International Journal of Security and Its Applications
Vol.7, No.5 (2013), pp.81-100
http://dx.doi.org/10.14257/ijsia.2013.7.5.07

A Novel Multi Scale Approach for Detecting High Bandwidth Aggregates in Network Traffic

Gagandeep Kaur*, Vikas Saxena*, J. P. Gupta⋆
Deptt. of Computer Science and IT
* *Jaypee Institute of Information Technology, Noida, UP, India*
⋆ *Sharda University, Noida, UP, India*
Email: { gagandeep.kaur,* vikas.saxena}@jiit.ac.in*
⋆ jaip.gupta@gmail.com

Abstract

Today the scale, complexity and intensity of Denial of Service attacks has increased many folds. These attacks have moved from simple flooding based attacks to sophisticated Application based attacks as well as Protocol specific attacks.The challenge is to develop detection algorithms that can distinguish between the attacks like the new pulsating denial of service and legitimate traffic like Flash events. The presence of self-similarity in computer network traffic has introduced a newer dimension in techniques being developed for anomaly detection in aggregated network traffic.We propose use of wavelets to distinguish between legitimate flash events and pulsating distributed denial of service attacks and generating images to show point-of-presence of the attack.The detection methodology has also been tested on KDD Dataset.

KEYWORDS: Self Similarity, Wavelets, DDoS,PDDoS,flash events,KDD Dataset

1. Introduction

The Internet is growing at an unprecedented rate. With its wide spread use amongst network users it has become the favorite platform for network criminals and hackers. The number of intrusions into computer systems is growing and raising concerns about computer security. The ever increasing list of Computer Emergency Research Team (CERT) [1] is proof enough of the urgency to look out for robust security mechanisms. High bandwidth aggregates or Denial of service (DoS) attacks are one such security threat in the computer network traffic. In a computer network a denial of service (DoS) attack tries to degrade the quality of service available to the legitimate users. Distributed Denial of Service (DDoS) uses large number of DoS to attack the service providers from distributed nodes. Therefore DDoS attacks are very lethal and hamper the services within seconds of their deployment [30]. More recently a new breed of DDoS attacks has been discovered and is named as Low-rate DDoS attacks or LDoS[32]. Pulsating DoS or PDoS is a similar variation of the former attack. In LDoS instead of flooding the communication channels with large number of packets timed packet pulses of small duration are sent to the target machine. Because there is sudden increase in the number of packets for a short duration of time these pulses help the attacker to disrupt TCP services. When combined with distributed architecture, these pulses of still shorter duration can be so generated such that on reaching their target the impact is very high number of packets for short duration. The attacker in PDoS attacks periodically sends high-rate traffic over a

ISSN: 1738-9976 IJSIA

short period of time. Ironically anything that is consuming high bandwidth is not an attack. Flash crowds are one such example.A flash crowd is a large spike or surge in traffic to a particular Web site. Major news Web sites experience this problem during major world events. As per the paper [33] a flash event has three phases namely ramp-up phase, a sustained traffic phase and a ramp-down phase. Considering regular every day traffic as normal in case of flash event the rate of traffic sees a large surge in traffic for small fraction of time, stays in the sustained phase for small fraction of time and then decays very fast in the ramp-down phase.

The work done so far in the detection and categorization of anomalous traffic in the computer network traffic especially denial of service attacks deals with the principal challenge of automatically detecting and classifying these that can span a vast range of events and even to new, previously unknown events. An anomaly detection system should therefore be able to detect not only the anomalous behaviour of data in the signal but shall also be able to categorize them.Considered as an alternative to the traditional network anomaly detection approaches or a data preprocessing for conventional detection approaches, recently signal processing techniques have been successfully applied to the network anomaly detection due to their ability in point change detection and data transformation [2, 16]. Self-Similarity in the network traffic is one such feature. It measures the repetition in a pattern at different levels of aggregation. Wavelets are an excellent tool for computing aggregates at different scales. Combined with Entropy variations in the network traffic we can therefore look out for presence of DoS attacks in the traffic.

In the study and analysis of computer network attacks publicly available Datasets play a very crucial role. Researchers use these Datasets for testing and validating their attack detection algorithms. Lots of datasets are publicly available that are daily used by researchers for testing and validation. The most widely used evaluation datasets are the KDD Cup 1999 and its modified version, the NSL-KDD dataset [13]. The KDD Cup 1999 dataset is the benchmark dataset for intrusion detection. Although the KDD CUP dataset has many flaws [16], yet it is the only dataset that is publicly available for Intrusion Detection. It has labeled attack samples which are obtained by passive monitoring, rather than by inserting the attack packets into the normal traces. Therefore, we have selected this dataset for validation of our detection algorithm.

The main objective of our work can therefore be stated as to use wavelets to test self similarity in the network traffic and hence measure entropy changes at different scales.In doing so we look out for change in patterns that can be categorized as outliers and can be further tested for attacks both old as well as modern day Denial of Service attacks.

This paper covers related work to our approach,definitions, detection methodology, simulation model , results, validation using KDD Dataset and discussion & conclusion.

2. Related Work

V.Alarcon-Aquino and J.A.Barria brought in the application of Wavelets into Network Intrusion Detection [3]. Many of the approaches rely on known statistical properties of normal traffic when the observed traffic deviates significantly from the normal behavior. Initial line of Intrusion Detection Systems (IDSs) used spectral techniques. Current applications of spectral techniques look for high-frequency occurrences to identify anomalous behavior [4][5]. The wavelet tool allows a single signal to be decomposed in several signals representing different frequencies. High frequencies indicate spontaneous behavior by traffic while low frequencies exhibit global behavior by traffic. Methods of detection involve finding global and local variances in wavelet coefficients to detect respective short and long-term anomalies. Hussain et al. [7] apply spectral techniques to time se-

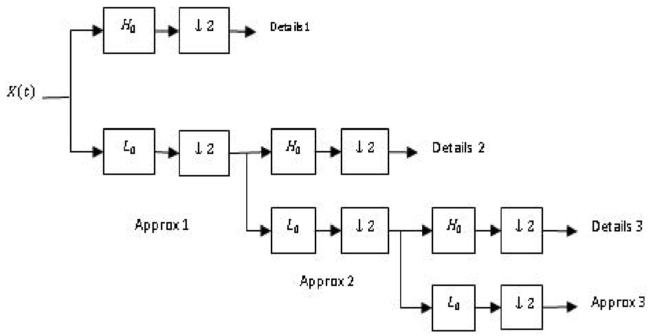


Figure 1. Recursive pyramidal algorithm for the multi-scale wavelet transform

ries of packet arrival times. Based on spectral characteristics, they are able to distinguish between single- and multi-source attacks, and identify repeat attacks. Barford et al. in [4] use wavelets to analyze Simple Network Management Protocol (SNMP) and flow-level information to identify DoS attack and other high frequency anomalies. Magnaghi et al. in [8] detect anomalies within TCP flows using a wavelet-based approach to identify network misconfigurations. Spectral techniques have also been employed to identify bottleneck links [9] and routing information. G. Bartlett et al. in [14] look at periodicity between flows to identify hosts which maintain regular contact while considering low frequency behavior under long observation windows and use iterated filtering for full decomposition. Carl et al. in [10] applied wavelets transform for detecting change-points in the Cumulative SUM (CUSUM) statistic. Hamdi and Boudriga in [11], Xunyi et al. in [12], Dainotti et al. in [19] devise wavelet techniques for detecting DoS attacks. Lu et al. in [13] study intrusion detection performance with wavelet basis functions and state their impact on detection mechanisms.

3. Theoretical Background

Self Similarity: The self similarity or long range dependence (LRD) of network traffic was found by Leland et al. [15] in 1993. The degree of self-similarity is measured with Hurst parameter, H [23]. The essence of self-similarity is that the same pattern is repeated at different levels of aggregation by time sequences. Therefore, if an object is self-similar, its parts, when magnified, resemble-in a suitable sense-the shape of the whole.

Wavelets: Wavelets are an excellent tool suitable for long discrete data sequences. As per definition, any signal can be decomposed by using a dyadic discrete family $\{2^{\frac{j}{2}}\psi(2^j t - k\}$ which is an orthonormal basis in L^2. The translation (k) and scale (j) functions are based on mother wavelet (ψ). Wavelet analysis therefore defines a collection of nested subspaces V_j that satisfy $\{\ldots V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \ldots\}$ where $\{\overline{\cup_{j\epsilon} V_j} = L^2 and \cap_{j\epsilon} V_j = \phi\}$.

The wavelet transform can therefore be represented as a pyramidal combination of low-pass (L_0) and high-pass (H_0) filters derived from scaling function and mother wavelet chosen for transformation as shown in Figure 1 .

The outputs of the low-pass are termed as approximations ($Approx$) and outputs from high-pass are termed as details (det). An input signal can therefore be represented in terms of wavelet function ψ and scaling function ϕ based on the given dilation equations:

$$\phi_{j,k}(t) = 2^{\frac{-j}{2}}\phi_0(2^{-j}t - k), k\epsilon Z \tag{1}$$

$$\psi_{j,k}(t) = 2^{\frac{-j}{2}}\psi_0(2^{-j}t - k), k\epsilon Z \tag{2}$$

The approximation coefficient $Approx$ is given by $approx_j(t) = \sum_k a_x(j,k)\phi_{j,k}(t)$ and detail coefficient det is given by $detail_j(t) = \sum_k d_x(j,k)\psi_{j,k}(t)$.

Hence signal can be written as:

$$x(t) = approx_N(t) + \sum_{j=1}^{N} detail_j(t) \tag{3}$$

Where $d_x(j,k)$ is the Discrete Wavelet Transform (DWT) of $x(t)$.

Energy Measurement: The Internet in its present size [29] is very complex. This system is difficult to measure and analyze. On the other hand statistical methods have proved to be useful in aggregation methods and analysis. Entropy statistics are one such measure. Entropy in general measures level of disorder in a system. A sudden increase or decrease in the system disperses its energy and hence increases its entropy.

We know from [20],[21],[22], [24], [25] and [31] that for mean-zero stationary stochastic processes the variance of the DWT is $\{Ed_x^2(j,k) = \int_{-\infty}^{\infty}\Gamma(2^{-j}v) \mid \hat{\psi}(v) \mid^2 dv\}$ where $\{\hat{\psi}(v) = \int \psi(t)e^{-j2\pi vt}dt\}$.For LRD processes the equation becomes $\{Ed_x^2(j,k) \sim 2^{j\alpha}C(\psi,\alpha),\}$ where $\{C(\psi,\alpha) = \int \mid v \mid^{-\alpha}\mid \hat{\psi}(v) \mid^2 dv\}$. Energy associated with every resolution can be estimated by performing time average of wavelet detail coefficients at a given resolution j,$E_j = \sum_k \mid d_x(j,k) \mid^2$. Similarly, for a signal of length N, the mean wavelet energy at resolution j comes out to be $\tilde{E}_j = \frac{\sum_k E|d_x(j,k)|^2}{N_j}$, where $j = 1,\ldots,\log_2 N$. An entropy analysis of wavelet coefficients can therefore be used for attack detection.

4. Detection Methodology

Our objective in this work is to distinguish DDoS,PDDoS and flash events. We propose our detection methodology for the same. We have used entropy based detection mechanism and visualization techniques for attack detection.The simulation is run for the required amount of time and trace file is obtained for the ingress traffic at the access router.The aggregated data values are output to the trace file every 10ms and stored in the tuple format $< timestamp, no.ofbytes >$. The sampling sliding window is of 10ms and for each window entropy is calculated.

4.1. Detection and Characterization of the attacks

4.1.1. Detection of attacks:

Let us define total incoming traffic at the server as $ser\_tot(t)$ then we can say it to be equal to normal incoming traffic $normal(t)$ and incoming attack traffic $attack(t)$.Hence we can write it as : $ser\_tot(t) = normal(t) + attack(t)$ where t is time. Under normal conditions the incoming server traffic is due to regular legitimate users only but during attack this traffic is comprised of both normal traffic and DDoS traffic.

Mathematically, we can define the incoming traffic as a random process $\{X(t), t = j\triangle, j\epsilon N, \triangle\epsilon N\}$, where $\triangle$ is a constant sliding factor,N is set of positive integers and for each $t, X(t)$ is a random variable. Here $X(t)$ represents count of incoming bytes of network traffic in $\{t - \triangle, t\}$ interval. The task is accomplished in three phases: Hurst Computation, Multi Scale Analysis with Wavelets and Image Plotting. The reader can also refer to [34] though to keep the continuity the detailed description is provided here.

Phase I:Hurst Computation

Our first goal is to test self similarity in the network traffic. We know that network traffic exhibits long range dependence and hurst parameter H is a measure of self-similarity [4]. Under normal circumstances the network traffic follows the self similar behavior which can be measured with H. Whereas in the case of DDoS attacks the LRD nature of the traffic changes because DDoS attacks and their detection are short-range phenomena. It is because of the fact that the attackers emit packet bursts and attack the target for short periods of time only. This in turn changes the self similar behaviour of the traffic.

Algorithm 1 Hurst(L)

Require: len=length(L),x=L(:,1), y=L(:,2), $tlower$=1
Ensure: Hurst value
```
while tlower <= len do
  tupper=tlower+9;
  if tupper <= len then
    sumx=0; sumy=0; sumsqu=0; sumprodxy=0;
    for h=tlower:tupper do
      sumx=sumx+x(h); sumy=sumy+y(h); squ=x(h) .* x(h);
      sumsqu=sumsqu+squ; prodxy=x(h) .* y(h); sumprodxy=sumprodxy+prodxy;
      n=(tupper-tlower)+1; A=[n sumx;sumx sumsqu]; b=[sumy;sumprodxy];
      N = max(size(A));
      {Perform Gaussian Elimination}
      for j=2:N do
        for i=j:N do
          m = A(i,j-1)/A(j-1,j-1);
          A(i,:) = A(i,:) - A(j-1,:)*m;
          b(i) = b(i) - m*b(j-1);
        end for
      end for
      {Perform back substitution}
      bx = zeros(N,1); bx(N) = b(N)/A(N,N);
      for j=N-1:-1:1 do
        bx(j) = (b(j)-A(j,j+1:N)*bx(j+1:N))/A(j,j);
      end for
    end for
    gamma=bx(2,1);
    hurst=(1+gamma)/2;
    Write hurst To File
  end iftlower=tupper+1;
end while
return
```

Let us consider the incoming traffic at the server as discrete time series of length P. Divide it into N non overlapping sections. Each section is divided into M non overlapping segments. Divide each segment into K non overlapping blocks. Each block is of L length. Let $m(n)$ be the hurst value at aggregation level L in the segment of the section $(m = 0, 1, \ldots, M-1; n = 0, 1, \ldots, N-1)$. The H values for all the segments of the series are never the same so we can find the average H value $H(n)$ within a confidence interval. When the system is under attack we represent the changed

hurst value as $H(c)$.
Let us assume that at time t_a the attackers start emitting attack packets and as a result the system comes under attack. We can therefore say that before time t_a the system is under normal state and is in attack state after that. Let t_D denote our estimate on t_a. At time t_D we have total incoming traffic at server as $serv_{tot}(t_D) = normal(t_D) + attack(t_D)$. As a result following event triggers attack detection:

$$\varphi > \delta \qquad attack\ detected \tag{4}$$

where $\varphi = \| H_n - H_c \|$ represents the deviation of H of monitored traffic time series and $\delta > 0$ is threshold.However $\Gamma = \| H_n - H_l \| > \delta$ means false alarm. The algorithm details for Hurst calculation are explained in Algorithm 1.

Phase II: Multi Scale Analysis with Wavelets
In the second phase our goal is to find periodic behavior in Network Traffic and generate an Image of it. For this we have used Daubechies Discrete Wavelets that can be used as an iterative filter bank. We want to detect and distinguish between low-rate pulsating DDoS attacks and Flash crowds.For a multi-scale decomposition the Wavelet coefficients define the energy for a given Network Traffic series. A concentration of energy in the given level signifies the presence of a periodic signal and can be visualized for periodic PDoS patterns. The major benefit of using Wavelet is that energy is independent component at each level of decomposition. Therefore energy can be normalized at each scale independently. Once a presence of concentrated energy is detected we need to trace its point of presence. We therefore need to map the scales of wavelet decomposition to periods of time. We have used daubechies-6 filter for our computations. The filtered signal is down-sampled by 2 at each level of the analysis procedure; the signal of each level has an effect that sampling period extends 2 times. What it means is that if we have a sample of *t* seconds then the time range at level *j* will extend it to $t * 2^j$ seconds. And we know that for a discrete signal of length *n*, DWT can consist of $log_2 n$ levels at most. For instance, when we use 1-minute sampling interval,the cD1 equals to $1 * 2^1 = 2$ minutes, the cD2 equals $1 * 2^2 = 4$ minute interval and so on. For our study we have taken 10ms interval and therefore we decompose to 13th level.

For a given incoming traffic we generate a signal and find out the wavelet energy at resolution level j for the time window i as $\tilde{E}_j^{(l)} = \frac{\sum_{k=(i-1).L+1}^{i.L} C_j^2(k)}{N_j}$ with $i = 1, \ldots, N$, where N_j represents the number of coefficients at resolution j corresponding to time window i, whereas total energy estimator in this time window is $\tilde{E}_{tot}^{(l)} = \sum E_j^{(i)}$.The wavelet energy at resolution j is $< E_j >= \frac{\sum_{i=1}^{N} E_j^{(i)}}{N}$. Total wavelet energy average is therefore given as $< E_{tot} >= \sum < E_j >$.Algorithm 2 details the energy calculation with daubechies wavelet.

Phase III:Image Plotting We know that number of coefficients generated per scale by daubechies wavelet are different.The image to be plotted required the vectors for all scale levels to be of equal length. Therefore we extrapolated the received coefficients in order to generate vectors of equal length.The received Image has time on the x-axis and levels of decomposition on the y-axis starting from 1 to 13. There are 13 rows plotted as different levels in the Image and each one of them is independent of each other. Because these levels are independent of each other therefore we can normalize the energy values per row and can plot the normalized ones. For better visualization of the Image we have assigned colors to these values. Matlab function imagesc has been used to draw images. It uses a color scale from 1 to 64 with blue at 1 and red at 64.For each value in the row vector imagesc draws a bar of color depending upon the value in the vector.

Algorithm 2 Calculate Approximation Coefficients

```
Require: j=13, [C3,L3]=wavedec(x,13,db)
Ensure: approximation coefficients at different levels.
  {Calculate coefficients,squares, sum of squares, average, tot energy, log of avg.}
  for i = 13 to 1 do
    cDi=appcoef(C3,L3,db,i);
  end for
  for i = 13 to 1 do
    lDi=length(cDi);
  end for
  for k = 13 to 1 do
    for i = 1 to lDk do
      sqDk(i)=cDk(i)*cDk(i);
    end for
    for j = 13 to 1 do
      totenergyjD(j)=sum(sqDk); meanenergyjD(j)=(totenergyjD(j)/lDk);
      aveg(j)=-meanenergyjD(j); logDk=log2(meanenergyjD(j));
      log(j,1)=j; log(j)=logDk;
    end for
  end for
  for k = 1 to 13 do
    for i = 1 to lDk do
      WRITE sqDk(i) TO File
    end for
  end for
  for t = 1 to length(log) do
    WRITE t, log(t) TO File
  end for
  for t = 1 to length(aveg) do
    WRITE aveg(t) TO File
  end for
  for t = 1 to length(totenergyjD) do
    WRITE totenergyjD(t) TO File
  end for
  return
```

In our case we have hotter color like red when the energy is high and cooler color like blue when normal traffic is there. The details of image plotting are given in Algorithm 3.

Thresholds: Knowing that the energy values from non-periodic events show dispersion when seen at different levels of decomposition and therefore we decided to use energy threshold. We see that energy is less for non-periodic events and strong for periodic events. We estimate when the events started and stopped by looking at the timeseries of coefficients corresponding to that frequency range. Recall that if we have a sample of t seconds then the time range at level *j* will extend it to $t * 2^j$ seconds. In other words we can say each row value in the decomposition contains a timeseries $t * 2^j$ and therefore each row vector value *i* in this timeseries indicates a time x_i^j. Now in order to detect the start and end of a periodic pulse in time we look for a consecutive series of

Algorithm 3 imagels(dosf,tcpf,period,leng,name,g,x)

Require: $icolormap$=0, colorvector=(1:-.01:0)';
Ensure: 13-levels Network Traffic Image.

```
{Design a special colormap}
Set mycolormap=[colorvector colorvector colorvector];
if icolormap == 0 then
   colormap('default');
else if icolormap == 1 then
   colormap('gray');
else if icolormap == 2 then
   colormap(mycolormap);
end if
{Draw Image}
image(x(1:13,1:1:4096));
i=g;
save Image;
return
```

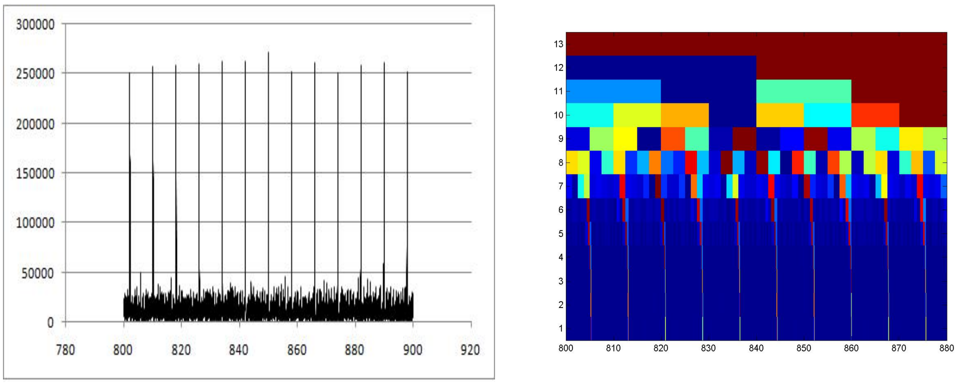


Figure 2. A Network Traffic pattern and Image of the pattern

strong coefficients of x_i^j. Our objective is to be able to detect periodicity in the PDoS pulses and for that we follow a simple technique. Since we have normalized our values in the range of 1 to 64 with red as a hotter color and blue as a cooler color we realized after extensive study of the vector values that anomalies can be linked to color range from yellow to red.And as a result we read the vector values equal to or greater than yellow which is actually the mean value of the color scale. We therefore search for the mean value and then set the counters mean forward and mean backward. Mean forward is used to find next mean value in the future and mean backward is used to find last mean value in the past. Their distance in time is calculated and if the difference is equal with allowable deviations we declare it as a PDoS pulse.

*Example:*Figure 2 shows a snapshot of captured computer network traffic from 800 to 900s and its generated image. The image clearly identifies the regions where there is increase in entropy values.Dark colored bars are visible for high traffic bursts.

5. Simulation Model

Due to lack of publicly available datasets for testing of DoS attacks especially the recently emerged PDoS attacks we have simulated their attack patterns using network simulator NS2. The works by Mirkovic in [26] have been referred to for creating the test environment. There are total of 200 nodes following three level hierarchies: domain, cluster, and nodes generated with GT-ITM topology generator. The edge nodes are host nodes and intermediate nodes act as access routers and core routers. Access routers are connected to core routers. Links between core to core routers are of 100Mbps, core to access router are of 10 Mbps and access router to edge nodes are of 5Mbps with edge probability of 0.85. Backbone link delays are of 0s.

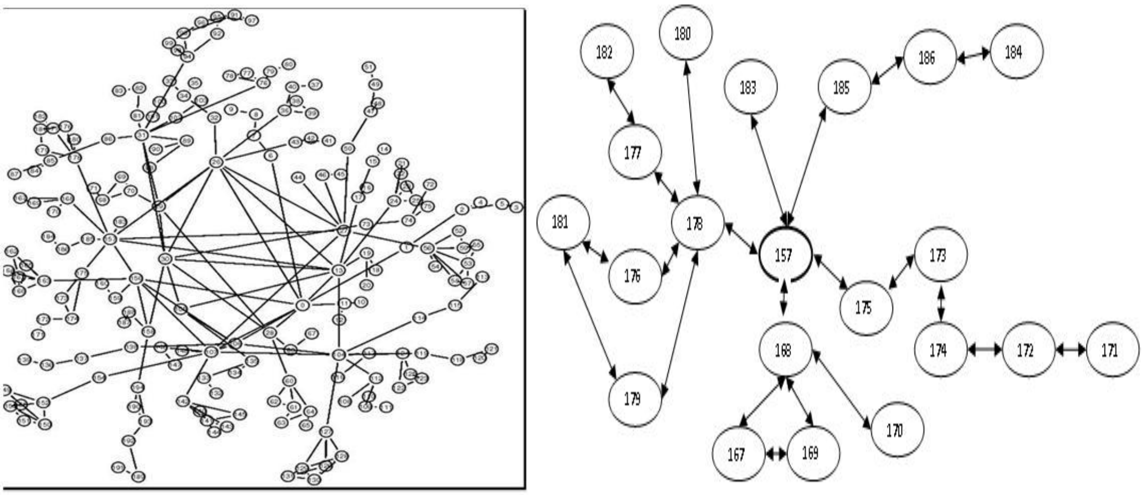


Figure 3. 200 nodes GT-ITM generated network topology and its Zoom in at Gateway node

Figures 3 show the simulation topology. The system consists of following components:

5.1. Clients

We discuss two main cases.

Case 1: There are two types of clients in our network. Majority of the clients are legitimate users of the service whereas a small number of clients are illegitimate users or the attackers. According to [29] the traffic is application-based and more than 90 percent of it consists of web and P2P. We therefore have application-based traffic between legitimate clients and server from ftp, voip,http and exponential traffic for flash events. There are three types of legitimate clients. Firstly, the clients that run FTP application on TCPReno(a type of TCP implementation), second ones are VoIP application and third ones are web based service based on HTTP. We used the following modules in NS-2 to generate the traffic: Pagepool/Webtraf for Web, Pagepool/Webtraf with $pagesize\_$ for ftp, UDP packets of packetsize 210 with burst time of 180ms and idle time of 50ms to simulate VoIP.

To model flash event the mathematical model given in [33] has been followed and set the packetSize 800 bytes, burst time 0ms, idle time 100ms and 10000000k rate for Pareto distribution.Total number of flash event generators is 20 and there is one sink.The attack traffic has been simulated using CBR packets following the bot attack pattern as per Moore [27].

Table 1. Topology generator parameters

S. No.	Parameter	Value
1	Domains	10
2	No. of Stub Domains	4 /domain
3	Edge probability	0.85
4	Number of nodes	5 /stub-domain
5	Backbone link B/Ws	100Mb for domain 10Mb for stub-domain 5Mb for hosts
6	Backbone link delays	0s

Case 2: We have TCP senders $C_1 - to - C_n$ and attackers from $A_1 - to - A_n$. All senders and attackers have one sink S. The traffic goes through bottleneck link R1-R2-S. Bottleneck link bandwidth is 10Mbps and all other links have 100Mbps bandwidth. The start time of each TCP sender is a random variable uniformly distributed from 0 to 1500s. The attackers begin to send pulses at 20s and stop at 1000s. We have used traffic trace to simulate train of square pulses [32]. The size of attack packet is 50 bytes. As shown in Figure 4 the attacker nodes send small cooperative pulses such that when their pulses reach the router the combined effect is one large burst of traffic that inundates the router. The simulations were carried out for low, medium and high traffic rates. Different scenarios were studied for 5,20,50 and 100 DoS flows. Burst lengths consisted of 50, 100, 150 and 200ms for periods ranging from 2500 to 200,000ms.

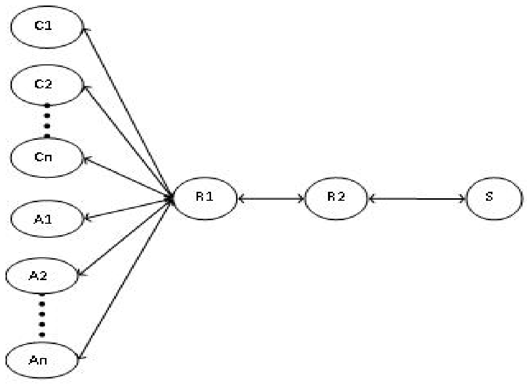


Figure 4. Simulation topology for case 2

5.2. Server

The simulation parameters for server side are mentioned below:

Case 1&2: The server provides generic TCP-based service. During normal scenario the legitimate clients connect to the server and avail satisfactory services from the server. When the network is under attack the zombies either send flooding packets or pulsating pulses towards the server and hence the network suffers congestion in its bottleneck links. Because of heavy amount of traffic on the links the routers drop packets. Since congestion control protocols are meant to tackle loss

Table 2. Simulation Parameters

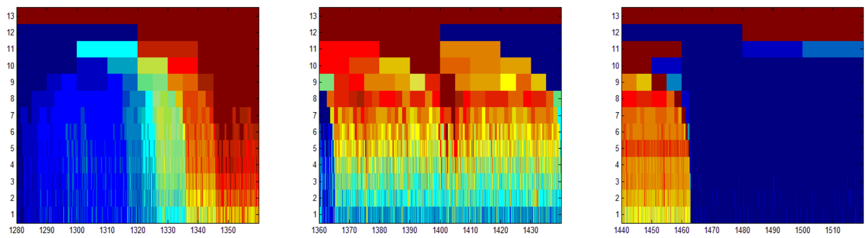
Parameter	*Value*	*Description*
Client load	0-90%	Relative load generated by clients
Attack load	<= 10%	Relative load by attackers
Simulation time	6000s	Simulation run time
Attack time	3100s	Attack duration
Legitimate traffic type	FTP /TCP	
	HTTP /TCP	Application layer protocols
	VoIP /UDP	
Attacker traffic type	UDP	Constant bit rate
Client-router link B/W	5Mb	Link B/W
Attacker-router link B/W	5Mb	Link B/W
Router-router link B/W	10Mb	Link B/W at level-1
Router-router link B/W	100Mb	Link B/W at level-2
Router-server link B/W	10Mb	Link B/W

of packets in order to provide quality of service therefore their services are most affected due to packet loss. As a result the throughput decreases.Topology generation parameters and simulation parameters are listed in tables 1 and 2.

6. Experiments, Results & Discussions

6.1. Pre Test Experiments

6.1.1. Wavelet Selection: The data sets of Lawrence Berkeley Laboratory (lbl-pkt-4) [28] were used to choose an appropriate wavelet for our work. We did extensive tests for 10ms sampling window with Wavelets family of Coiflet, Mexlet, Haar,Symlet and Daubechies. Based on the tests it was realized that Daubechies performs better to other wavelets. For details the reader can refer to [34]. We shortlisted Db6 for our experiments.

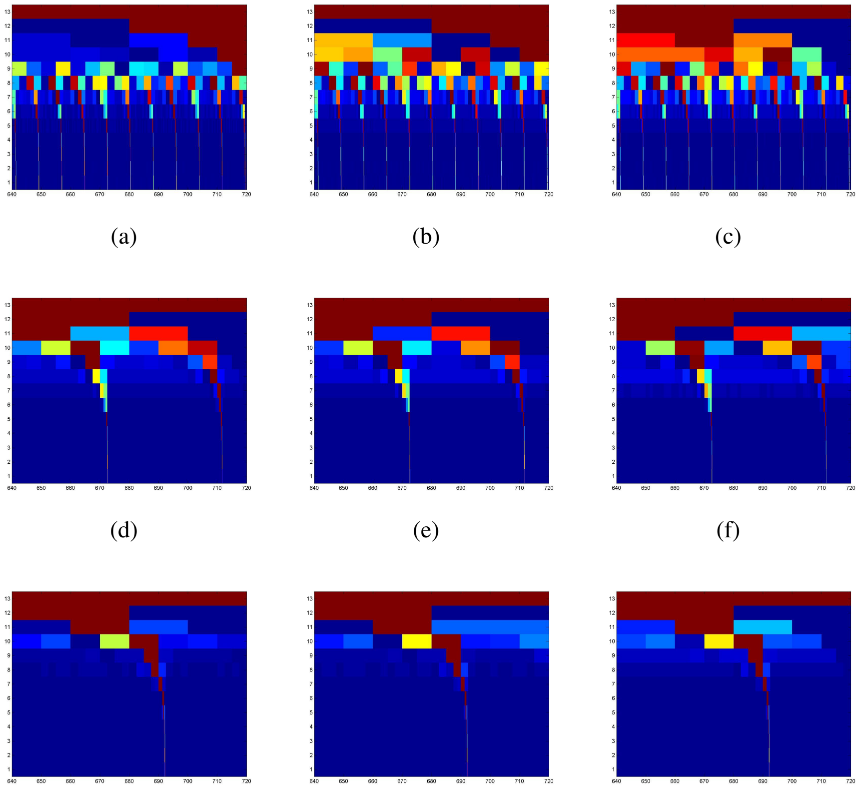


(a) (b) (c)

Figure 5. Flooding Attack

6.2. Hypothesis Testing:

We consider three cases namely flooding attack detection, PDoS attack detection and flash events.



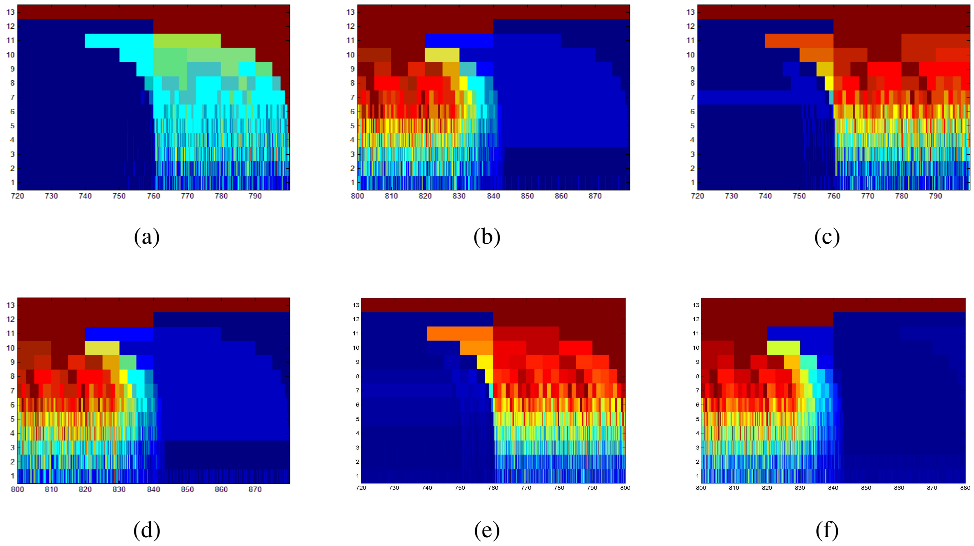
(a) (b) (c)
(d) (e) (f)
(g) (h) (i)

Figure 6. Variants of PDDoS Attacks

6.2.1. Flooding attacks: Figure 5 $(a), (b)$ and (c) show the outputs for time periods 1200s to 1600 s. In our simulation the DDoS attacks start at 1300s . As we can see there is change in color of the bars from blue to lighter blue to lighter green to yellow to orange and then red.The end of flooding attack can also be detected when the color changes to blue again at 1460secs. Flooding attacks are brute force attacks and try to make most benefit of the link bandwidth and therefore the inter-bar distance is very less but from the image the attack can be detected easily.

6.2.2. PDDoS attacks and their detection In order to do a detailed study of the impact of pulsating ddos attacks on legitimated TCP flows we divided them into three categories namely Category A $(DoS5, DoS20)$, Category B $(DoS50)$ and Category C $(DoS100)$ where DoSx is DoS attack of x number of flows.Each category had low$(TCP20)$, medium$(TCP40)$ and high$(TCP80)$ legitimate flows where TCPy is TCP traffic of y number of flows. The pulse lengths(L) were taken as short(50), middle$(100, 150)$ and long(200) ms and time period(T) was grouped as high frequency$(4000ms$ to $12000ms)$,moderate frequency$(16000ms$ to $24000ms)$ and low frequency $(280000ms$ to $40000ms)$.

Because of space constraint it is difficult to show and discuss all the generated scenarios. Figure 6 has thumbnails for following outputs that we received. Figure 6 $(a), (b)$ and (c) are of Category A PDoS, TCP20/40/80 middle length and high frequency; $(d), (e)$ and (f) are of Category B PDoS, TCP20/40/80 middle length moderate frequency and $(g), (h)$ and (i) are of Category C PDoS, TCP20/40/80 middle length low frequency.



(a) (b) (c)

(d) (e) (f)

Figure 7. Flash Event for varied TCP Flows

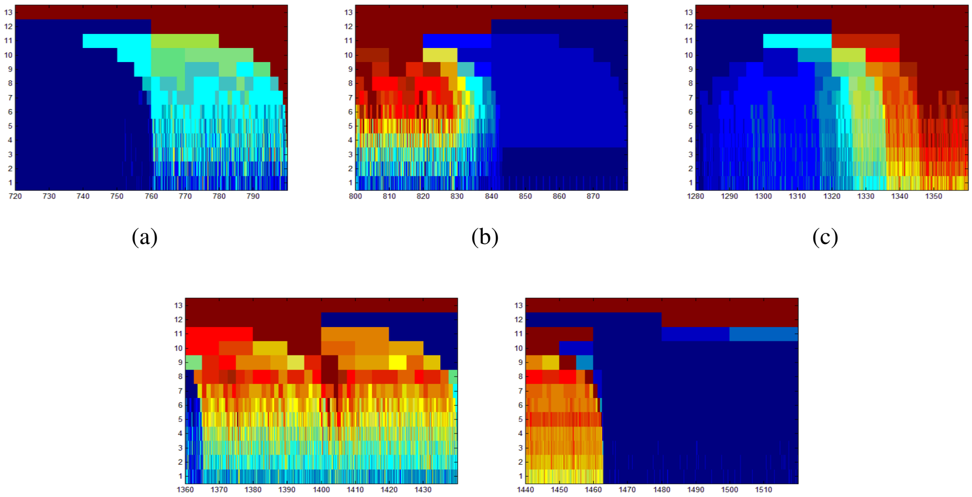
From the figures we can note important facts.In case of Category A PDoS attacks the lower scale values from scale 1-scale 5 portray the attack distinctively. Figures for TCP20 and has bluer rows in the scale range scale 9 - scale 12 whereas figures for TCP40 has more visibility of orange and green which are closer to mean yellow color and figures for TCP80 has a large portion as red and dark red. Therefore for a network link which has low, moderate and high traffic rate the impact of small PDoS attack can be clearly noticed. Figures (d), (e) and (f) are for Category B PDoS which is 50DoS flows and TCP20,TCP40,TCP80 respectively.

One can see hotter colors like red upto scale 10 distinctively. There is not much change in upper scale regions in all the three figures thus showing that DoS50 with large number of DoS flows is quite damaging in all three scenarios of low,moderate and high traffic. Figures (g), (h) and (i) are for Category C PDoS due to 100DoS. All the three figures are similar to each other with very little variation in colors. The results are visible from scale 1 to scale 13 here. There are mostly yellow,red and blue color bars.

6.2.3. Distinction between a flash event and flooding attack: Our main concern was to be able to distinguish a flash event from not only the PDoS type of attack but also from flooding attacks.

We discuss two experiments in this case.

Case 1: Flash Events in the presence of low,moderate and high traffic rate: We referred to the works of [33] for generating flash traffic in network simulator NS2. Three experiments were conducted to test flash traffic namely low traffic of 20 TCP flows, moderate traffic of 40 TCP flows and High traffic of 80 TCP flows. TCP traffic start time was 5s and stop time was 1400s whereas flash traffic start time was 760s and stop time was 770s.We received the Figure 7 (a), (b) for low traffic,(c), (d) for moderate traffic and (e) and (f) for high traffic respectively. When flash event starts the impact can be seen in the figures (a), (b) and (c). It was also noticed that because of different traffic load the bar colors were found to be closer to red in figures (c) and (e) compared to figure (a).



(a) (b) (c)

(d) (e)

Figure 8. Flash Event and Flooding Attack

Case 2: Flash event and Flooding attack: Figures 8 (a), (b) show flash event and (c), (d) and (e) flooding attack respectively. There start patterns are similar but mid way one can see that for flooding attacks there is slowly decaying tail whereas in case of flash event the decay happens in shorter time interval.

7. Validation based on KDD Dataset

7.1. KDD Dataset

Although the KDD CUP dataset [35] has many flaws as stated in [36] yet it is the only dataset that is publicly available and is considered benchmark dataset for testing of intrusion detection algorithms. It has labeled attack samples which are obtained by passive monitoring, rather than by inserting the attack packets into the normal traces.

The datasets consist of two types of data: training and testing. Each record of the training data is labeled as either normal or a specific kind of attack. The training data contain a total of 22 attack types and an additional 15 attack types in the test data only.

The attacks fall in one of the four categories which are Denial of Service (DoS), User to Root (U2R), Remote to local (R2L) and Probe.

- Denial of Service(DoS): An attacker tries to prevent legitimate users from using a service. For example, SYN flood, smurf and teardrop.
- User to Root (U2R): An attackers has local access to the victim machine and tries to gain super-user privilege. For example, buffer overflow attacks.
- Remote to Local (R2L): An attackers tries to gain access to victim machine without having an account on it. For example, password guessing attack.
- Probe: An attacker tries to gain information about the target host. For example, port-scan and ping-sweep.

Table 3. Attacks distribution in KDD Cup Training Dataset

Dataset	*DoS*	*U2R*	*R2L*	*Probe*	*Normal*	*Total*
Corrected KDD	2,29,853	70	16,347	4,166	60,593	3,11,029
10-percent Corrected KDD	3,91,458	52	1,126	4,107	97,278	4,94,021
Whole KDD	38,83,370	41,102	52	1126	9,72,780	48,98,430

The numbers of samples of each category of attack in Corrected KDD and 10-percent KDD training dataset are shown in Table 3.

Reasons for choosing KDD Dataset:

Firstly, it has been observed by William and Martin [37] that the DARPA data exhibits self-similarity within certain interval of time. We choose KDD data set for our detection algorithm because KDD dataset is derived from DARPA data set and is therefore expected to have self similarity present in it.

Secondly, it has been argued by Sabhnani and Serpen [36]that KDD data set cannot be used for misuse-based IDS because of high variation in the attacks distribution in training and testing data values. The coverage of new attacks, in the R2L and U2R categories, in testing data is more than the known attacks. Therefore no misuse-based IDS can show high accuracy in its detection rate.On the other hand the accuracy of anomaly based detection algorithms is not badly affected by the weaknesses of KDD dataset. Therefore, we use KDD dataset for our experiment.

Features Used in the Dataset:

Each record of the dataset represents a connection between two network hosts according to existing network protocol and is described by 41 attributes (38 continuous or discrete numerical attributes and 3 categorical attributes). This set of attributes includes general TCP features like duration, protocol_type, service, src_bytes, dst_bytes, flag, land, wrong_fragment and urgent and derived features like the-same-host features and the-same-service features. A detailed description of these features can be found in [38]. These features are grouped into three sets: basic, content, and time-based traffic features.The basic and content feature groups describe the host audit and log information. The time-based traffic features describe the network connections and traffic information. For example, one of the basic features called src_bytes and dst_bytes describe the number of data bytes from source to destination and number of data bytes from destination to source repectively; the feature land tells about whether the connection is from the same host or not. Detecting different categories of intrusions requires different feature groups. The basic and content features are suitable in the detection of host-based intrusions such as U2R or R2L intrusions, while the time-based traffic features are more suitable in the detection of DoS and probing attacks, which are typical flooding attacks. We have worked on DoS attacks only and considered Smurf, Teardrop, Neptune and POD as attacks for testing our detection algorithm:

Smurf: Smurf attacks in KDD dataset use ICMP echo request packets directed to IP broadcast addresses from remote locations to create DoS attack. It can be identified by watching large number of Echo requests and replies from the victim machine. The coloumn ”count” values from the dataset can be read and analysed for Smurf attacks. In our experimentation we have considered 28,07,886 Smurf attacks in total.

Neptune: To initiate Neptune attack a large number of SYN packets are sent to target machine to exhaust its buffer. Neptune attack never establishes the TCP session resulting in many zero packets in each connection attempt. In our experimentation we have considered 10,72,017 Neptune attacks in altogether. As a result of large number of SYN packets sent by the attacker nodes in case of Neptune attack without actually establishing an actual session there is loss of self similarity and the Hurst values go beyond the required limits. This happens because of same size of incoming bytes for a long duration of time in the traffic. No variation in hurst is used as a signal for attack.

Teardrop: Teardrop uses overlapping of IP fragments. It causes machines to reboot. This attack affects systems that are still using old versions of Windows and Linux operating systems. In our experimentation we have considered 979 Teardrop attacks in total.

Ping-Of-Death (POD): POD affects older Operating Systems. It uses oversized IP packets to crash, freeze or reboot the system.During the experimentation POD affected none of the victim systems. ICMP packets longer than 64000 bytes can be due to POD attack. In our experimentation we have considered 264 POD attacks in total.

We have carried out our simulation work using Daubechies Wavelet (Db6) of MATLAB package. We compute the Hurst parameter H for 13 levels. The computed H values largely fall in the range (0.5,1) for self-similarity thus proving that KDD data values show self similarity though in certain periods of time.

7.2. Results

Since our detection algorithm is a two step process in which we firstly compute Hurst values and then find its point of presence therefore we look at variation in H to signal possibility of an attack and then to find the point of occurence we draw an Image of the block of values in the given window.

We have divided DoS attacks into two categories.First one consists of long duration attacks like Smurf and Neptune.Second category consists of short duration attacks like PoD and Teardrop.We state our observations accordingly.

Long Duration Attacks: In our experiment we use Daubechies Wavelet to find out variations in self similarity of the traffic. Our block size is 8192 which is equivalent to 2^{13}. Long duration attacks are the ones that span to more than one block of data. Both Smurf and Neptune come under this category. We have observed that under normal conditions there is very small variation in the H values but in the presence of these attacks there is considerable change. Few points worth noting are:

Firstly, In case of Smurf attacks the change in self similarity is in the range of 0.1 to 0.3 where as for Neptune it is 0.1 to 0.25. In general for both the attacks the change in self similarity is followed by no change at all for long interval of time. We think that the reason for this is that there is no change in the size of packets and hence the values remain constant. And because we measure variation in the traffic therefore absence of variation results in out of range results. Secondly, absence of incoming traffic followed by Smurf attack gives H values above threshold and therefore we look out for second parameter i.e average packet size in order to decide the possible type of DoS attack.

Short Duration Attacks: There are two important observations in this case. Firstly, for PoD attacks the segment size is considered to be 65,535 bytes and the duration of these segments is 19 on average. The self similarity in the dataset is in the range of 0.8 to 1 most of the times. Whenever we have an attack like PoD the self similarity drops below 0.8 to the range of 0.7. Secondly,if the number of PoD segments is more then the average of 19 in a single block the self similarity drops further to the range of 0.5. Thirdly, in case there are more than one instances of PoD attacks in a sinlgle block then there is loss of self similarity and the values fall below 0.5.

8. Conclusion

The work presented in this paper proposes a two stage wavelet based statistical detection algorithm to calculate self similarity for the detection and visualization of anomalous traffic in the computer network traffic. The computer network traffic under normal circumstances exhibits long range dependence and when under attack it deviates from its behaviour. Along with the detection of an attack its point of presence is also identified. The detection algorithm was tested on simulated patterns of flooding attacks, pulsating denial of service attacks and flash crowds under low, moderate and heavy traffic.The tested scheme was then validated with KDD based publicly available dataset. Although the work presented in this paper gives a strong footing to the use of wavelets in the detection and visualization of network based attacks, yet more extensive work can be done by applying the technique to wireless network traffic, distributed network traffic like cloud applications as well as emerging denial of service attacks.

References

[1] CERT, " Overview of Dos and DDoS attacks", http://www.us-cert.gov/cas/tips/ST04-015.html

[2] W. Lu and A. A. Ghorbani (Eds.), "Network Anomaly Detection Based on Wavelet Analysis", EURASIP Journal on Advances in Signal Processing, pp. 1-17, Vol. 2009, (2009)

[3] V. Alarcon-Aquino and J.A. Barria (Eds.), "Anomaly Detection in Communication networks using wavelets", in IEE Proceedings-Communications,pp 355-362, (2001)

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

[5] C.M. Cheng, H.T. Kung and K.S. Tan (Eds.), "Use of spectral analysis in defense against DoS attacks", in IEEE GLOBECOM proceedings, pp. 2143-2148, (2002)

[6] K. Limthong, F. Kensuke and P. Watanapongse (Eds.), "Wavelet-Based Unwanted Traffic Time Series Analysis", in IEEE International conference on computer and electrical engineering, pp. 445-449, (2008)

[7] A. Hussain, J. Heidemann and C. Papadopoulos (Eds.), "Identification of repeated denial of service attacks", in Proceedings of the IEEE Infocom, pp. 1-15, (2006)

[8] A. Magnaghi, T. Hamada and T. Katsuyama (Eds.), "A Wavelet-Based Framework for Proactive Detection of Network Misconfigurations ", in Proceedings of ACM workshop on Network Troubleshooting, (2004)

[9] X. He,C. Papadopoulos, J. Heidemann, U. Mitra and U. Riaz (Eds.)," Remote detection of bottleneck links using spectral and statistical methods", in ACM International Journal of Computer and Telecommunications Networking, pp. 279298, (2009)

[10] G. Carl, R.R. Brooks and S. Rai (Eds.)," Wavelet based denial-of-service detection", in ELSEVIER journal on Computers & Security, vol. 25, pp. 600615, (2006)

[11] M. Hamdi and N. Boudriga (Eds.), "Detecting denial-of service attacks using the wavelet transform", in ELSEVIER Computer Communications, vol. 30, pp. 32033213, (2007)

[12] R. Xunyi, W. Ruchuan and W. Haiyan (Eds.), "Wavelet analysis method for detection of DDoS attack on the basis of self-similarity", in Frontiers of Electrical and Electronic Engineering in China, vol. 2, no. 1, pp. 7377, (2007)

[13] W. Lu, M. Tavallaee and A. A. Ghorbani (Eds.), ”Detecting network anomalies using different wavelet basis functions”, in Proceedings of the Communication Networks and Services Research Conference, pp. 149156, (2008)

[14] G. Bartlett,M.D. Rey, J. Heidemann and C. Papadopoulos (Eds.), ”Using Low-Rate Flow Periodicities for Anomaly Detection”, Extended Technical Report ISI-TR-661, (2009)

[15] W. Leland, M. Taqqu, W. Willinger and D. Wilson (Eds.),” On the self-similar nature of Ethernet traffic” in Proceedings of ACM SIGCOMM, pp. 183193, (1993)

[16] L. Li and G. Lee (Eds.),” DDoS attack detection and wavelets”, in 12th International Conference on Computer Communications and Networks, pp. 421-427, (2003)

[17] J.C.R Pacheco and D.T. Roman (Eds.),” Distinguishing fractal noises and motions using Tsallis Wavelet entropies”, in 2010 IEEE Latin/American Conference on Communications, pp. 1-5, (2010)

[18] S. Abe and N. Suzuki (Eds.),”Itineration of the Internet over Non-equilibrium Stationary States in Tsallis Statistics”, in Physical Review E, Vol. 67, (2003)

[19] A. Dainotti, A. Pescape and G. Ventre (Eds.), ”Wavelet-based Detection of DoS Attacks”, in IEEE conference on Global Communications, pp. 1-6,(2006)

[20] D. G. Perez, L. Zunino, M. Garavaglia and O.A. Rosso (Eds.), ”Wavelet entropy and fractional Brownian motion time series”, Physica A, vol 365(2), pp. 282-288, (2006)

[21] Karmeshu and S. Sharma (Eds.), ”Power Law and Tsallis Entropy: Network Traffic and Applications”, in Chaos, Nonliniearity and Complexity (Studies in Fuzziness and Soft Computing) Springer Verlag, vol. 206, pp. 162, (2006)

[22] P. Abry and D. Veitch (Eds.),”Wavelet analysis of long-range dependent trafc”, in IEEE Transactions on Information Theory, vol. 44, pp. 1111-1124, (1998)

[23] S. Stoev,M.S. Taqqu,C. Park and J.S. Marron (Eds.),” On the Wavelet Spectrum Diagnostic for Hurst Parameter Estimation in the analysis of Internet Trafc”, in ACM Journal on Computer Networks, vol 48, pp. 423-445, (2005)

[24] P. Abry,D. Veitch and P. Flandrin (Eds.),” Long-Range Dependence: Revisiting Aggregation with Wavelets”, in Journal of Time Series analysis, vol 19, Issue 3, pp. 253-266, (1998)

[25] B. Tellenbach, M. Burkhart,D. Sornette and T. Maillart (Eds.),” Beyond Shannon: Characterizing Internet Traffic with Generalized Entropy Metrics”, pp. 239248. Springer, Berlin ,(2009)

[26] J. Mirkovic, A. Hussain, S. Fahmy, P. Reiher and R. Thomas (Eds.),”Accurately Measuring Denial of Service in Simulation and Testbed Experiments”, in IEEE Transactions on Dependable & Secure Computing, vol 6, no 2, pp 81-95,(2009)

[27] D. Moore,C. Shannon and J. Brown (Eds.),”Code-Red: A case study on the spread and victims of an Internet worm”, in Proceedings of Internet Measurement Workshop, (2002)

[28] The Internet Traffic Archives http://ita.ee.lbl.gov/html/traces.html

[29] C. Labovitz, S.I. Johnson, D. McPherson, J. Oberheide and F. Jahanian (Eds.),” Internet interdomain traffic”, in Proceedings of ACM SIGCOMM, vol. 40, pp. 75-86, (2010)

[30] T. Peng,C. Leckie and K. Ramamohanrao (Eds.),” Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems”, in ACM Computing Surveys, Vol. 39, No. 1, (2007)

[31] M. Roghan,D. Veitch and P. Abry (Eds.),”Real-time estimation of the parameters of long-range dependence”, in IEEE/ACM Transactions on Networking, pp. 467478,(2000)

[32] A. Kuzmanovic and E. Knightly (Eds.),”Low-Rate TCP-Targeted Denial of Service (The Shrew vs. the Mice and Elephants)”, in ACM SIGCOMM Proceedings, pp. 75-86, (2003)

[33] I. Ari,A. Hong,E.L. Miller,S.A. Brandt and D.D.E. Long (Eds.),”Managing Flash Crowds on the Internet”, in 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 246-249, (2003)

[34] G. Kaur,V. Saxena and J.P. Gupta (Eds.),”DDoS Detection with Daubechies”, in Springer Proceedings of 5th International Conference on Contemporary Computing,IC3 2012,(2012)

[35] The KDD Archive. KDD99 cup dataset, 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[36] M. Sabhnani and G. Serpen,”Why Machine Learning Algorithm for Detecting Misuse in KDD Intrusion Detection Dataset”, in Journal of Intelligent Data Analysis, vol 8,issue 4, (2004)

[37] W.H. Allen and G.A. Marin (Eds.),” On the Self-Similarity of Synthetic Traffic for the Evaluation of intrusion Detection Systems”, in Proceedings of the IEEE/IPSJ International Symposium on Applications and the Internet (SAINT), pp. 242-248,(2003)

[38] W. Lee and J. Stolfo Salvatore (Eds.), ”Data Mining Approaches for Intrusion Detection”, in Proceedings of the 7th USENIX Security Symposium (SECURITY-98), pp. 79-94, (1998)

