct row patterns (without column patterns like in Welchia worm discussed below). These patterns exist when scanning is attempted by a single host to multiple destinations without acknowledgement. Thus, within-frame dynamics take prevalence when only a few geometric alignments are distinguishable (but not too few to make differentiation from random noise possible). It goes away when these alignments get lost due to the sheer volume of patterns, that is when frame-to-frame dynamics take over.

### D.4 Welchia

Our third case study was based on a real traffic audit file capturing the outbreak of a Welchia worm in real time. Activity of this worm is identifiable by the presence of packets on port tcp/135 broadcast to multiple IP addresses. Scans consistent with the Welchia behavior (initiation of a large /16 network

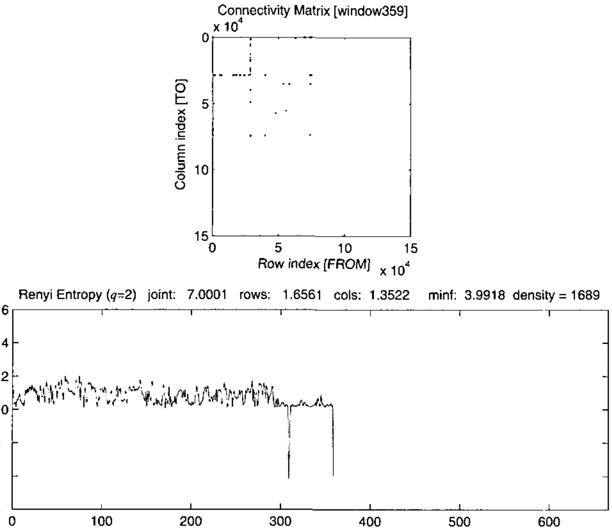


Fig. 6. Welchia Worm: Snapshot of MI. Well defined dives in MI value correspond to a broadcasting node after acknowledgments have been received. The first occurrence is at window 306 and the second occurrence is at window 359.

scan starting at A.B.0.0. and counting up to at least A.B.255.255 exhaustively followed by a complete halt of scanning activities) were first confirmed to exist in the log file.

These scans are clearly detectible from both connectivity plots and mutual information graphs and correlate with the evidence from the log file. As shown in Fig. 6, the patterns that are most characteristic of the worm's behavior manifest themselves in a significant drop of mutual information when rows and columns of the connectivity matrix form an unmistakable pattern reflecting scans followed by acknowledgments. Quite puzzling, we observed negative values for the MI in the figure presented above. We are not attempting to interpret the physical meaning of such values at this phase of our effort. In our opinion, the fact that patterns were clearly isolated is of great importance and cannot be ignored even though the values might be off. Moreover, we believe this matter can be resolved by more careful normalization without compromising the overall model.

## V. CONCLUSIONS AND FUTURE WORK

These investigations based on both synthetic and real data have indicated that mutual information based on generalized Rényi entropy can be quite useful for certain attack scenarios. Our observations suggest that anomalous traffic will be likely accompanied by a sudden change in mutual information compared to the baseline traffic. Under intrusion, the mutual information might increase or decrease, depending on the nature of the attack and the status of the traffic before the attack. Using our diluting technique, we can alleviate deterioration in sensitivity of the method due to repeated occurrence of patterns of the same type.

We acknowledge that our model is an over-simplification of real network traffic. We make simplifying assumptions both about the structure of the network and the nature of attacks. We felt these simplifications were justifiable for the initial investigation and shed the light on the intuition behind the effectiveness of the technique. In future work, we will conduct more comprehensive experiments, consider an alternative normalization process, validate the results statistically, and extend our studies to a broader class of attacks beyond fast spreading worms.

## REFERENCES

[1] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner, "State of the practice of intrusion detection technologies," *Technical Report CMU/SEI-99-TR-028, Software Engineering Institute*, Carnegie Mellon, USA, 2000.

[2] S. Axelsson, "A preliminary attempt to apply detection and estimation theory to intrusion detection," *Technical Report 00-4, Chalmers Univ. of Technology*, Goteborg, Sweden, 2000

[3] T. Brugger, "Data mining methods for network intrusion detection," Ph.D. Dissertation, University of California, Davis, USA, June 2004.

[4] D. A. Buell, "Calibrating entropy functions applied to computer networks," in *Proc. the Third International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security*, St. Petersburg, Russia, 2005.

[5] A. Golan and J. Perloff, "Comparison of maximum entropy and higher-order entropy estimator," *Journal of Econometrics*, vol. 107, no. 1, pp. 195–211, 2002.

[6] V. Gudkov and S. Nussinov, "Graph equivalence and characterization via a continuous evolution of a physical analog," *eprint arXiv:cond-mat/0209112*, 2002.

[7] V. Gudkov, S. Nussinov and Z. Nussinov, "A novel approach applied to the largest clique problem," *eprint arXiv:cond-mat/0209419*, 2002.

[8] G. Helmer, J. Wong, V. Honavar, and L. Miller, "Automated discovery of concise predictive rules for intrusion detection," *Technical Report 99-01, Iowa State Univ.*, Ames, USA, 2000.

[9] I. Kojadinovic, "On the use of mutual information in data analysis: An overview," in *Proc. Conference International Symposium on Applied Stochastic Models and Data Analysis*, Brest, France, May 2005.

[10] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 3rd Ed. PearsonEducation, Inc., 2005

[11] W. Lee, "A data mining framework for constructing features and models for intrusion detection systems, " Ph.D. Thesis, Columbia Univ., New York, USA, 1999.

[12] W. Lee and D. Xiang, "Information-theoretic measures for anomaly detection," in *Proc. 2001 IEEE Symp. Security and Privacy*, Oakland, CA, 2001, pp. 130–143.

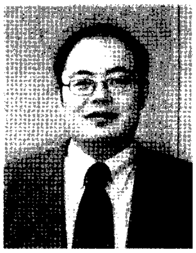
[13] M. Liljenstam, D. Nicol, V. Berk, and R. Gray, "Simulating realistic network worm traffic for worm warning system design and testing," in *Proc. 2003 ACM workshop on Rapid Malcode (WORM'03)*, Washington DC, USA, Oct. 2003.

[14] H. Manilla, H. Toivonen and A. Verkamo, "Discovery of frequent episodes in event sequences," *Data Mining and Knowledge Discovery.*, Netherlands, vol. 1, pp. 259–289, 1997.

[15] K. Zyczkowski, "Rényi Extrapolation of Shannon Entropy," *Open Syst. Inf. Dyn.*, Netherlands, vol. 10, pp. 297–310, 2003.

**Yuliya Kopylova** received the B.Sc. degree in Computer Science Engineering from University of Anchorage Alaska in 2000. Upon graduating from UAA, she worked as a Software Engineer developing real-time embedded systems and autonomous acoustic recording packages that required the use of signal processing, image processing, and pattern recognition techniques. She joined Univerity of South Carolina in 2004 to pursue Ph.D. in Computer Science Engineering. Her research interests include network security, VANET, anonymity, and group communication.

**Duncan A. Buell** received the Ph.D. degree in 1976 in mathematics from the University of Illinois at Chicago. He was an assistant and then associate professor in the Department of Computer Science at Louisiana State University in Baton Rouge. From 1986 to 2000 he worked at the Supercomputing Research Center (now the Center for Computing Sciences), a division of the Institute for Defense Analyses, doing research in high performance computing and computational mathematics. He has written two books and more than fifty research papers in number theory, document and information retrieval, parallel algorithms, and computer architecture. While at IDA he was project manager for the Splash 2 reconfigurable computing project, one of the first successful ventures into the use of FPGAs as a programmable "CPU" in what is now known as a reconfigurable a computing machine. He joined the University of South Carolina in October 2000 as Professor and Chair of the Department of Computer Science and Engineering, where he continues his research in parallel and reconfigurable computing.



**Chin-Tser Huang** received the B.S. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1993, and the M.S. and Ph.D. degrees in computer sciences from the University of Texas at Austin in 1998 and 2003, respectively. He joined the faculty at the University of South Carolina at Columbia in 2003 and is now an Assistant Professor in the Department of Computer Science and Engineering. His research interests include network security, network protocol design and verification, and distributed systems. He is the director of the Secure Protocol Implementation and Development (SPID) Laboratory at the University of South Carolina. He is the author (along with Mohamed Gouda) of the book "Hop Integrity in the Internet", published by Springer in 2005. He is a member of Sigma Xi, Upsilon Pi Epsilon, IEEE, and ACM.

**Jeff Janies** graduated from the University of South Carolina with a Master's in Computer Science in 2006. He is currently a member of the technical staff for CERT in the Network Situational Awareness group. His research interests include Anomaly-based IDS, ad-hoc sensor network security, and network traffic visualization.