Wavelet-based Real Time Detection of Network Traffic Anomalies

Chin-Tser Huang[1], Sachin Thareja[1], and Yong-June Shin[2]

(Corresponding author: Chin-Tser Huang)

Department of Computer Science & Engineering, University of South Carolina[1]

301 Main St., Columbia, SC29208, USA (Email: huangct@cse.sc.edu)

Department of Electrical Engineering, University of South Carolina, Columbia[2]

(Received Aug. 5, 2006; revised and accepted Oct. 6, 2006)

Abstract

Real time network monitoring for intrusions is offered by various host and network based intrusion detection systems. These systems largely use signature or pattern matching techniques at the core and thus are ineffective in detecting unknown anomalous activities. In this paper, we apply signal processing techniques in intrusion detection systems, and develop and implement a framework, called Waveman, for real time wavelet-based analysis of network traffic anomalies. Then, we use two metrics, namely percentage deviation and entropy, to evaluate the performance of various wavelet functions on detecting different types of anomalies like Denial of Service (DoS) attacks and portscans. Our evaluation results show that Coiflet and Paul wavelets perform better than other wavelets in detecting most anomalies considered in this work.

Keywords: Entropy, intrusion detection, network traffic anomaly, percentage deviation, wavelet

1 Introduction

Intrusion detection and network security is of increasing significance in today's world. An exponential rise in the use of networks (and the Internet) has subsequently led to various types of exploits and malicious activities, each having its own negative impact. Many organizations have faced attacks from unknown entities, with sometimes unknown motives. As the reliance on network resources increases, so does the need to secure them. There are various methods of monitoring for intrusions in current practice; our work is based on a network-based, signal processing approach. Network-based intrusion detection differs from host-based techniques in that a host-based technique analyzes activities on the local host machine, and is not concerned directly with the analysis of network traffic. Both network-based and host-based techniques involve sampling of available data, preprocessing, pattern matching and/or transform analysis, and policy based actions. The feature of pattern or signature matching approach is the identifying feature: based on known models, current data can be evaluated against them to flag alerts. However, in the case of an unknown anomaly, this approach is ineffective. On the other hand, the feature of the signal processing approach is monitoring the point of change, applying transforms to the data and flagging events based on thresholds, and it is the approach that we adopt in this work.

To investigate the effectiveness of signal processing techniques (wavelet specifically) applied on network traffic anomaly detection, we develop a framework called Waveman, which use an open source tool called LastWave [1] to provide a real time analysis of network traffic. We have developed and evaluated wavelet filters based on a heuristic approach. Four different families of wavelets, namely Coiflet, Morlet, Daubechies, and Paul, are used in this work. To evaluate the various wavelets considered in this paper, we have used passive as well as active methods. As defined in [17], a "passive" method of evaluation refers to a technique in which current traffic is monitored for anomalies, while an "active" method refers to a technique which injects traffic as stimulus to the network and studies the resulting effects. For passive evaluation we used the data obtained from a Virginia-based registrar of Internet domain names, EnetRegistry, Inc., and for active evaluation, the 1999 MIT Lincoln Laboratory Intrusion Detection System Evaluation data set was used. From these data sets, we have analyzed five different anomalies; three are Denial of Service (DoS) attacks from the active data set, and two are scanning anomalies taken from the passive data set. The DoS attacks also include a distributed attack (DDoS) that uses many compromised hosts to launch an attack against a single victim. The scanning anomalies are characteristic of black hat hacking and worm activity. While various Intrusion Detection Systems (IDS) use pattern matching [6, 29] and application-specific parameter correlating [19] techniques to identify and block such anomalies, in this work we use a change point monitoring approach [30] aiming to detect all types of anomalies, known or unknown.

There are two main contributions of this paper: to achieve a real time wavelet analysis of network traffic, and to evaluate different wavelets for their performance on identifying network anomalies like Denial of Service (DoS) attacks, floods and port scans. The first objective was achieved by the Waveman framework we developed. Then, after different wavelet filters were designed for use in Waveman, the second objective is achieved by using two metrics, namely percentage deviation and entropy, to evaluate these wavelet filters when they are applied on the two data sets containing a variety of anomalies.

It has been discovered that ethernet traffic exhibits self-similarity in nature [21], from which stems the reasoning that the aggregation or decomposition of a network traffic signal contains similar amount of “burstiness”. Thus, a few samples of such a signal would contain an equivalent variance of a longer signal of the same type. Wavelet-based techniques exploit this self-similarity property and analyze signals at various levels of decomposition, which is demonstrated to be effective in this work.

2 Background and Related Work

2.1 Signal Processing and the Fourier Transform

Mathematical transforms are applied to a signal to perform processing and obtain further details than what is available in the original signal. The Fourier transform is a traditional method for the spectral analysis. For many applications, it is necessary to obtain information regarding the spectral contents of a time-series signal, rather than its amplitude in time domain. The Fourier transform does provide frequency contents of the signal, but does not provide information regarding “where in time” those frequencies exist. Hence, if one is only interested in the spectral components of the signal then the Fourier transform is useful, but if one is interested in the time locality information, then the Fourier Transform is not enough. Therefore, for most real world applications, the FT is not suitable for non-stationary signals, including the signal obtained from sampling network traffic.

The Short Time Fourier Transform (STFT) performs time-localized analysis for non-stationary signals, but it suffers from the drawbacks due to the Uncertainty Principle [27]. This principle states that it is impossible to know the exact time-frequency representation of a signal, i.e. one cannot know what spectral components exist at what instances of times. What one can know is limited by the time intervals in which a certain band of frequencies exist, which is a time resolution problem. These resolution problems relate to the STFT in that, a narrow window offers good time resolution but poor frequency resolution, and a wide window results in good frequency resolution but poor time resolution.

The wavelet transform overcomes the resolution problems of the STFT by using a varying window size. The wavelet transform coefficients contain the results due to analysis from all window sizes. A discrete version of a Continuous Wavelet Transform (CWT) enables its computation by digital computers; however, it is not a true discrete transform. Such wavelet series is simply a sampled version of the CWT, and the information it provides is highly redundant as far as the reconstruction of the signal is concerned. This redundancy requires a large amount of computation time and resources. The Discrete Wavelet Transform (DWT), on the other hand, provides sufficient information both for analysis and synthesis of the original signal, with a significant reduction in the computation time. The discrete version of such a decomposition thus involves an iterative process; in each iteration, a signal of length x is the input, and two or more signals are derived from it. In the analysis, a specially designed filter F is used, which operates on x and eliminates the coefficients not related to the required output $F(x)$. A low pass filter L produces a low-frequency output $L(x)$. $H_1, H_2, \ldots, H_r$ produce a high frequency output $H_i(x)$ and can be thought of as discrete differentiation. Further iterations produce a further decomposed $L(x) : L^2(x), H_1L(x), \cdots, H_rL(x)$. Finally, a family of output signals of the form $H_iL^{j-i}(x)$ is obtained, where j is the number of low pass filtering iterations performed to obtain the final signal. A larger value of j corresponds to a lower frequency signal. The values of the derived signals $H_iL^{j-1}(x)$ are known as the ”wavelet coefficients”. In brief, a synthesis operation takes as input the signals obtained from the decomposition phase, and reproduce the original signal. We are mainly concerned with the decomposition of the raw signal for the purpose of this work.

A filter H_i has k vanishing moments if $\hat{H}(0) = \hat{H}'(0) = \ldots = \hat{H}^{(k-1)}(0) = 0$, where $\hat{H}$ is the Fourier series of H. An indication of the vanishing points affecting the analysis has been given in [4], so we do not address this issue in this paper.

2.2 Motivation and Related Work

In [4], Barford et al. implement a wavelet analysis of Cisco NetFlow [10] data. A three month long signal containing different kinds of anomalies was analyzed. It was found that an increase in the local variance of a time-series signal that is generated from raw traffic strongly indicated an anomaly. Wavelet analysis on a raw traffic signal allows for observation on many different levels of traffic, by removing certain components of the signal at each level, and generating wavelet coefficients. This feature extraction at different levels is known as Multi-Resolution Analysis (MRA), and has gained popularity as the method of analysis for non-stationary signals. The High and Mid frequency portions are normalized to have variance 1. The variability of these parts is computed using a moving window. A deviation algorithm can then identify anomalies based on thresholding the variable part of the signal generated from the wavelet coefficients at different frequency levels. All the analyses in the work were done with the same wavelet system, but justification of the choice of that wavelet was left open.

In [17], Huang et al. proposes a wavelet based method, called WIND, to detect network failures and other problems. The energy function of a signal with exponentially increasing arrival time of components shows that the values are roughly constant across all scales, which could refer to the white noise of a signal. However, the wavelet transform of a signal that is comprised of values taken at different sample points exhibits details hidden within the signal. The authors have suggested methods by which this system could be used to plot the RTT values of a network path. It has been shown that the self similar nature of network traffic [21] does not allow conventional methods of analysis to provide statistics about local periodicities in the signal.

In [18], the authors use wavelets to detect congestion on shared links in networks. The technique suggests that two paths sharing a congested link have a high correlation between their one way delays. This correlation has been monitored, and wavelet denoising is used to remove noise due to queuing delay and mild congestion. Thresholding is applied to arrive at a binary decision regarding a shared link being congested. The authors have found the Daubechies 6 wavelet has the most correlation with the congestion implementation and hence it was used as the mother wavelet for wavelet denoising in this work. The superior characteristics of wavelets for denoising have been demonstrated.

The motivation for this work is to justify the assumptions that wavelets can be used to develop a real time network intrusion detection system and that some wavelets perform better than the others when used in a real time network intrusion detection system. For this purpose, it is necessary to monitor traffic on the fly and use different wavelet filters to analyze the monitored traffic. Through experiments on traffic data sets we aim to identify the wavelet(s) that perform better in this regard. We believe that this knowledge could indeed be useful in developing such an intrusion detection system.

3 Implementation

3.1 Initial Testing and Tuning

We use LastWave 2.03 [1], an open source signal processing command language developed by Emmanuel Bacry. LastWave constitutes the core of our framework, and is used for all the analysis. Then we prepare sample flow files, based on Cisco IOS flow exports. These flows are exported from Cisco routers running cflowd, and contain per flow information such as source IP address, destination IP address, source port, destination port, packets, bytes etc. A signal is extracted from these flow files and analyzed using some of the wavelets supplied with the software. We note that as of this writing, a similar QoS and traffic engineering protocol called IP Flow Information eXport (IPFIX) [11] is being considered for standardization by the IETF. Vermont [20] is an open source flow monitor for IPFIX, similar to nProbe [14] for Cisco's NetFlow. A signal is extracted from these flow files and analyzed using some of the wavelets supplied with the software.

Next, in preparation of the framework for real time analysis, libpcap and tcpdump are used to capture, filter and analyze raw traffic. Initially we have to use some tests to establish some parameters for the wavelets, for example the window size. Therefore we start with filtered data to observe only the known anomalies, without any background traffic. For example, to observe a Neptune attack involving TCP SYN packets, the following tcpdump filter is applied:

tcpdump -r enet_tcpdump_09_30_2004 'tcp[13] & 0x02 = 2' dst port not 22 and dst port not 80 and dst port not 53 -w enet_tcpdump_09_30_2004.SYN

This filter reads only the TCP SYN packets (tcp[13] is the base filter for TCP packets, $0x02 = 2$ checks if the SYN bit is set) from the raw data, removes SSH, HTTP and DNS packets by examining the TCP header, and writes the filtered contents to the new file enet_tcpdump_09_30_2004.SYN, in libpcap format. This file is then replayed at an interface to observe the actual attack.

After initial tests establish the parameters for the wavelets, unfiltered traffic is analyzed. This paper presents only data resulting from unfiltered data, in order to demonstrate that our findings are applicable to arbitrary network traffic.

3.2 Waveman Framework

We design and implement a framework, called Waveman, to carry out a real time wavelets analysis. A representation of the Waveman framework is shown in Figure 1. Traffic is captured at an available interface using libpcap. Two counters corresponding to packet and byte counts are incremented on a per packet basis. It was observed that for the types of attacks analyzed in this work, there was a strong positive correlation between the packet and byte counts for the duration of the traces. So, it was assumed, for this work, either metric could be chosen without loss of generality over the other. This may not be the case for other types of attacks not considered in this work. To manage the capturing and sampling, two processes are used: one to capture the traffic on a per packet basis and update the appropriate byte and packet counters, and the other to access these counters via shared memory (shmget()), every 5 seconds.

Next, a time series signal is implemented in the form of a linked list data structure. A time series signal of packets vs. time (sampled every 5 seconds) is built, prepared and sent to LastWave. Since LastWave can also be used as a scripting language, we develop our own scripts

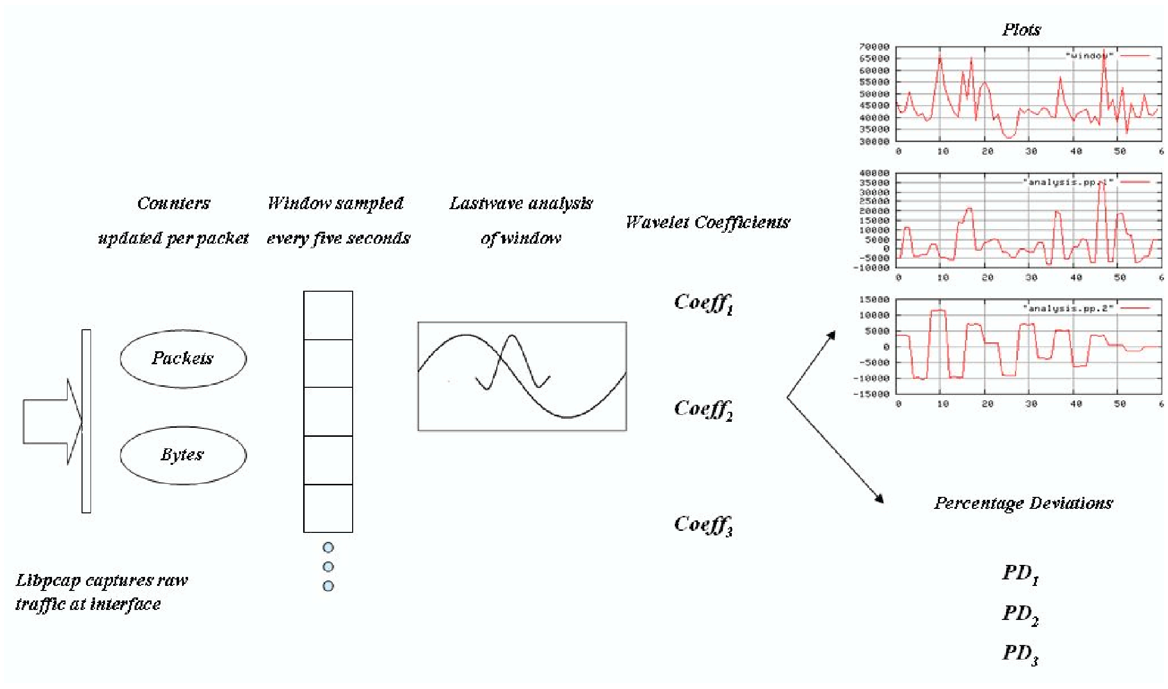


Figure 1: Framework of Waveman, a wavelet-based real time network traffic analysis

for the analysis, which are executed on a per analysis basis. The first three coefficients are of value to us (since any greater coefficients of the analysis would contain very sparse information), and these are calculated as the output of LastWave ($Coeff_1, Coeff_2, Coeff_3$ in Figure 1).

LastWave output is then processed, for purpose of normalization and ease of calculation of percentage deviations. The window we work with is five minutes long; i.e. five minutes worth of traffic, sampled every five seconds (these values are consistent with general network monitoring practices). Hence our window contains sixty samples. For the most part, this window size seemed suitable for the types of anomalies (and their short lengths) we are concerned with. This size is also consistent with the fact that a small window is good for localization. Several intermediate scripts are written in Perl to process and prepare the data for the next phase. The percentage deviations are calculated and recorded at each analysis. These values are normalized for ease of comparison.

In the last stage, Gnuplot is used to plot the graphs in the form of JPEG files, and an Apache Web server is used to serve the current results of the analysis to remote viewers. The graphs were plotted every five seconds by default, providing an updated real time snapshot of the current analysis every five seconds. Most of the framework and analysis work was done on a Pentium 4 (Hyperthreaded), 1 GB RAM, Gigabit interface NIC, running Fedora Core 3, and initial development and testing was done on a Dual Xeon (Hyperthreaded), 1 GB RAM, Gigabit NIC, running RHEL 3.

3.3 Wavelets

The wavelets that have been evaluated in this work are of the common families used in many research and commercial applications. The following representative examples of the wavelet functions are provided: Coiflet (COIF), Morlet (MORE), Daubechies (DAUB), and Paul (PAUL/MEX), as shown in Figure 2. The parameters (length/order) of each of these functions are varied. It

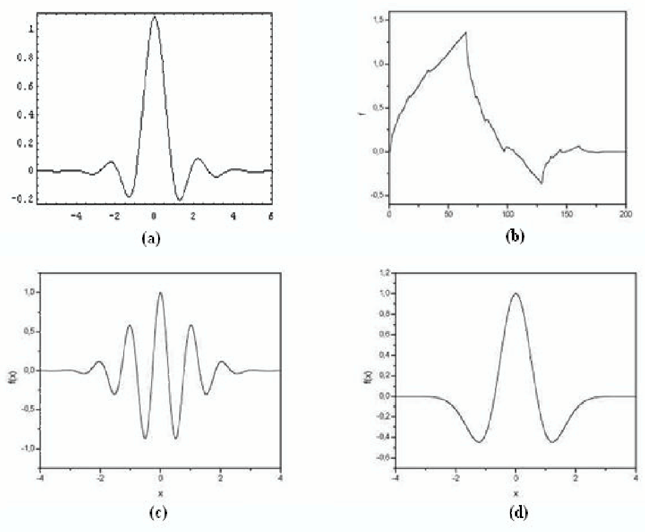


Figure 2: Wavelets evaluated in this work: (a) a Coiflet wavelet, (b) a Daubechies wavelet, (c) a Morlet wavelet, and (d) a Mexican hat or Paul wavelet

has been noted in [4] that a common method of deciding upon a wavelet for a certain time series signal is to choose a wavelet that most matches the variations in the data itself. This is an adequate technique when one is dealing with (semi)stationary signals, where the signal frequencies are constant throughout the signal. However, in the case of non-stationary signals (like network traffic), we do not have this privilege. Hence, no single wavelet can be easily matched to all the types of traffic and/or the anomalies discussed in this paper.

4 Evaluation

4.1 Anomalies Analyzed

Once the Waveman framework for the real time analysis was in place, we evaluate it with different anomalies. These anomalies include three types of Denial of Service (DoS) attacks, namely Neptune, Smurf and Mailbomb, and two types of portscan traffic, namely a simple portscan (ipsweep), and a stealth scan. The traffic that contains three DoS attacks was taken from the MIT Lincoln Laboratory Intrusion Detection System Evaluation Data [16], and the portscan traffic was collected from ERI, a domain name service company. A brief description of each anomaly is as follows.

1) Neptune Attack:

   Also known as a SYN flood attack, Neptune attack targets all TCP/IP implementations. When a tcpd server receives a SYN message, it reserves some of its resources for the expected connection (called half-open connections) and sends a "SYN-ACK" message to the requesting client. When the client receives the SYN-ACK message, it replies by sending back an "ACK" message to the server. If the server receives the ACK message, the connection is fully established and the two computers can start exchanging data messages over the connection. However, the data structure that a tcpd server uses to record all half-open connections is of finite size, which can be made to overflow if there is a large increase in half-open connections. When the half-open connection table on the victim server is full, the server is unable to accept any new incoming connection requests until the table is emptied out. Timeouts associated with each connection assure that the entry will eventually be cleared, but the attacker can keep up with a steady stream of SYN connection requests, which may lead to the crash of the victim [8].

2) Mailbomb:

   A large number of e-mail messages are sent to a victim user, by a compromised host connecting to the SMTP port of the mail server directly. This attack can result in thousands of unwanted messages delivered to a single user's account. In the Lincoln Lab data, this attack was crafted via a Perl program that created mail messages and connected to the SMTP port of the victim machine directly. The program was designed to accept the email addresses of the victims and the number of e-mail messages to send. A typical attack would send a total of around 10 MB of spurious mail to a user. In the simulation, the attack was not enough to effect the performance of the server or cause system failure (thereby compromising its security), but was more of a nuisance for the particular user targeted as the victim.

3) Smurf Attack:

   ICMP is a Layer-3 protocol which is used to convey status and error information including notification of network congestion and of other network transport problems. The smurf attack involves the attacker sending ICMP echo packets to the broadcast address of several subnets with the source address spoofed to be that of the victim's. This causes all hosts on each subnet to respond with ICP echo replies to the victim's address. ICMP echo replies are sent back by all the active hosts on each of these subnets. Amplification is achieved by using the broadcast address, resulting in a large flood of echo replies to the victim [9]. The victim and target subnet may suffer degraded network performance to the point that the network cannot be used.

4) Portscan:

   Portscan involves a remote host scanning TCP ports on victim machines running vulnerable services. There are two types of portscans. The first type is called vertical sweep, in which a single host is scanned for all open ports to determine what services are currently provided by the host. The second type is called horizontal sweep, in which a whole branch of network prefix could be scanned for the same open port. Vertical sweep is generally used by an attacker actively looking for open ports on an isolated machine, while horizontal sweep is usually the result of a worm on a compromised host, looking for other vulnerable machines. This simple scan involves the remote host sending TCP SYN packets to the corresponding port(s), and confirms that a port is open when the local host responds with an ACK. Very fast scans are possible in this way [24]. Several variations of this anomaly are possible, most of which attempt to conceal the scanner from the victim network. One of such variations is discussed in more detail in the following sub-section.

5) Stealth Scan:

   A stealth scan is called so because it is more difficult to detect, and many intrusion detection and prevention systems allow it to go unnoticed. Instead of sending a TCP SYN packet to a port on the target host like in a simple portscan, a FIN packet is sent to the port in question. According to RFC 793

[28], the correct response for a closed port to such a packet is to send a RST packet to the remote host. If the port is open, the FIN packet is dropped, and no packet is sent in response. It should be noted however, that Microsoft, Cisco, BSDI, HP/UX, MVS, and IRIX OSes do not exhibit this behavior (they deviate from RFC 793); they send RST packets from open ports as well. This behavior can be used by an attacker to determine which OS is running on remote machines [24].

A plot of each anomaly analyzed in this work is shown in Figure 3.

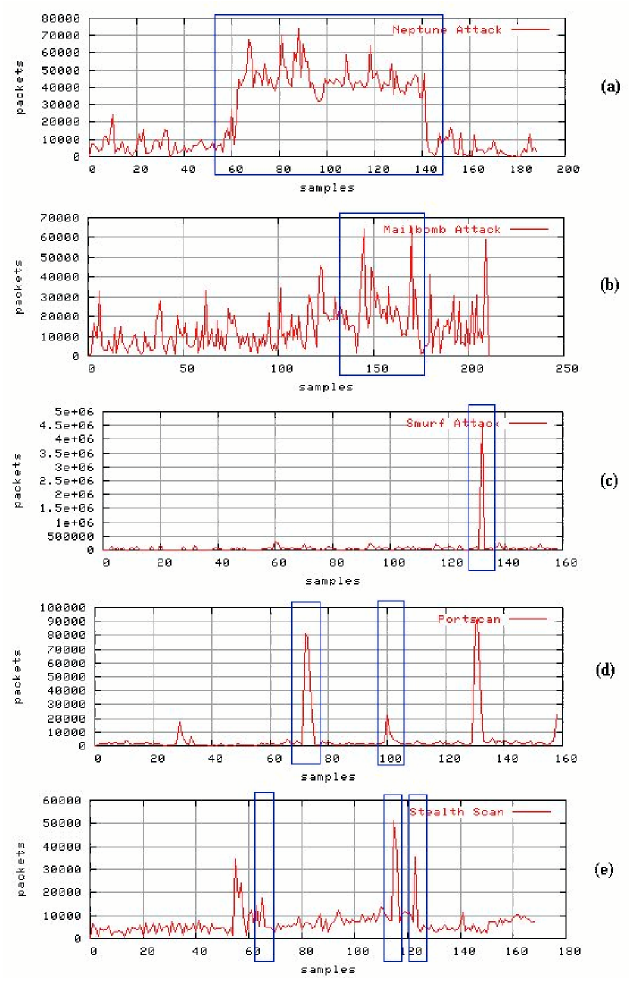


Figure 3: Plots of network traffic corresponding to the various anomalies analyzed in this work. (a) shows a Neptune attack, (b) a Mailbomb attack, (c) a Smurf attack, (d) a portscan and (e) a Stealth scan

4.2 Data Sets

As stated earlier, the above data has been taken from two data sets; they are discussed as follows.

1) MIT Lincoln Laboratory IDS Evaluation Data Set

This is second (1998, 1999, 2000) in a series of data sets created at MIT, under a DARPA sponsored project to evaluate intrusion detection systems, and to guide research directions. The network testbed shown below was modelled after a small Air Force base, including host computers that were attacked, and traffic generators that produced live traffic. The network is shown in Figure 4. Of the hosts shown, Calvin and Hobbes were used for the attacks, Solomon was used to sniff the inside traffic and Locke was used to sniff the outside traffic. This data set contains six weeks of traffic. Raw tcpdump files, with labeled attacks, were available for use in this work. It should be noted that the synthetic nature of Lincoln Lab data sets is not without criticisms [23].

2) EnetRegistry, Inc. data set (ERI)

ERI is an ICANN accredited registrar of Internet domain names. This research was partially supported by them in that they permitted capturing of raw traffic at their border routers, for use in this work. The second author was the system and network administrator for two server farms, one of which was collocated at a data center in northern Virginia. The traffic was collected at the data center routers. These "routers" are actually Dell PowerEdge 1750 servers, running Debian Woody Linux. These acted as routers as well as firewalls for the (separate) downstream LAN and DMZ networks as shown in Figure 5. The LAN is the trusted network, with restricted access, and the DMZ (Demilitarized Zone) is the non-trusted network, in which the public production servers are located. Rules are defined regarding the flow of traffic between the LAN and the Internet, the DMZ and the Internet and the LAN and DMZ. Two routers were used for redundancy. Both routers have 100BaseFX connections to the upstream ISP routers. ERI has two IP branches of 32 addresses each. Approximately ten weeks of traffic was collected.

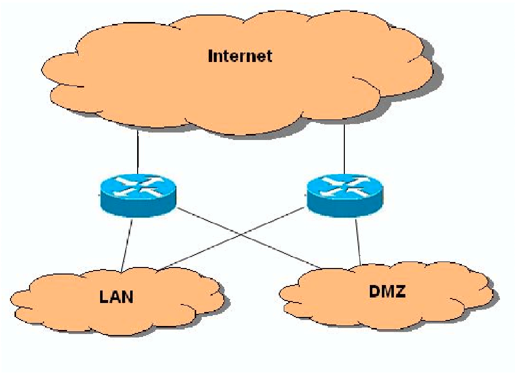


Figure 5: The ERI network at the collocated data center

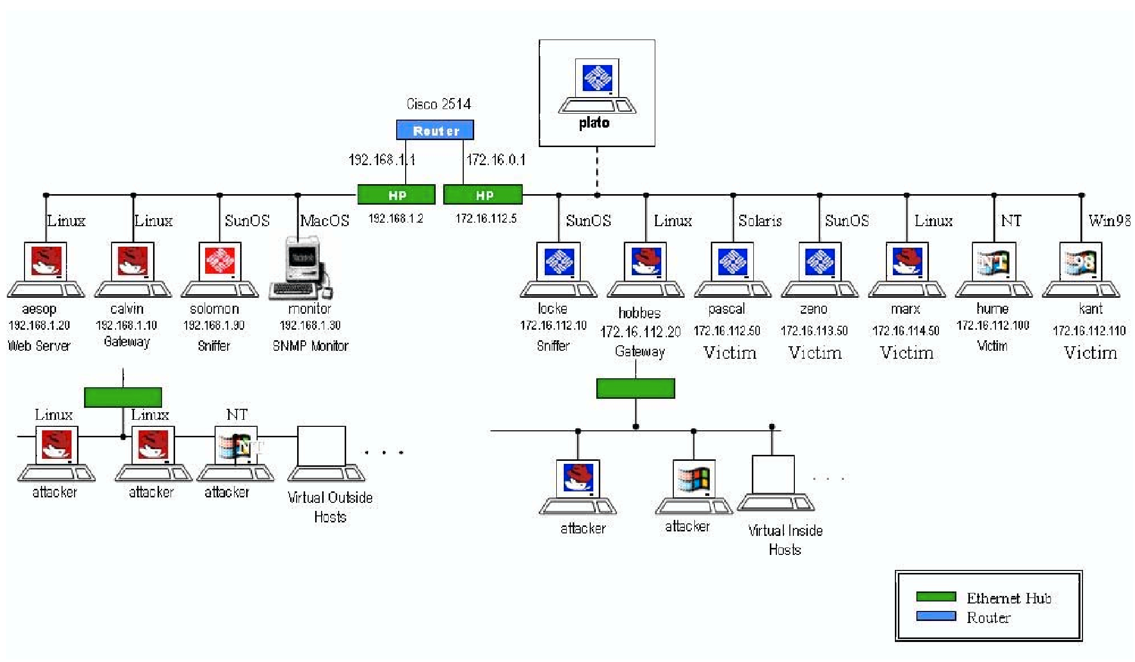


Figure 4: The MIT Lincoln Laboratory Simulation Network. (Source: http://www.ll.mit.edu/IST/ideval/docs/1999/Network_Topology.gif)

4.3 Procedure

To pinpoint the aforementioned attacks, we referred to the attack database (MIT), and manually searched through the raw data (ERI). Once the start and end points for the required attacks were established, the trace files were cut close to those points using tethereal, to obtain a new trace containing a few minutes of traffic leading up to the attack, the attack itself, and a few minutes of normal traffic after the attack. The number of sample points was calculated for each trace.

Next, the wavelet filters were designed. LastWave came with Daubechies 3 and Coiflet wavelets with arbitrary lengths. These wavelets were modified to fit our framework. Each wavelet listed above was designed with varying lengths.

Finally, the necessary options were set in the framework (the wavelet to use, window length, etc), and the analysis was started. At the same time, the attack traffic was replayed over an interface. All testing was done on an isolated network; none of the involved traffic was forwarded to any other network. Analysis began after waiting for the window to fill with samples (60 samples, 5 minutes). Thereafter, for the duration of the attack, the window was analyzed, the 1st, 2nd and 3rd wavelet coefficients noted, and then the window was slid forward one sample. Once the trace was complete, the analysis and traffic replay were manually stopped. At this point, the results were saved, the wavelet parameters and/or the attack traffic changed for the next iteration, and the analysis was started again. Thus we analyzed a total of 13 wavelets (4 COIF, 3 DAUB, 3 MEX, 3 MORE), each on 5 anomalies, and three coefficients per analysis. It should be noted here that the first coefficient of the analysis contains the high frequency information of the signal, which is usually "noise". The second coefficient, corresponding to the second octave for this work, contains the bulk of the information, and is of the most interest to us. The third coefficient (third octave) contains very sparse data, and is of less value.

4.4 Evaluation Metrics

In order to evaluate the performance of various wavelet functions on detecting different types of anomalies, we employ two metrics, namely percentage deviation and entropy.

1) Percentage Deviation

To compare and contrast the characteristics of the analyses, the percentage deviation of the coefficient value is calculated for each analysis. For all the sample points in a coefficient, the median is calculated. Then the percentage deviation PD for a sample value x is calculated as

$$PD_x = (x - median) * 100.$$

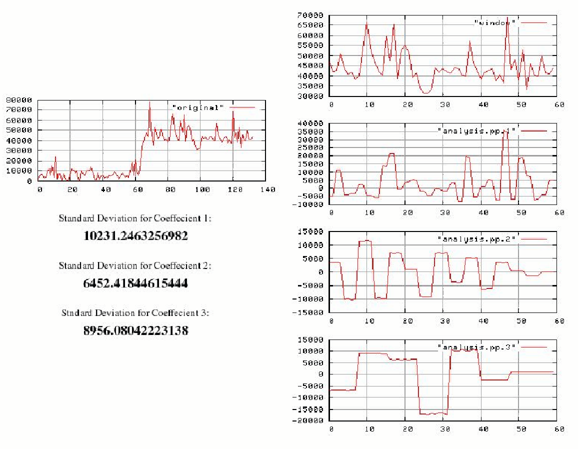
The rationale behind this is that those coefficients that display a lower PD are better, because the amount of deviation from the origin is indicative of an anomaly. To be more specific, a "better" wavelet should show a larger deviation at the locations of the start and end of an anomaly and show smaller deviations at all other locations in the signal, such that the contrast is larger and the anomaly is more identifiable. Since every trace contains only one or two anomalies (unless otherwise noted), a "good" candidate for the analysis would have the least deviation compared to the others.

Then, the mean PD per analysis is calculated as

$$PD_{avg} = \frac{1}{n}\sum_{i=1}^{n} PD_i,$$

where n is the window size. Thus, for a trace which is N samples long, we have $(N - n)$ PD values. A PD_{avg} value is calculated for each analysis, and these PD_{avg} values form the basis of the evaluation.

A typical point in the analysis of an NP anomaly using a COIF wavelet is shown in Figure 6. The plot on the left contains the time series signal corresponding to the raw traffic captured at the interface. The analysis is on approximately 130 samples, or a little over 12 minutes into the trace. The plots on the right correspond to the current window, which is 60 samples of the signal, up to its termination. The first plot from the top is a representation of this window. The following three plots are the first, second and third coefficients of the analysis. It can be seen that the first and second coefficients contain most of the information of the signal. Also shown are the percentage deviations for each coefficient, at this point in the analysis.



Standard Deviation for Coeffecient 1:
10231.2463256982

Standard Deviation for Coeffecient 2:
6452.41844615444

Stndard Deviation for Coeffecient 3:
8956.08042223138

Figure 6: Example of an analysis in progress

2) Entropy

In addition to percentage deviation, we also use an entropy-based method [2] to evaluate the performance of wavelets. Entropy is a type of information measure of disorder in signals and systems. A spontaneous change in a system disperses energy and increases its entropy. While the measurement of entropy is limited within the probability density function, one can extend the measure of information via the definition of the Rényi information as follows:

$$H_r(x) = \frac{1}{1-r}\log(\int_0^{T_{max}} f_x^r(t)dt), 0 < r < \infty, r \neq 1.$$

Note that $r = 1$ corresponds to the definition of the classical entropy. In order to consider general measure of information, we substitute $r = 3$, such that the value of the function does not need to be limited to between 0 and 1. For a discrete series of N samples, the entropy H is given by

$$H_3(x) = -\frac{1}{2}\log(\frac{1}{N}\sum_{i=1}^{N} f_x^3(i)).$$

An entropy analysis of the wavelet coefficients may be indicative of the properties of the wavelets, as shown in the evaluation results in next section.

5 Evaluation Results

During the analysis, it soon became evident that the first and second coefficients (corresponding to analysis at the first two octaves), had the sufficient information we require; any larger coefficients generally contained too sparse information to be of any value. Hence, for simplicity and clarity, we refer to just the first two coefficients for the remainder of this work.

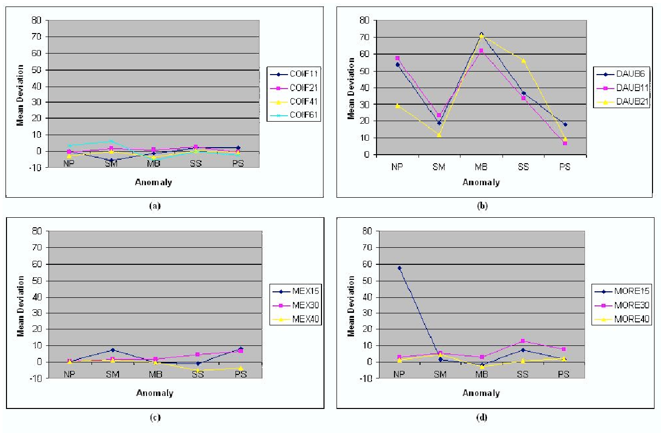


Figure 7: Mean deviation on various anomalies for (a) Coiflet wavelets, (b) Daubechies wavelets, (c) Mexican hat wavelets, and (d) Morlet wavelets

5.1 Results Based on Percentage Deviation

As is seen in Figure 7(a), the COIF.21 (Coiflet wavelet, length 21), with the lowest mean deviation (variance) values, shows the best characteristics across all the anomalies. The second best one is COIF.41 (Coiflet wavelet, length 41). The comparative results can be seen in the graphs. The anomalies are shown on the x axis, and the different wavelets on the y axis. The values plotted are the mean percentage deviations, per wavelet per anomaly. It can be seen that the Daubechies wavelets (Figure 7(b)) show poor characteristics for most anomalies, performing slightly better for the SM and PS attacks. Figures

8, 9, and 10 in the next subsection display the deviations in time, to compare the effectiveness in localizing the attack. The start and end points of the anomalies are marked by boxes in Figures 8, 9, and 10. Recall that according to our definition, a "better" wavelet should show a larger deviation at the locations of the start and end of an anomaly, and show smaller deviations at all other locations in the signal. This feature of good localization in time, along with low mean deviation values compose the characteristics of a good wavelet for this kind of analysis. Note that although the MORE wavelets (Figure 7(d)) attain low mean deviation values, they do not exhibit good localization in time, hence are not considered to be appropriate for these analyses. Hence, we consider the ability of the filter to exhibit favorable characteristics such as low number of false positives and negatives.

In [4], an indication of the lengths of the filters affecting the analysis has been suggested. Here, we demonstrate the varying characteristics of the lengths of the filters. It can be easily noted that non uniform characteristics are displayed for the same wavelet, but with different lengths. In the case of the better performing COIF, we increased the length of the filter to up to the window size, approximately 60 samples to see the effect. As is noted in the following graphs, the COIF.21 outperformed the COIF.41 and COIF.61 in terms of localization in time, and low mean deviation values. Similar traits are shown for the other filters as well.

5.2 Thresholds

On comparing the coefficients and PDavg values, it is evident that the MORE and DAUB wavelets perform poorly. Furthermore, it has been established in [4] that the MORE wavelet does not show good localization in discrete wavelet transforms, like what we have attempted in this work. It is more suitable for continuous transforms. Of the remaining wavelets, the COIF and MEX wavelets seem to show the best characteristics. From the experimental results, we can effectively claim that for the COIF and MEX wavelets, a 50 percent or greater deviation in frequency components strongly suggest an anomaly, when analyzed with a window size of 60 samples. Varying the window lengths and/or filter lengths may possibly lead to changes to these threshold values.

What follows is the per-anomaly based analysis, for a window length of sixty samples. The analyses of each of the five anomalies using different wavelets are conducted. Due to space limit, we only show the analyses of Naptune attack, Mailbomb attack, and portscan in Figures 8, 9, and 10 respectively. The percentage deviations are shown on the y axis, for each wavelet, for each anomaly. For the sake of brevity, we have included the second coefficient data only. All the data has been normalized. The legend of each graph follows the format: "anomaly.wavelet.coefficient.length.normalized". For example, in the first plot, **NP.DAUB.2.11.NM** refers to a **NP** attack, analyzed using the **DAUB** wavelet of length **11** samples. In addition, the data corresponds to the **second** coefficient of the analysis, and is **normalized**. Again, a better wavelet will show a large deviation at the locations of the anomaly start and end, with minimum deviations at all other locations in the signal. The start and end points of the anomalies have been marked by the addition of rectangles.

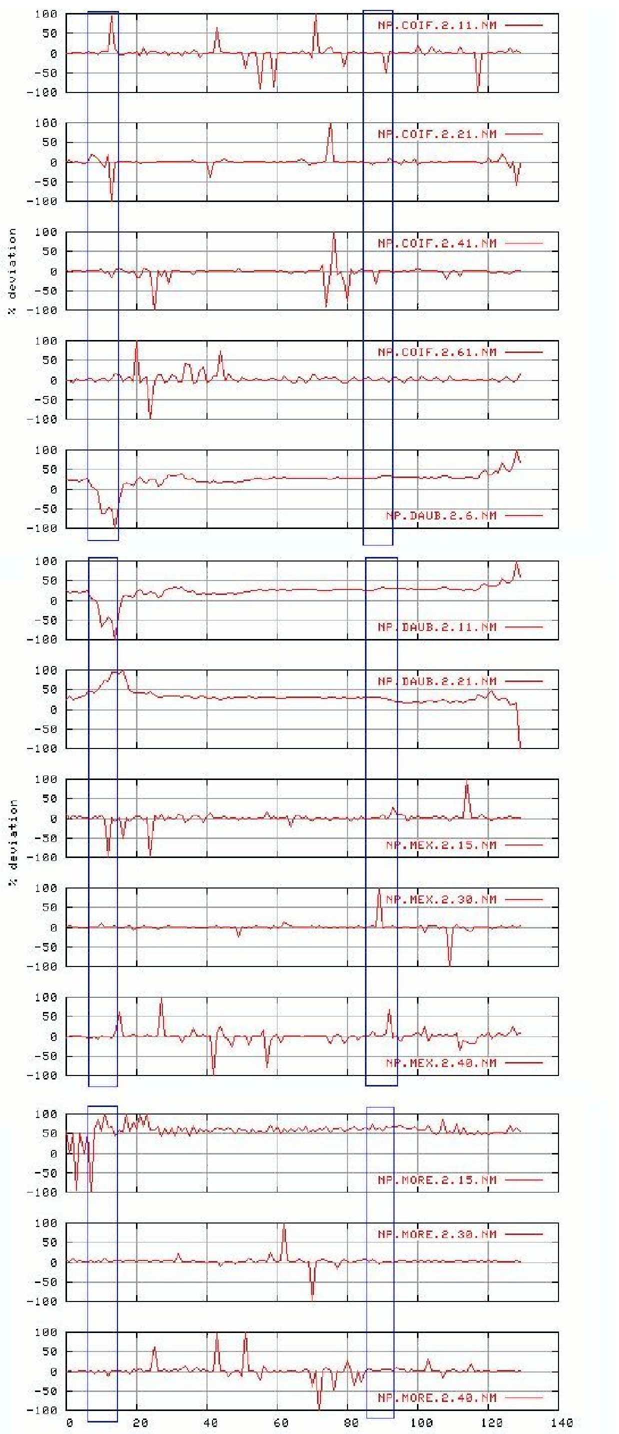


Figure 8: Analysis of Naptune attack using different wavelets

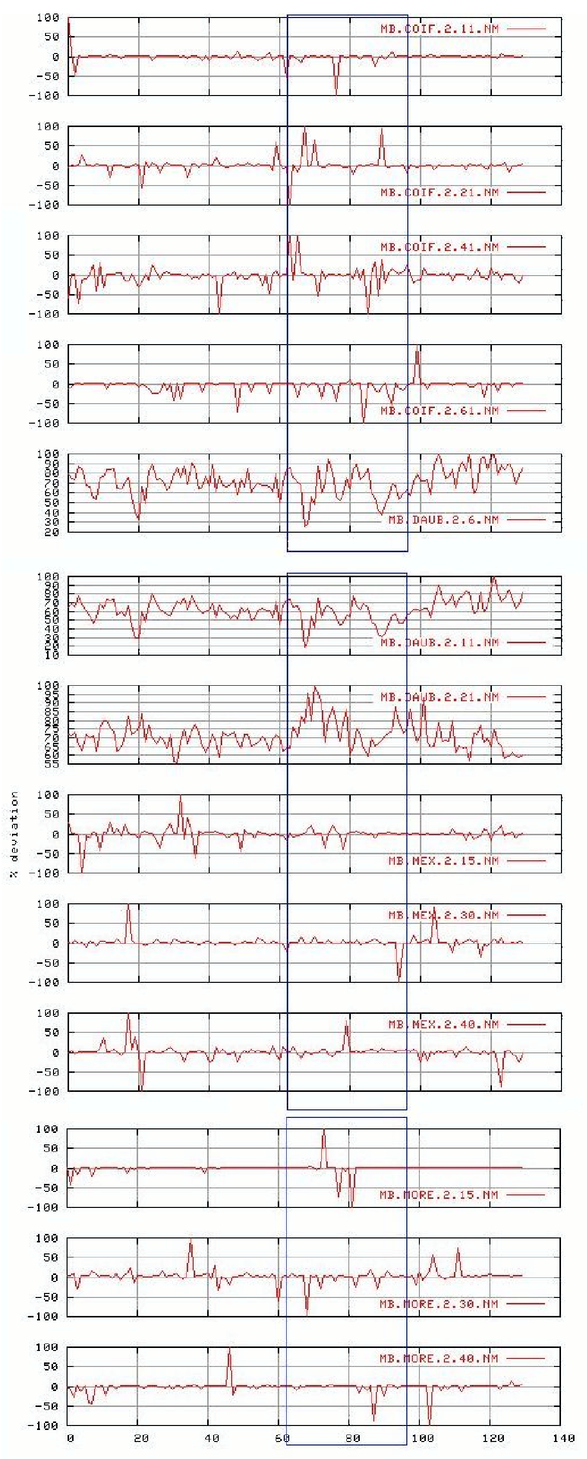


Figure 9: Analysis of Mailbomb attack using different wavelets

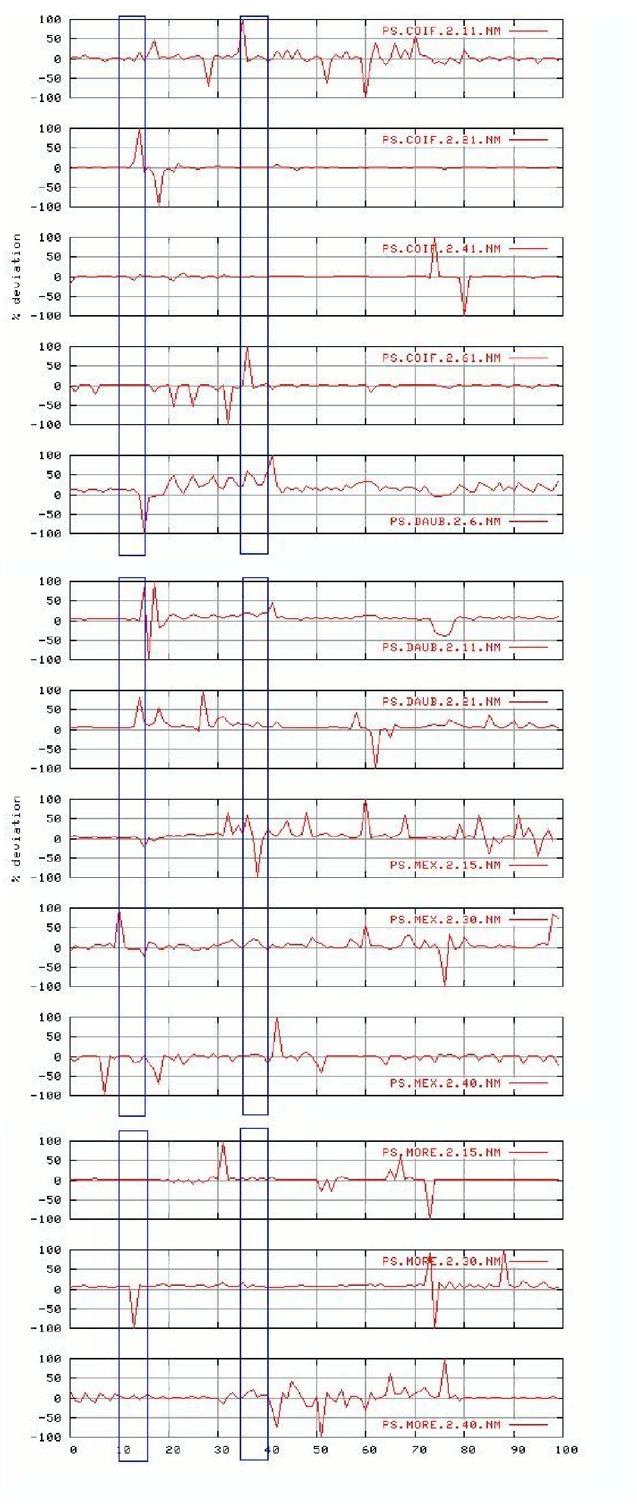


Figure 10: Analysis of Portscan using different wavelets

5.3 Results Based on Entropy

The evaluation results of using the entropy-based method as explained in IV.D.2 are shown in Figure 11. For this analysis, the window length was taken as one minute, and entropies were calculated for this window, analyzed every five seconds. All the wavelets were analyzed, against the Neptune attack. Figure 11(a) refers to the entropy function plots for the trace, for the Coiflet (upper plot) and Daubechies (lower) wavelets respectively. We can see that the Daubechies wavelet shows better characteristics for this attack (for a one-minute window) than the Coiflet wavelet. For verification and comparison, we show alongside it the evaluation results of percentage deviation method applied on the same trace, in Figure 11(b); the Daubechies does indeed show better characteristics than the Coiflet for this trace (for a one-minute window), according to this method as well.

Hence, we can derive the conclusion that the entropy-based method and the percentage deviation-based method generate consistent evaluation results in this case and they can both be used to evaluate the effectiveness of the wavelets on detecting and analyzing network anomalies.

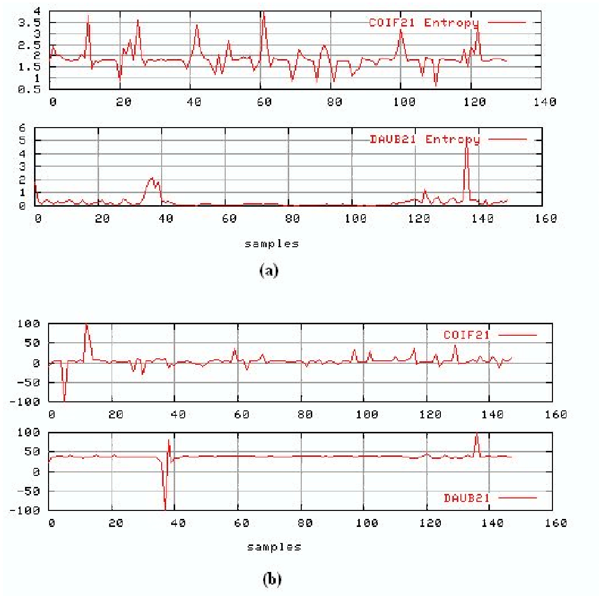


Figure 11: Results of entropy-based evaluation

6 Concluding Remarks

In this paper, we present a framework for real time wavelet analysis of network traffic. We have also evaluated various wavelets for the sole purpose of detecting short-term anomalies like Denial-of-Service attacks and port scans. Evaluation of the wavelets was based on twofold criteria: to have good localization in time characteristics, and to have a low mean deviation over the duration of the signal. The evaluation results show that Coiflet and Mexican Hat wavelets have better characteristics when faced with the anomalies considered in this work, based on a five-minute, sixty-sample window. We believe that the knowledge and experience obtained in this work could indeed be used to develop a wavelet-based real time intrusion detection system.

For future work, we will extend the framework to provide automated classification of anomalies detected in network traffic. To achieve this, we will construct a profile for each type of anomalies that describes common characteristics despite different strength and duration of single anomalous event. Additionally, we will employ multiple wavelet functions (of the same family) in parallel, each using a different window length, to detect anomalies of different strengths and durations.

Acknowledgements

This work is supported in part by an AFRL/DARPA grant (FA8750-04-2-0260). The authors thank EnetRegistry, Inc. for allowing us to use collected traffic for the purpose of analysis. The authors also thank Dr. John Zachary for his initial contribution. Last but not least, the authors thank the anonymous reviewers for their helpful comments and suggestions.

References

[1] E. Bacry, *LastWave 2.0*, http://www.cmap.polytechnique.fr/ bacry/LastWave/index.html

[2] R. Baraniuk, P. Flandrin, A. J. E. M. Jensen, and O. Michel, "Measuring time frequency information content using the Rényi entropies," *IEEE Transactions on Information Theory*, vol. 47, Issue 4, Apr. 2001.

[3] P. Barford and D. Plonka, "Characteristics of network traffic flow anomalies," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, 2001.

[4] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, 2002.

[5] S. Basu, and A. Mukherjee, *Time Sseries Models for Internet Traffic*, Technical Report GIT-CC-95-27, Georgia Institute of Technology, 1996.

[6] *Bro Intrusion Detection System*, http://bro-ids.org/.

[7] *CERT Advisory CA-96.01*, http://www.cert.org/advisories/CA-1996-01.html.

[8] *CERT Advisory CA-96.21*, http://www.cert.org/advisories/CA-1996-21.html.

[9] *CERT Advisory CA-98.01*, http://www.cert.org/advisories/CA-1998-01.html.

[10] B. Claise, *Cisco Systems NetFlow Services Export Version 9*, RFC 3954, Oct. 2004.

[11] B. Claise, *IPFIX Protocol Specification*, Internet-Draft (work in progress), draft-ietf-ipfix-protocol-22.txt, June 2006.

[12] M. Crovella and A. Bestavros, "Self-similarity in world wide Web traffic: Evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835-846, Dec. 1997.

[13] I. Daubechies, B. Han, A. Ron, and Z. Shen, *Framelets: MRA-based Constructions of Wavelet Frames*, preprint, 2001.

[14] L. Deri, "nProbe: An open source netflow probe for gigabit networks," in *Proceedings of TERENA Networking Conference (TNC'03)*, Zagreb, Croatia, May 2003.

[15] Fyodor, "The art of port scanning," *Phrack Magazine*, vol. 7, Issue 51, Sep. 1997.

[16] J. W. Haines, R. P. Lippmann, D. J. Fried, E. Tran, S. Boswell, and M. A. Zissman, *1999 DARPA Intrusion Detection System Evaluation: Design and Procedures*, in MIT Lincoln Laboratory Technical Report, 2000.

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

[18] M. Kim, S. Lam, T. Kim, Y. Shin, and E. J. Powers, "Wavelet-based approach to detect shared congestion," in *Proceedings of ACM SIGCOMM'04*, Portland, Oregon, Aug. 2004.

[19] C. Kruegel and G. Vigna, "Anomaly detection of Web-based attacks," in *Proceedings of 10th ACM Conference on Computer and Communication Security (CCS'03)*, pp. 251-261, Washington, DC, Oct. 2003.

[20] R. T. Lampert, C. Sommer, F. Dressler, and G. Münz, "Vermont - a versatile monitoring toolkit using IPFIX/PSAMP," in *Proceedings of IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM'06)*, Tübingen, Germany, Sep. 2006.

[21] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 1-15, Feb. 1994.

[22] U. Maimon, "Port scanning without the SYN flag," *Phrack Magazine*, vol. 7, Issue 49, Nov. 1996.

[23] J. McHugh, "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA off-line intrusion detection system evaluation as performed by lincoln laboratory," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 262-294, Nov. 2000.

[24] *Nmap: Security Scanner For Network Exploration and Security Audits.* http://www.insecure.org/nmap/.

[25] T. Oetiker, *RRD Tool*, http://people.ee.ethz.ch/õetiker/webtools/rrdtool/

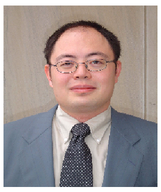
[26] D. Plonka, "FlowScan: A network traffic flow reporting and visualization tool," in *Proceedings of the 14th Systems Administration Conference*, New Orleans, Louisiana, 2000.

[27] R. Polikar, *The Wavelet Tutorial: The Engineer's Ultimate Guide to Wavelet Analysis*, http://users.rowan.edu/p̃olikar/WAVELETS/WTtutorial.html

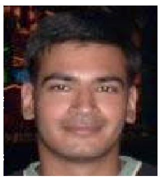
[28] J. Postel, *Transmission Control Protocol*, RFC 793, Sep. 1981.

[29] *Snort*, http://www.snort.org/

[30] H. Wang, D. Zhang, and K. G. Shin, "Change-point monitoring for detection of DoS attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 4, pp. 193-208, 2004.



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.



Sachin Thareja has a B. S. in Computer Science from Ambedkar University, India, and an MS in Computer Science from the University of South Carolina, USA. His interests lie in Network Programming, Protocols and Security, Systems Administration and Engineering, and Open Source implementations of these aspects of computing. During his Research Assistantship at USC he worked on signal processing techniques of analyzing network traffic, under the guidance of Dr. Chin-Tser Huang. He has worked for EnetRegistry Inc., Yamaha Music Interactive, and currently as a Network Systems Programmer for Bloomberg LP in New York City.

Yong-June Shin received the B. S. degree from Yonsei University, Seoul, Korea, in 1996 with early completion honors and the M.S degree from The University of Michigan, Ann Arbor, in 1997. He received the Ph. D. degree from the Department of Electrical and Computer Engineering, The University of Texas at Austin, in 2004. Upon graduation, he joined the Department of Electrical Engineering, The University of South Carolina, Columbia, as an Assistant Professor. His research interests include time-frequency analysis, wavelets, and higher order statistical signal analysis. His fields of application are power transmission and distribution, communications systems, measurement and instrumentation.