8-2005

NABOH system: Gathering intelligence from traffic patterns

Angelica M. Delgado
University of Texas-Pan American

NABOH SYSTEM:

GATHERING INTELLIGENCE FROM TRAFFIC PATTERNS

A Thesis

by

ANGELICA M. DELGADO

Submitted to the Graduate School
of the University of Texas-Pan American
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2005

Major Subject: Computer Science

NABOH SYSTEM:

GATHERING INTELLIGENCE FROM TRAFFIC PATTERNS

A Thesis
by
ANGELICA M. DELGADO

Approved as to style and content by:

Richard Fowler, Ph. D.
Chair of Committee

Zhixiang Chen, Ph. D.
Committee Member

Wendy A Lawrence Fowler

Wendy A. Lawrence-Fowler, Ph. D.
Committee Member

August 2005

ABSTRACT

Delgado, Angelica M., NABOH System: Gathering Intelligence from Traffic Patterns. Master of Science (MS), August, 2005, 50 pp., 15 Figures, references, 37 titles.

Network traffic anomalies are important indicators of problematic traffic over a network. Network activity has patterns associated with it depending on the applications running on the local hosts connected to the network. There are traffic parameters into which network traffic of a local host can be divided: bandwidth usage, number of remote hosts that a local host is connecting to and vice versa, and number of ports used by the local host. This thesis develops a system for detecting and profiling network anomalies by analyzing traffic parameters using intelligent computational techniques. The developed system gathers intelligence by examining only the headers of IP packets. Thus the system is referred to as NABOH (Network Anomalies Based On Headers).

DEDICATION

I would like to dedicate my thesis to my mother, Irene Escobar, who has been a constant source of encouragement and unrelenting support.

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge the supervision of Dr. Fitratullah Khan, Professor of the Computer Science Department and Director of Infrastructure, Telecommunications, and Networks Department (ITNet) of the University of Texas at Brownsville and Texas Southmost College, during this thesis. I want to thank my colleagues who gave useful suggestions during the course of research. This thesis would not have been possible without the research environment and tools made available at ITNet.

TABLE OF CONTENTS

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

Network analysis and security have never been more important than present time. The growth of new technologies has made Internet the best communication tool now days. However, new technologies have brought new threats with need for new solutions. A single intrusion of a computer network can result in loss of connectivity for the whole network bringing down productivity of a business. Just in 2003, there was a large increase in network attacks. One of these attacks started early in 2003. This was the well-known worm called W32/SQL Slammer. It attacked a known vulnerability in Microsoft SQL Server 2000 Web servers and slowed down Internet traffic worldwide. The worm used a buffer overflow to take over a server to send out a flood of packets causing a similar effect as experienced in a denial of service attack. The most affected country was South Korea, where most of the nation's fixed-line and mobile Internet users were unable to access Web sites for nearly half a day [3, 4, 5]. The graph in Figure 1 shows a single infected machine on a 100Mbps Ethernet link. It shows how the traffic jumped close to 100% as it got infected. The outgoing flood by one host prevented packets of other hosts from going over the network, thereby, virtually cutting off other machines on the network [6].

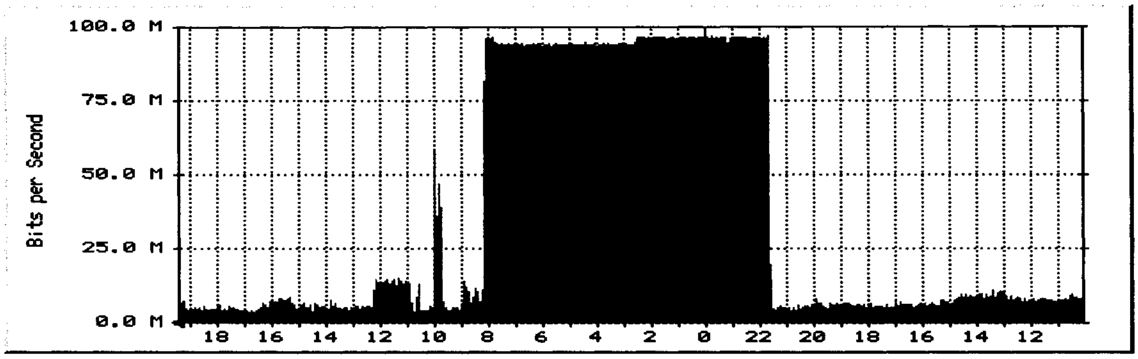
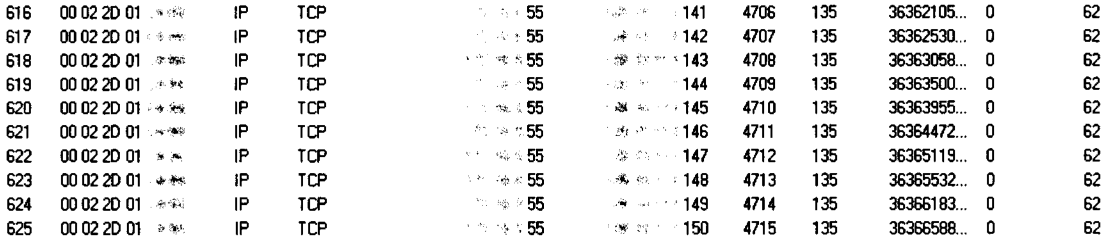


Figure 1 Graph of W32/SQL Slammer

Another tremendous network attack occurred in August 2003. This attack was caused by Distributed Component Object Model (DCOM) Remote Procedure Call (RPC) vulnerability in some Windows Operating Systems. The attacks were carried by W32.Blaster.Worm which was infecting 30,000 systems per hour [7]. Other similar worms appeared at the same time such as Welchia and Nachi, bringing down businesses for a couple of days [3, 4]. The damage caused by a worm can be huge. For example, W32.Blaster.Worm was thought to be the cause of the massive power outage that struck the Eastern United States and Canada on August 14, 2003, leaving sections of New York City, Detroit, Cleveland and Toronto without electricity [8, 9]. The main characteristic of W32.Blaster.Worm was the scanning on tcp port 135 as shown in Figure 2 [10]:



```
616  00 02 2D 01   IP   TCP        55       141   4706   135   36362105...  0     62
617  00 02 2D 01   IP   TCP        55       142   4707   135   36362530...  0     62
618  00 02 2D 01   IP   TCP        55       143   4708   135   36363058...  0     62
619  00 02 2D 01   IP   TCP        55       144   4709   135   36363500...  0     62
620  00 02 2D 01   IP   TCP        55       145   4710   135   36363955...  0     62
621  00 02 2D 01   IP   TCP        55       146   4711   135   36364472...  0     62
622  00 02 2D 01   IP   TCP        55       147   4712   135   36365119...  0     62
623  00 02 2D 01   IP   TCP        55       148   4713   135   36365532...  0     62
624  00 02 2D 01   IP   TCP        55       149   4714   135   36366183...  0     62
625  00 02 2D 01   IP   TCP        55       150   4715   135   36366588...  0     62
```

Figure 2 Trace of W32.Blaster.Worm

Not only worms and viruses are concerns for network administrators, but also the popularity of P2P (Peer-to-Peer) applications within its Internet users has become another threat for the network bandwidth. P2P networking applies to individual computers serving as clients and servers to other peer computers. These P2P applications are famous because they allow the users to share files including music, movies, etc. Currently, the famous P2P file sharing programs are Gnutella, iMesh, Kazaa, Morpheus, and Ares/Warez just to mention a few. The problem with P2P traffic is that it is unattended and always on. Normally, the user is not in front of a computer while files are being uploaded or downloaded. This aspect of P2P traffic is the one that causes network congestion, consequently, slowing down the network for the rest of the users. Unfortunately, P2P applications use the process called "Port Hopping," which randomly defines how P2P traffic will appear as it travels over the network. Currently, it mostly appears as being web browsing traffic. This is why it is very difficult for network administrators to control or block this type of traffic.

TYPES OF INTRUSION DETECTION SYSTEMS

Some of the currently available network security solutions to prevent and/or to make aware of problematic activity on the network are based on Intrusion Detection System (IDS) concept. An intrusion detection system can use misuse detection model, anomaly detection model, or both to detect an intrusion. The anomaly detection model detects intrusions by looking for activity that is different from a user's or systems normal behavior. On the other hand, misuse detection model detects intrusions by looking for activity that corresponds to known intrusion techniques (signatures) or system vulnerabilities. There are three types of intrusion detection systems [1, 2, 16]:

- HIDS (Host based Intrusion Detection System) – The audit data from a single host is used to detect intrusions.

- Network based IDS (Network Intrusion Detection System) – The network traffic data, along with audit data from one or more hosts, is used to detect intrusions.
- Vulnerability-Assessment IDS- It detects vulnerabilities on internal networks and firewalls.

The three types of intrusion detection systems are important in network security. However, this thesis focuses on network based intrusion detection systems. This type of IDS focuses on all the traffic generated by all the network users, not only by a single host.

In the market, there are several network based intrusion detection systems. For example, some commercial network based intrusion detection systems are Real Secure from Internet Security Scanner (ISS), Cisco Secure IDS, NetRanger from Cisco, Centrax from CyberSafe Corporation and Network Flight Recorder (NFR). One famous free network based intrusion detection system is Snort which is an “open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks” [15]. Most Network based IDS are for misuse detection which is based on known signatures that need to be updated regularly.

CURRENT METHODS USED FOR MISUSE AND ANOMALY DETECTION

There are different methods in place for misuse and anomaly models of intrusion detection systems. Some methods for misuse detection model are [16, 22]:

- Signature Analysis- It translates attack scenarios into sequences of audit events.

- State-Transition Analysis- It describes attacks with a set of goals and transitions based on state-transition diagrams.

Misuse detection model, as documented earlier, searches for known patterns or signatures of attacks. However, a disadvantage of this model is that it would only be able to detect intrusions that follow predefined patterns. There are some network based intrusion detection systems that use signature analysis. One of these systems is RealSecure (1999) [22]. RealSecure was developed by Internet Security Systems. This network based intrusion detection system is composed of three modules. These three modules are the network engines, the system agents, and the managers. It monitors the content of network packets to look for signatures which could indicate an attack on the network. Another system using signature analysis is NetRanger (1999). This system is an "enterprise-scale, real-time, intrusion detection system designed to detect, report, and terminate unauthorized activity throughout the network" [23]. It is composed of two components. One component is made up of sensors. The sensors are "high-speed network appliances" that analyze the content and the context of individual packets to determine if it is legitimate traffic. If it is threatening traffic, it sends it to the second component which is the director. The director is responsible for monitoring and managing the sensors. It alerts the network administrator if there is an alert on the network. Figure 4 shows a NetRanger setup on the network [27,28]:

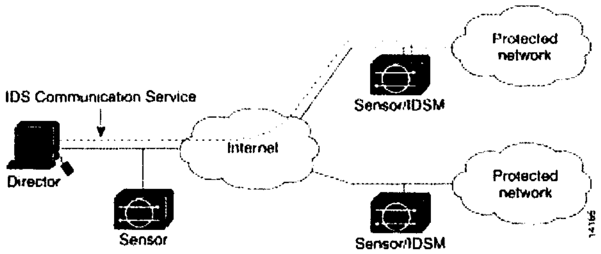


Figure 3 Diagram of NetRanger Setup

Also, there are systems using state-transition analysis for misuse detection. This method was first introduced in State-Transition Analysis Technique (STAT) [26]. It describes an attack as "a sequence of actions which progressively takes a computer from an initial normal state to a compromised state" [36]. This technique was first developed for the host-based intrusion detection system, called USTAT, but in 1999 it was extended to network traffic analysis. This intrusion detection system based on STAT was NetStat which was developed at the University of California at Santa Barbara. It performs real-time network based intrusion detection system by extending the "state transition analysis technique" to the network environment. It applies this technique by modeling both the guarded network and the attacks to determine which network events have to be monitored [36].

Also, there are methods utilized for anomaly detection model. As documented earlier, anomaly detection detects intrusions by searching for abnormal network traffic. These are some methods used for anomaly detection model [16, 22]:

- Statistical measures- It learns from historical events.
- Data mining- It analyzes the data using sophisticated search tools to look for trends or anomalies without knowledge of the meaning of the data.

One method using statistical measures for anomaly detection is Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD). Emerald was developed at SRI International, Menlo Park, CA in 1997 [25]. It helps detect intrusions in large networks by focusing on the scalability of the system. It provides high-volume event analysis and easy customization for new targets and specific policies [29]. Another work using statistical methods for anomaly detection is Cabrera et al. (2000). This intrusion detection system "examines the application of statistical traffic modeling for detecting novel attack against networks" [22]. This system tries to demonstrate that network activity models efficiently detect attacks by monitoring network traffic volume. A method using data mining technique for anomaly detection is Audit Data Analysis and Mining (ADAM) [22]. This system was developed at George Mason University Center for Secure Information Systems. It uses "a combination of association rules mining and classification to discover attacks in a TCPdump audit trail" [30]. The system works based on building a repository of normal frequent item sets that were collected during periods of no attacks [22, 30]. Then, ADAM uses a sliding window algorithm to find frequent item sets in the current set of TCP connections and compares them with those stored in the normal item set repository. It discards the item sets which are considered normal and classifies rest of them. The classifier that ADAM uses is previously trained to determine if the item set is a known type of attack, an unknown type, or a false alarm. The disadvantage of this system is the use of random sampling with the risk of missing some anomalies on the network.

CURRENT RESEARCH PROJECTS USING INTELLIGENT TECHNIQUES

Attacks are incrementing with time and the worst thing is that hackers are creating more sophisticated worms and viruses which make them more difficult for intrusion detection systems to detect in time. With the use of intelligent techniques, intrusion detection systems can be capable of detecting unknown threats faster than the systems not employing these techniques. Some of intelligent techniques that could be used for network based intrusion detection systems are expert systems, neural networks, and fuzzy logic.

Expert systems technique is one of the intelligent techniques used by network based intrusion detection systems. The technique consists of a set of rules that encode "the knowledge of a human expert" [14]. In intrusion detection systems, an expert system contains a set of rules that describes attacks. It permits the incorporation of an extensive amount of human experience into a computer application which then uses that knowledge to identify suspicious events that match the defined characteristics of misuse [14]. One expert system is Network Intrusion Detection Expert System (NIDX) created by Bauer and Kblentz in 1988 at Bell Communication Research. NIDX is an approach for misuse detection model. It "combines knowledge of the target system, history profiles of users' past activities, and intrusion detection heuristics" to create "a knowledge-based system capable of detecting specific violations that occur on the target system" [22]. The disadvantage of expert systems technique is the requirement of frequent updates to remain current.

Neural Network is another intelligent technique used in intrusion detection systems. It attempts to imitate the way human brain works. There are two potential implementations in neural networks for misuse detection. One of these approaches is to incorporate neural network into an existing expert system [15]. This approach involves using neural network to filter the incoming data for suspicious events and forward these events to an expert system. The second approach would involve in having neural network as a standalone misuse detection system. In this approach, the neural network will receive data from the network and it will analyze the data for instances of misuse. One disadvantage of using neural network in misuse detection is the training requirements which imply an accurate training of the system [14]. The training methods and training data are critical in order for neural network to function at its best.

Fuzzy Logic is yet another intelligent technique used for intrusion detection systems. It is a "means of specifying how well an object satisfies a vague description." [19]. The difference between normal and abnormal activities on the network is not distinct but rather fuzzy or uncertain [31]. In a fuzzy set, an object can partially be in a set reflected by the degree of membership.

One approach using fuzzy logic for network based intrusion detection systems is Fuzzy Intrusion Recognition Engine (FIRE) [20]. This system is based on anomaly detection model. It uses fuzzy logic and simple data mining techniques to identify malicious network activity. The disadvantage of this system is that web traffic is ignored at the monitoring stage.

CHAPTER II

NABOH: SYSTEM FOR DETECTING AND PROFILING NETWORK TRAFFIC ANOMALIES

The available intrusion detection systems gather intelligence to detect intrusion. They focus on what specific ports the local host and remote host use to consider a session as an intrusion. Most of the algorithms are bulky and intrusive because they depend on the contents of a packet rather than its traffic parameters. Also, these systems are limited to detecting intrusion rather than network anomalies which encompass intrusion and other network compromising traffic. This thesis takes a different approach. It develops a light-weight system to gather intelligence based on basic traffic parameters that are available from the headers of the packets only. Thus the system is referred to as NABOH (Network Anomalies Based On Headers). This approach makes it possible for (a) faster processing by looking at headers only, (b) preserving privacy by not looking at the contents of packets, and (c) comprehensively considering different traffic parameters to detect and profile network traffic anomalies.

An extremely important consideration in selecting the above approach is that the needed intelligence is gathered irrespective of data being encrypted. Since more encryption is utilized now days, this is an important consideration in designing a system for intelligence gathering for use in network security. The developed system only looks at the headers, therefore it works equally well for encrypted traffic. Another salient feature of the system is that it does not depend on specific ports. For example, FIRE uses fuzzy sets based on a composite key of source IP address, destination IP address, and destination port to determine anomalies in the network traffic. Its disadvantage is that it excludes tcp port 80/http in the monitoring stage [20]. This can cause the system to not detect some violations done using tcp port 80/http. However, if one can create a system capable of analyzing misuses and anomalies of the network without predefined port list, there would be fewer attacks. This is accomplished by collecting the following traffic parameters for each local host:

i) Bandwidth in bits per second (bps)

-incoming bps

-outgoing bps

ii) Number of hosts

-number of remote hosts that the local host is connecting to

-number of remote hosts connecting to the local host

iii) Number of ports

-number of source ports used by the local host

-number of destination ports used by the local host

Network traffic is examined over a predetermined period. Several time period windows sizes had been tested to determine which window size provides more detail of the traffic parameters. Some of these windows gave too little information such as the 15-minute period window or too much information such as the 1-hour period window. A 30-

minute period has been empirically found to render reasonable detail [Appendix C]. This is a sliding window which is examined every M minutes. Depending on the number of local hosts to be examined, M can be selected to be any where from one minute to several minutes for a network spanning over several subnets. For example, for an organization with 3000 nodes, the optimal value of M has been determined to be 5 [Appendix C]. That is to say, a 30-minute traffic sample is examined every 5 minutes for every host on the network.

There are six types of traffic samples for every host; two types for every one of the three traffic parameters listed above. Every pattern is normalized for pattern matching purposes. Each pattern is represented by 30 normalized values or points. As an example, a local host can have the following six network traffic patterns:

Incoming bps:
(0.4 0.5 1.0 0.7 0.9 0.8 0.7 0.9 0.5 0.1 0.8 0.3 0.5 0.4 0.9 0.7 0.1 0.9 0.3 0.8 0.8 0.9 0.3 0.2 0.5 0.8 0.9 0.3 0.1 0.1)

Outgoing bps:
(0.1 0.1 0.1 0.6 0.1 0.5 0.4 0.6 0.8 0.8 0.8 0.8 1.0 0.7 0.7 0.7 0.4 0.8 0.1 0.3 0.5 0.2 0.3 0.1 0.6 0.7 0.8 0.9 0.7 0.8)

Number of remote hosts being contacted by a local host:
(0.9 1.0 0.3 0.3 0.4 0.5 0.6 0.7 0.2 0.1 0.3 0.6 0.4 0.5 0.4 0.4 0.4 0.5 0.6 0.7 0.9 0.9 0.1 0.1 0.4 0.6 0.4 0.2 0.9 0.9)

Number of remote hosts contacting a local host:
(0.1 0.1 0.1 0.6 0.1 0.5 0.4 0.6 0.8 1.0 0.8 0.3 0.5 0.4 0.9 0.7 0.1 0.9 0.3 0.8 0.8 0.9 0.3 0.3 0.3 0.8 0.9 0.3 0.9 0.9)

Number of source ports used by a local host:
(0.1 0.2 1.0 0.3 0.3 0.3 0.2 0.2 0.2 0.2 0.4 0.2 0.3 0.5 0.4 0.3 0.2 0.2 0.1 0.1 0.1 0.1 0.2 0.3 0.3 0.3 0.3 0.3 0.3 0.3)

Number of destination ports used by a local host:
(0.6 0.6 0.0 0.7 0.9 1.0 0.7 0.8 0.5 0.1 0.9 0.3 0.5 0.4 0.2 0.3 0.1 0.3 0.2 0.3 0.4 0.5 0.6 0.9 0.8 0.8 0.9 0.8 0.5 0.4)

Using these 30-minute sliding window patterns, NABOH system detects anomalies by applying three different profiling methods on each of the above six different traffic patterns:

a) Profiling based on Fuzzy Sets: Compare the traffic patterns with repertoire of known patterns.

b) Profiling based on Traffic Shape: Determine shape using Method of Least Squares. For example, determine if the traffic shape has exponential growth or a constant unrelenting plateau.

c) Profiling based on Fluctuations: Determine the degree of severity of change.

Note that the above three profiling methods have to be applied to all the six types of network traffic patterns to detect different types of anomalies. Hence, NABOH system comprehensively evaluates eighteen different profiles for each local host. This is important to do because not all attacks or anomalies consume a large amount of bandwidth. Some merely consume a few kilobits per second but the number of hosts involved is large. In other cases, use of a large number of ports is a sign of threatening activity. As an example, Figure 6 shows a 30-minute activity of a host infected with Nachi worm. In this case, the bandwidth usage of the infected host was no more than 50kbps. Audit logs regarding Nachi worm indicated that the compromised host tried to established connection to more than 200 hosts per minute. The number of remote hosts is incremented by 70% in a couple of seconds as soon as the worm gets activated. Hence, by putting an alert system on the number of remote hosts per local host, these types of attacks are detected and profiled.

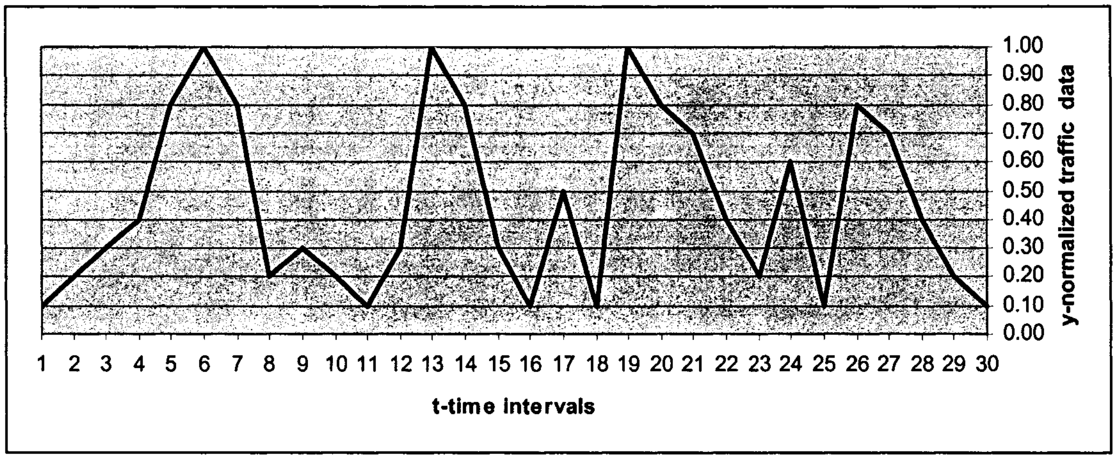


Figure 4 Trace of Nachi worm

NABOH system is implemented by organizing the implementation into three parts: (a) collection of data sets, (b) analysis of data, and (c) generation of alerts:

- Collection of data sets

  -IPtraf is used to collect headers every minute.

  -Traffic parameters are collected for every host.

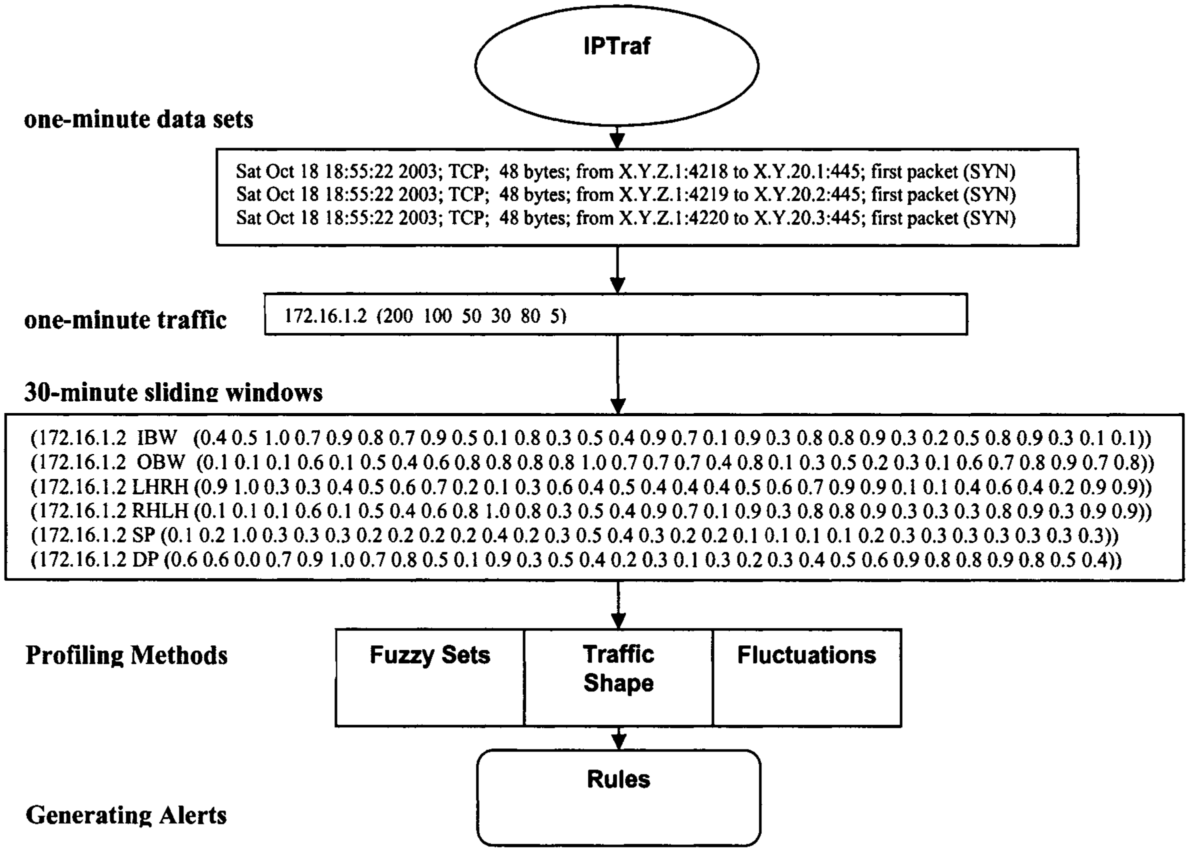
- Analysis of data

  -The collected parameters are entered into the sliding window for every host.

  -The sliding windows are analyzed and profiled for every host using the three profiling methods described earlier.

- Generation of alerts

  -Alerts are generated according to the defined rules.



```
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4218 to X.Y.20.1:445; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4219 to X.Y.20.2:445; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4220 to X.Y.20.3:445; first packet (SYN)
```

```
172.16.1.2 (200 100 50 30 80 5)
```

```
(172.16.1.2 IBW  (0.4 0.5 1.0 0.7 0.9 0.8 0.7 0.9 0.5 0.1 0.8 0.3 0.5 0.4 0.9 0.7 0.1 0.9 0.3 0.8 0.8 0.9 0.3 0.2 0.5 0.8 0.9 0.3 0.1 0.1))
(172.16.1.2 OBW  (0.1 0.1 0.1 0.6 0.1 0.5 0.4 0.6 0.8 0.8 0.8 0.8 1.0 0.7 0.7 0.7 0.4 0.8 0.1 0.3 0.5 0.2 0.3 0.1 0.6 0.7 0.8 0.9 0.7 0.8))
(172.16.1.2 LHRH (0.9 1.0 0.3 0.3 0.4 0.5 0.6 0.7 0.2 0.1 0.3 0.6 0.4 0.5 0.4 0.4 0.4 0.5 0.6 0.7 0.9 0.9 0.1 0.1 0.4 0.6 0.4 0.2 0.9 0.9))
(172.16.1.2 RHLH (0.1 0.1 0.1 0.6 0.1 0.5 0.4 0.6 0.8 1.0 0.8 0.3 0.5 0.4 0.9 0.7 0.1 0.9 0.3 0.8 0.8 0.9 0.3 0.3 0.3 0.8 0.9 0.3 0.9 0.9))
(172.16.1.2 SP (0.1 0.2 1.0 0.3 0.3 0.3 0.2 0.2 0.2 0.2 0.4 0.2 0.3 0.5 0.4 0.3 0.2 0.2 0.1 0.1 0.1 0.1 0.2 0.3 0.3 0.3 0.3 0.3 0.3 0.3))
(172.16.1.2 DP (0.6 0.6 0.0 0.7 0.9 1.0 0.7 0.8 0.5 0.1 0.9 0.3 0.5 0.4 0.2 0.3 0.1 0.3 0.2 0.3 0.4 0.5 0.6 0.9 0.8 0.8 0.9 0.8 0.5 0.4))
```

Figure 5 Diagram of NABOH System

NABOH uses concepts from different computational areas, from Fuzzy sets to statistics, in order to detect and profile threatening patterns. The following outline explains what is involved in implementing NABOH system:

1) Data sets: The information of IP traffic passing over the network constitutes the data sets.

2) Traffic Parameters: The aforementioned data sets are used to determine hosts' behavior in terms of three traffic parameters: (a) bandwidth usage, (b) number of remote hosts that a local host is connecting to and vice versa, and (c) number of ports used by the local host.

3) Sliding window: There are two 30-minute sliding windows for each traffic parameter of each host consisting of normalized traffic data sets.

4) Proposed Methods for Intrusion Detection: The following three methods are used to determine an anomaly on the network based on the three traffic parameters mentioned part 2 above:

    (a) Profiling based on Fuzzy Sets

    (b) Profiling based on Traffic Shape

    (c) Profiling based on Fluctuations

5) Alert System: This system is created for the above proposed intrusion detection system in order to flag an anomaly on the network.

DATA SETS

The information of IP traffic passing over the network constitutes the data sets. These are the packet and byte counts, protocol type, source IP address, destination IP address, source port, destination port, and additional detail depending on Protocol type. The application utilized for gathering of data sets is IPTraf package. This package is briefly described in the next section.

DATA SET COLLECTOR: IPTraf

This is a network sniffing utility for IP networks. It sniffs packets on the network and provides various pieces of information about the current IP traffic. The fields collected from a packet are the protocol type, source IP address, destination IP address, source port, destination port, and packet size. The following is a data sample of IPTraf logging:

```
Mon May 17 13:02:31 2004; TCP; eth0; from X.Y.Z.1:1340 to X.Y.Z.1:445: new source MAC address 0004760d9e3c
(previously          000d29f31480)
Mon May 17 13:02:31 2004; TCP; eth0; from X.Y.Z.1:1340 to X.Y.Z.3:445: new source MAC address 000d29f31480
(previously          0004760d9e3c)
Mon May 17 13:02:31 2004; TCP; eth0; 46 bytes; from X.Y.Z.2:25 to X.Y.Z.4:38868 (source MAC addr 00065b39494f);
FIN sent;  2419 packets, 115607 bytes, avg flow rate 0.00 kbits/s
Mon May 17 13:02:31 2004; UDP; eth0; 402 bytes; source MAC address 000e392c8800; from X.Y.Z.136:64538 to
X.Y.Z.1:33781
Mon May 17 13:02:31 2004; UDP; eth0; 159 bytes; source MAC address 000d29f31480; from X.Y.Z.12:53 to
X.Y.Z.4:53
Mon May 17 13:02:31 2004; UDP; eth0; 143 bytes; source MAC address 000bdb08ad72; from X.Y.Z.1:53 to
X.Y.Z.3:1026
Mon May 17 13:02:31 2004; ARP request for X.Y.Z.1; eth0; 154 bytes; from 000d29f31480 to ffffffffffff
```

The data provided by this application can aid in detection of worms and other anomalies on the network. For example, the following IPTraf 's output gives a trace of Blaster worm:

```
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4218 to X.Y.Z.24:80; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4219 to X.Y.Z.25:80; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4220 to X.Y.Z.26:80; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4221 to X.Y.Z.27:80; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4222 to X.Y.Z.28:80; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4223 to X.Y.Z.29:80; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4224 to X.Y.Z.30:80; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4225 to X.Y.Z.31:80; first packet (SYN)
................
```

TRAFFIC PARAMETERS

The anomaly detection and profiling methods used in NABOH system detect and profile abnormalities on the network using three traffic parameters: (i) bandwidth, (ii) number of hosts, and (iii) number of ports. These are described in this section.

TRAFFIC PARAMETER: BANDWIDTH

If a system only keeps track of number of remote hosts that each local host is connecting to, it would only be able to detect anomalies in worms and P2P traffic that generate a large number of connections. There are still other anomalies on the network which do not involve a large number of connections to remote hosts. For example, it is important to detect bandwidth hogs on the network in order to enforce a fair share of bandwidth. One of these anomalies is a local host connecting to one or a few remote hosts consuming a large amount of bandwidth for a long period of time, or a local host connecting to a remote host consuming a large amount of bandwidth. This can be a movie download or any other application, which takes away a large portion of the available bandwidth. For example, the following two graphs show a connection between two hosts for a long period of time at a high traffic rate.

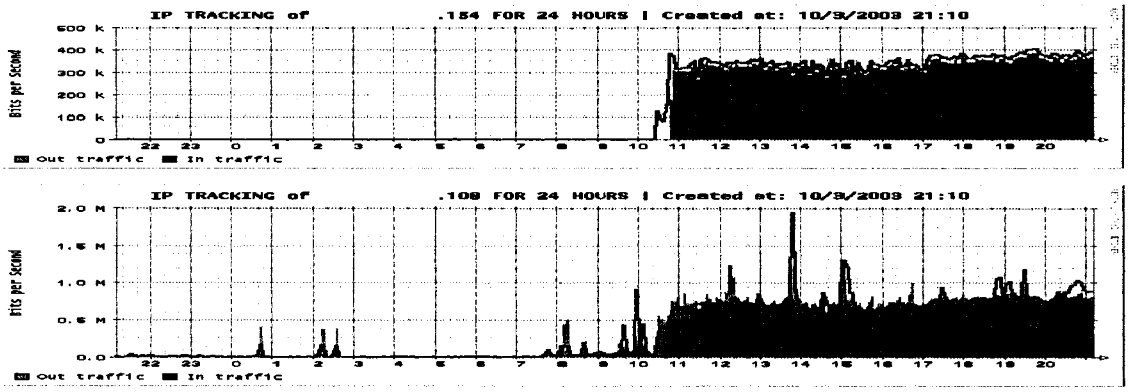


Figure 6 Bandwidth usage between two hosts

Also, P2P applications consume a large amount of bandwidth. In addition, some worms are also known for this type of behavior.

Normally, most of the hosts on the network are not servers. With this in mind, outbound traffic amount per local host should never be a high number in terms of bandwidth because normal local hosts are not providing any services. Therefore, examining bandwidth usage is a way of determining anomalies without looking at the content of the traffic. The sample traffic given by this traffic parameter consists of a local host's bandwidth usage.

TRAFFIC PARAMETER: NUMBER OF HOSTS

Keeping track of the number of remote hosts that each local host is connecting to aids in determining if there is an anomaly on the network such as caused by worms or P2P traffic. As documented earlier, an infected host tries to infect other hosts on the network thus it tries to connect to multiple hosts. The pattern of this behavior is depicted by the following:

```
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4218 to X.Y.20.1:445; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4219 to X.Y.20.2:445; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4220 to X.Y.20.3:445; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4221 to X.Y.20.4:445; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4222 to X.Y.20.5:445; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4223 to X.Y.20.6:445; first packet (SYN)
Sat Oct 18 18:55:22 2003; TCP;  48 bytes; from X.Y.Z.1:4224 to X.Y.20.7:445; first packet (SYN)
...........
```

The above trace indicates that host X.Y.Z.1 is trying to infect all the hosts in X.Y.20.0/24 subnet. Using number of hosts traffic parameter, one can determine the infected host's attack because the number of remote hosts connecting to the infected host will be a large number.

As another example, the following trace indicates that host X.Y.Z.2 is generating P2P traffic:

```
Mon Oct 20 08:35:46 2003; TCP;  73 bytes; from X.Y.Z.2:49592 to X.Y.Z.134:6346 ; FIN sent; 75 packets, 5578 bytes,
avg flow rate 0.00 kbits/s
Mon Oct 20 08:35:49 2003; TCP;  279 bytes; from X.Y.Z.2:49680 to X.Y.Z.67:24676 ; FIN sent; 7 packets, 1499 bytes,
avg flow rate 0.00 kbits/s
Mon Oct 20 08:35:53 2003; TCP;  52 bytes; from X.Y.Z.64:6346 to X.Y.Z.2:49678 ; FIN sent; 28 packets, 2637 bytes, avg
flow rate   0.00 kbits/s
Mon Oct 20 08:35:53 2003; TCP;  52 bytes; from X.Y.Z.2:49678 to X.Y.Z.64:6346 ; FIN acknowleged
Mon Oct 20 08:35:53 2003; TCP;  52 bytes; from X.Y.Z.2:49678 to X.Y.Z.64:6346 ; FIN sent; 28 packets, 2794 bytes, avg
flow rate   0.00 kbits/s
Mon Oct 20 08:35:53 2003; TCP;  52 bytes; from X.Y.Z.64:6346 to X.Y.Z.2:49678 ; FIN acknowleged
Mon Oct 20 08:35:58 2003; TCP;  52 bytes; from X.Y.Z.98:6346 to X.Y.Z.2:49659 ; FIN sent; 22 packets, 1968 bytes, avg
flow rate   0.00 kbits/s
Mon Oct 20 08:35:58 2003; TCP;  52 bytes; from X.Y.Z.2:49659 to X.Y.Z.98:6346 ; FIN acknowleged
Mon Oct 20 08:35:58 2003; TCP;  52 bytes; from X.Y.Z.2:49659 to X.Y.Z.98:6346 ; FIN sent; 22 packets, 3063 bytes, avg
flow rate   0.00 kbits/s
Mon Oct 20 08:35:58 2003; TCP;  60 bytes; from X.Y.Z.75:6350 to X.Y.Z.2:49682 ; first packet (SYN)
Mon Oct 20 08:35:58 2003; TCP;  52 bytes; from X.Y.Z.2:49682 to X.Y.Z.75:6350 ; first packet
Mon Oct 20 08:35:58 2003; TCP;  60 bytes; from X.Y.Z.99:5453 to X.Y.Z.2:49683 ; first packet (SYN)
Mon Oct 20 08:35:58 2003; TCP;  74 bytes; from X.Y.Z.2:49683 to X.Y.Z.99:5453 ; first packet
Mon Oct 20 08:35:59 2003; TCP;  52 bytes; from X.Y.Z.98:6346 to X.Y.Z.2:49659 ; FIN acknowleged
Mon Oct 20 08:35:59 2003; TCP;  54 bytes; from X.Y.Z.2:49683 to X.Y.Z.99:5453 ; FIN sent; 4 packets, 600    bytes, avg
flow rate   0.00 kbits/s
Mon Oct 20 08:35:59 2003; TCP;  60 bytes; from X.Y.Z.2:49686 to X.Y.Z.98:54848 ; first packet (SYN)
Mon Oct 20 08:35:59 2003; TCP;  261 bytes; from X.Y.Z.75:6350 to X.Y.Z.2:49682 ; FIN sent; 4 packets, 463  bytes, avg
flow rate   0.00 kbits/s
Mon Oct 20 08:35:59 2003; TCP;  52 bytes; from X.Y.Z.2:49682 to X.Y.Z.75:6350 ; FIN acknowleged
Mon Oct 20 08:35:59 2003; TCP;  52 bytes; from X.Y.Z.2:49682 to X.Y.Z.75:6350 ; FIN sent; 5 packets, 606    bytes, avg
flow rate   0.00 kbits/s
Mon Oct 20 08:35:59 2003; TCP;  52 bytes; from X.Y.Z.99:5453 to X.Y.Z.2:49683 ; FIN acknowleged
Mon Oct 20 08:35:59 2003; TCP;  52 bytes; from X.Y.Z.99:5453 to X.Y.Z.2:49683 ; FIN sent; 6 packets, 985    bytes, avg
flow rate   0.00 kbits/s
Mon Oct 20 08:35:59 2003; TCP;  52 bytes; from X.Y.Z.2:49683 to X.Y.Z.99:5453 ; FIN acknowleged
```

In the above IPTraf log the number of remote hosts that the local host, X.Y.Z.2, is trying to connect to is 400 hosts in a minute. This kind of anomaly can be detected by generating an alert in order to flag that a local host has exceeded connecting to a predefined limit on the number of remote hosts in a certain period of time. The sample traffic given by this parameter consists of the number of remote hosts that each local host is connecting to and vice versa.

TRAFFIC PARAMETER: NUMBER OF PORTS

Keeping track of the number of ports used by a local host in a certain period of time aids in determining anomalies on the network. For instance, P2P application normally opens a large number of ports to establish a large number of connections. Also, the behavior of some worms is to establish connections to other hosts by using different source ports but same destination port. Usually, an e-mail worm exhibits this type of behavior by generating a large amount of connections to distribute the worm through e-mail. In other words, an e-mail worm makes the infected host an e-mail server. Also, one to one attack can be detected if a remote host is trying to find any vulnerability on a local host by doing a port scan on a single host. For example, some backdoors' behavior can be detected through number of ports usage because the compromised local host becomes a server and normally opens a large number of ports where clients/hackers connect to access the host's services. The sample traffic given by this parameter consists of the number of ports used by the local host.

SLIDING WINDOWS

Sliding windows consist of the collected traffic parameters for every host. There are three parameters: (a) bandwidth usage, (b) number of remote hosts that a local host is connecting to and vice versa, and (c) number of ports used by the local host. As explained before, there are two sliding windows for each traffic parameter. Therefore, each host has six sliding windows. Each sliding window is based on an ordered set of values of a traffic parameter measured in 1-minute intervals. The traffic sample is over 30 minutes, thereby a 30-minute sliding window consists of 30 normalized members. For example, the six 30-minute sliding windows of a local host can be as follows:

Inbound bandwidth Sliding Window
(172.16.1.2 IBW (0.4 0.5 1.0 0.7 0.9 0.8 0.7 0.9 0.5 0.1 0.8 0.3 0.5 0.4 0.9 0.7 0.1 0.9 0.3 0.8 0.8 0.9 0.3 0.2 0.5 0.8 0.9 0.3 0.1 0.1))

Outbound bandwidth Sliding Window
(172.16.1.2 OBW (0.1 0.1 0.1 0.6 0.1 0.5 0.4 0.6 0.8 0.8 0.8 0.8 1.0 0.7 0.7 0.7 0.4 0.8 0.1 0.3 0.5 0.2 0.3 0.1 0.6 0.7 0.8 0.9 0.7 0.8))

Number of remote hosts being contacted Sliding Window
(172.16.1.2 LHRH (0.9 1.0 0.3 0.3 0.4 0.5 0.6 0.7 0.2 0.1 0.3 0.6 0.4 0.5 0.4 0.4 0.4 0.5 0.6 0.7 0.9 0.9 0.1 0.1 0.4 0.6 0.4 0.2 0.9 0.9))

Number of remote host containg local host Sliding Window
(172.16.1.2 RHLH (0.1 0.1 0.1 0.6 0.1 0.5 0.4 0.6 0.8 1.0 0.8 0.3 0.5 0.4 0.9 0.7 0.1 0.9 0.3 0.8 0.8 0.9 0.3 0.3 0.3 0.8 0.9 0.3 0.9 0.9))

Number of source ports Sliding Window
(172.16.1.2 SP (0.1 0.2 1.0 0.3 0.3 0.3 0.2 0.2 0.2 0.2 0.4 0.2 0.3 0.5 0.4 0.3 0.2 0.2 0.1 0.1 0.1 0.1 0.2 0.3 0.3 0.3 0.3 0.3 0.3 0.3))

Number of destination ports Sliding Window
(172.16.1.2 DP (0.6 0.6 0.0 0.7 0.9 1.0 0.7 0.8 0.5 0.1 0.9 0.3 0.5 0.4 0.2 0.3 0.1 0.3 0.2 0.3 0.4 0.5 0.6 0.9 0.8 0.8 0.9 0.8 0.5 0.4))

These sliding windows are analyzed by the three profiling methods documented earlier in this chapter in order to determine if there is an anomaly on the network.

PROFILING METHODS

As mentioned in the previous section, NABOH system uses three traffic parameters in order to detect and profile anomalies. These three parameters are (a) bandwidth usage, (b) number of remote hosts that a local host is connecting to and vice versa, and (c) number of ports used by the local host. Also, as documented in Section 2, NABOH system applies three profiling methods on the three traffic parameters of a host. These profiling methods are i) profiling based on Fuzzy Sets, (ii) profiling based on Traffic Shape, and (iii) profiling based on Fluctuations. The next three subsections explain the three profiling methods.

PROFILING METHOD: PROFILING BASED ON FUZZY SETS

This technique consists of converting a given pattern into a "temporal" fuzzy set in which each member is based on a parameter measured over a 1-minute interval. The fuzzy set is being referred to as "temporal" because order of the members matter. The traffic sample is over 30 minutes, thereby representing a 30-minute traffic sample by a 30-member fuzzy set. The sample may represent one of the three traffic parameters (a) bandwidth usage, (b) number of remote hosts that a local host is connecting to and vice versa, and (c) number of ports used by a local host. A fuzzy set representing a traffic sample is normalized so that it can be matched with existing signatures. For example, a sample based on bandwidth may look like the following:

(0.1 0.3 0.5 0.8 0.9 1.0 0.3 0.4 0.5 0.6 0.7 0.8 0.5 0.4 0.2 0.3 0.6 0.7 0.8 0.2 0.1 0.3 0.4 0.5 0.6 0.3 0.6 0.4 0.7 0.8)

In this method, a repertoire of known signatures is defined using fuzzy sets. These are referred to as signature fuzzy sets. For example, a problematic traffic could have the following normalized signature fuzzy set:

(0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 0.9 0.9 0.8 0.7 0.8 0.8 0.8 0.9 0.9 0.9 0.9 0.9 0.8 0.8 0.8 0.8 0.8 0.9 0.9 0.9)

Assuming that there are N signature fuzzy sets, the traffic sample fuzzy set is matched with each signature fuzzy set using Euclidean distance formula [Appendix D].

The signature fuzzy set, which yields the minimum Euclidean distance, is considered the candidate representing the type of traffic. However, the Euclidean distance has to be below a certain threshold in order to flag an alert. This threshold determines the sensitivity of the alert system.

PROFILING METHOD: PROFILING BASED ON TRAFFIC SHAPE

This method models a traffic sample to a repertoire of known functions in order to find the best matching shape for characterizing traffic behavior. For example, it is useful to know if the traffic is exponentially rising. Each function has a different degree of severity in reference to how potentially dangerous is the traffic represented by a traffic sample. For example, a polynomial of third degree has a higher degree of severity than its second degree counterpart. Therefore, traffic sample matched to a third degree polynomial would be considered more dangerous than those matched to a second degree polynomial.

Method of Least Squares is used to determine the closest matching function that models the traffic shape. Trapezoid Rule is used to integrate the traffic sample with respect to time in order to find the total volume of traffic. The combination of the results of the two methods is applied to a traffic sample in order to determine its degree of severity. Method of Least Squares and Trapezoid Rule of integration as applicable to NABOH are explained in the next subsections.

METHOD OF LEAST SQUARES

In this method, a traffic sample's shape is modeled after one of the known analytical functions representing threatening traffic. For example, if the traffic sample represents a sharply rising polynomial in half an hour period, it could be considered suspicious. Again, a traffic sample may represent one of the three traffic parameters (a) bandwidth usage, (b) number of remote hosts that a local host is connecting to and vice versa, and (c) number of ports used by the local host. As explained in Section 2, these yield six network traffic patterns. Method of Least Squares is used to determine the closest analytical function for each of the six network traffic patterns. Error is calculated in each case, and the function yielding the least error is selected to represent the traffic type.

A traffic sample is described as a normalized set of 30 members. Each member represents a traffic parameter measured over 1-minute interval. The 30 members represent a 30-minute period:

(0.1 0.3 0.5 0.8 0.9 1.0 0.3 0.4 0.5 0.6 0.7 0.8 0.5 0.4 0.2 0.3 0.6 0.7 0.8 0.2 0.1 0.3 0.4 0.5 0.6 0.3 0.6 0.4 0.7 0.8)

As an example, the following three functions are utilized to find the traffic shape of a given traffic pattern:

Function 1: $y = at + b$

Function 2: $y = at^2 + b$

Function 3: $y = at^3 + b$

where *y-values* are the data set values and *t-values* are the time intervals.

The Method of Least Squares is applied to each of the above functions to find the coefficients and calculate the error of each data set using $\ell2$ approximation [35]. The traffic sample has m points. Here, m represents a 30-minute period. In essence, m is 30 because each point is measured over 1 minute. Considering $\ell2$ approximation, the total error is given by:

Total Error for Function 1 = $\sum(a * t_k + b - y_k)^2$

Total Error for Function 2 = $\sum(a * t^2_k + b - y_k)^2$

Total Error for Function 3 = $\sum(a * t^3_k + b - y_k)^2$

The function yielding the least error is considered to be the best function representing the traffic sample. Appendix A show the detailed analysis of how the coefficients are determined based on minimum error as a function of the coefficients [35].

TRAPEZOID RULE

Trapezoid Rule of integration is applied in this profiling method to find the integral of the traffic sample with respect to time in order to calculate its volume. Note that a traffic sample may represent one of the three traffic parameters (a) bandwidth usage, (b) number of remote hosts that a local host is connecting to and vice versa, and (c) number of ports used by the local host. The volume of the traffic sample is needed in order to determine the degree of severity with respect to the selected traffic shape determined by Method of Least Squares. Appendix B explains the Trapezoid Rule of integration as applied to a traffic sample.

PROFILING METHOD: PROFILING BASED ON FLUCTUATIONS

This profiling method consists of measuring the fluctuations based on turning points in the traffic sample over a 30-minute interval. A traffic sample is described as a normalized set of 30 members. Each member represents a traffic parameter measured over a 1-minute interval. The sample may represent one of the three traffic parameters (a) bandwidth usage, (b) number of remote hosts that a local host is connecting to and vice versa, and (c) number of ports used by the local host. This method determines dramatic changes in the host's traffic behavior. For example, the following normalized traffic sample yields 28 fluctuations:

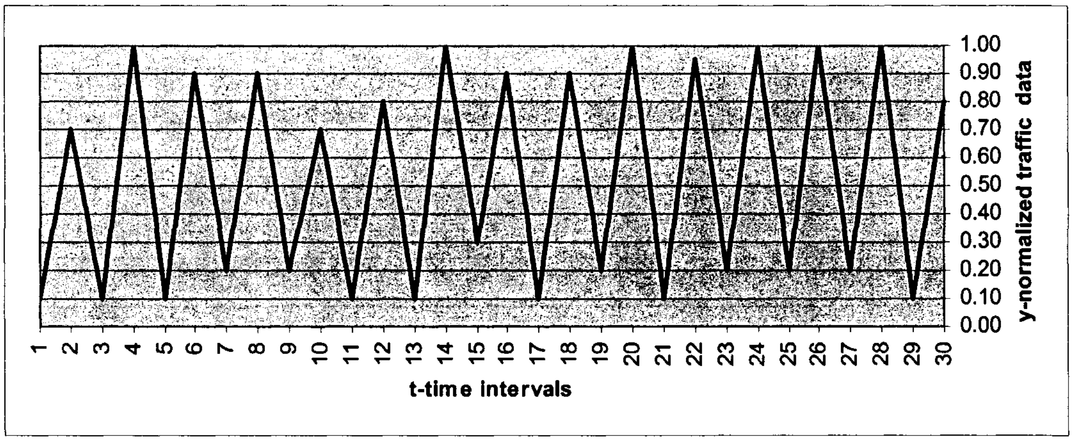


Figure 7 Graph of a normalized traffic sample

Normalized traffic sample (0.1 0.7 0.1 1.0 0.1 0.9 0.2 0.8 0.2 0.9 0.1 0.8 0.1 0.9 0.1 0.9 0.1 0.7 0.1 0.8 0.2 0.9 0.2 0.8 0.3 0.9 0.1 0.8 0.2 0.6)

For a traffic sample over a 30-minute interval, 28 fluctuations is the maximum value that a host's traffic behavior can give. Examining the fluctuation value, an anomaly is detected. However, obtaining also the rise and fall magnitudes of the fluctuations aids in determining the severity of the host's behavior.

ALERT SYSTEM

Alert system is created by the outcome of each of the profiling methods described in the previous section. Of course, the result of each method needs to be handled differently to produce a possible alert. Following is a description of alerts generated for each method.

i) Alerts: Profiling based on Fuzzy Sets

If the matching signature results in a Euclidean distance of less than the predefined tolerance, ε, then an alert is issued. Basically, it is reported that the traffic sample matches a certain known signature. Empirically, typical value of ε is around 0.01 [Appendix D].

ii) Alerts: Profiling based on Traffic Shape

If the selected function of the traffic sample is increasing, the following alert rules are applied depending on the function. Note that the traffic shape by itself does not determine if the traffic sample is an anomaly. That is why volume is used in conjunction with traffic shape to determine if there is an anomaly in host's traffic behavior.

a) Increasing Function 1: $f(t)=at + b$, where *a* is the slope

If *Slope > ST and Volume > (VT / (Slope + 1))* where *ST* is the Slope threshold, e.g. 1.5; *VT* is the Volume threshold, e.g.150 hosts or 10Mb or 150 ports over a 30-minute period. Note that higher slope reduces the effective volume threshold triggering an alert for dangerously increasing parameter.

b) Increasing Function 2: $f(t)=at^2 + b$

If *coefficient a* > *CT* and *Volume* > *VT/(|CT|* + *1)* where *CT* is the Coefficient threshold, e.g. 0.8; *VT* is the Volume threshold, e.g.150 hosts or 10Mb or 150 ports over a 30-minute period. Note that higher coefficient reduces the effective volume threshold triggering an alert for dangerously increasing parameter.

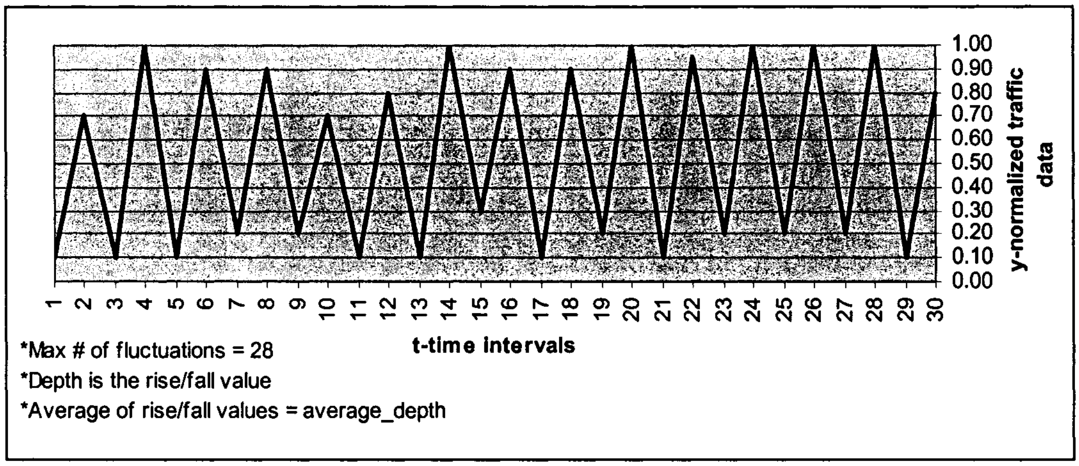
c) Increasing Function 3: $f(t)=at^3 + b$

If *coefficient a* > *CT* and *Volume* > *VT/(|CT| + 1)* where *CT* is the Coefficient threshold, e.g. 0.8; *VT* is the Volume threshold, e.g.150 hosts or 10Mb or 150 ports over a 30-minute period. Note that higher coefficient reduces the effective volume threshold triggering an alert for dangerously increasing parameter.

iii) Alerts: Profiling based on Fluctuations

The component of the alert system based on fluctuations examines the fluctuations value and the rise and fall magnitudes. The fluctuations value provides the number of changes in the host's traffic behavior and the rise and fall magnitude gives the degree of severity of the change. The alert rule is as follows:

If *fluctuations* ≥ 14 and *average_depth* > 0.2 where the *average_depth* is the average of the rise and fall magnitudes. The value of 0.2 is empirically determined to be a threshold for anomaly. In essence, if the traffic is rising and falling on average more than 20% of the highest magnitude, it represents an anomaly. The empirically determined threshold of 14 for number of fluctuations represents 50% of fluctuations because the maximum number of fluctuations is 28 as depicted in the figure below:



*Max # of fluctuations = 28
*Depth is the rise/fall value
*Average of rise/fall values = average_depth

Figure 8 Graph of a normalized traffic sample with 28 fluctuations

The above alert system determines that using profiling based on Traffic Shape method has the ability of detecting anomalies with high traffic in terms of the three traffic parameters: (a) bandwidth usage, (b) number of remote hosts that a local host is connecting to and vice versa, and (c) number of ports used by the local host. Also, profiling based on Fuzzy Sets method detects known anomalies through signature matching. Finally, profiling based on Fluctuations detects anomalies based on errant traffic where no shape can be defined. These methods have been tested on real-time data sets collected by a sniffer using a Linux system.

CHAPTER III

CONCLUSION

NABOH system developed in this thesis detects anomalies on the network. This system is based on a light-weight approach to gather intelligence whereby only headers of packets are examined. Three basic network traffic parameters are noted from the headers of the packets. The three traffic parameters are (a) bandwidth usage, (b) number of remote hosts that a local host is connecting to and vice versa, and (c) number of ports used by the local host. These three traffic parameters are collected for every host and entered into a sliding window to be analyzed and profiled using three profiling methods. These three profiling methods are i) profiling based on Fuzzy Sets, (ii) profiling based on Traffic Shape, and (iii) profiling based on Fluctuations. For instance, profiling based on Fuzzy Sets compares the traffic patterns with repertoire of known signatures, profiling based on Traffic Shape determines the shape of the traffic pattern, and profiling based on Fluctuations determines the degree of severity of change of the traffic pattern. NABOH system is capable of detecting anomalies such as worms, P2P traffic, and other anomalies that compromised the stability of the network.

NABOH system is equally effective in detecting and profiling anomalies in case of encrypted traffic because intelligence is gathered from the headers of packets only. It is a light-weight system to gather intelligence based on basic traffic parameters that are available from the headers of the packets only. Thus the system is referred to as NABOH (Network Anomalies Based On Headers). This approach makes it possible for (a) faster processing by looking at headers only, (b) preserving privacy by not looking at the contents of packets, and (c) comprehensively considering different traffic parameters to detect and profile network traffic anomalies.

REFERENCES

[1] Graham, R. (2001). *FAQ: Network Intrusion Detection Systems* [WWW document]. URL http://www.robertgraham.com/pubs/network-intrusion-detection.html

[2] Crosbie, M. and Price, K.. *Intrusion Detection Systems* [WWW document]. URL http://www.cerias.purdue.edu/coast/intrusion-detection/ids.html

[3] Kucan, B. (2003, December). *A Look Into The Viruses That Caused Havoc In 2003* [WWW document]. URL http://www.net-security.org/article.php?id=622

[4] HNF (2003, January). *MS SQL Worm Roundup* [WWW document]. URL http://www.net-security.org/article.php?id=369

[5] Cowley, S. and Williams, M., IDG News (January, 2003). *Slammer Worm Slaps Net Down, But Not Out* [WWW document]. URL http://www.pcworld.com/news/article/0,aid,108988,00.asp

[6] Graham, R. *Advisory: SQL slammer* [WWW document]. URL http://www.robertgraham.com/journal/030126-sqlslammer.html

[7] Jacques, R. (August, 2003). *Worm virus infecting 30,000 systems an hour* [WWW document]. URL http://www.accountancyage.com/News/1134555

[8] Verton, D. (August, 2003). *Blaster worm linked to severity of blackout* [WWW document]. URL

http://www.computerworld.com/securitytopics/security/recovery/story/0,10801,84510,00.html

[9] Hunt, K. (August, 2003). *Blackout Hits Parts of Eastern U.S., Canada* [WWW document]. URL http://kennethhunt.com/archives/000857.html

[10] Infrastructure, Telecommunications, and Networks Operation (ITNet) (2003). *Network Security - Hack Samples* [WWW document]. URL http://blue.utb.edu/itnet/default.asp?load=security_Hack-Samples

[11] Ellacoya Networks (2003). *P2P Control* [WWW document]. URL http://www.ellacoya.com/solutions/featured_p2p.html

[12] Bradley, T. *Internet/Network Security Glossary* [WWW document]. URL http://netsecurity.about.com/library/glossary/bldef-p2p.htm

[13] Nigrin, A. (1993). *Neural Networks for Pattern Recognition,* Cambridge, MA: The MIT Press, p.11

[14] Cannady, J. *Artificial Neural Networks for Misuse Detection* [WWW document]. URL http://csrc.nist.gov/nissc/1998/proceedings/paperF13.pdf

[15] HYPERDICTIONARY. *Expert Systems* [WWW document]. URL http://www.hyperdictionary.com/dictionary/expert+system

[16] Planquart, J. (2001). *Application of Neural Networks to Intrusion Detection* [WWW document]. URL http://www.sans.org/rr/papers/30/336.pdf

[17] Caswell, B. and Roesch, M. (2003). Snort [WWW document]. URL http://www.snort.org/about.html

[18] Bivens, A., Palagiri C., Smith, R., Szymanski, B., Embrechts, M. (2002). *NETWORK-BASED INTRUSION DETECTION USING NEURAL NETWORKS* [WWW document]. URL http://www.cs.rpi.edu/~szymansk/papers/annie02.pdf

[19] Russell, S. and Norvig, P. (1995). *Artificial Intelligence A Modern Approach*, Prentice Hall, p. 463

[20] Dickerson, J. and Dickerson, J. (2000). *Fuzzy Network Profiling for Intrusion Detection* [WWW document]. URL http://clue.eng.iastate.edu/~julied/publications/NAFIPSpaper2000.pdf

[21] Dickerson, J. (2002). *FIRE* [WWW document]. URL http://clue.eng.iastate.edu/~julied/research/FIRE/

[22] Noel, S., Wijesekera, D., and Youman, C. (2001). *MODERN INTRUSION DETECTION, DATA MINING, AND DEGREES OF ATTACK GUILT* [WWW document]. URL http://www.isse.gmu.edu/~snoel/IDS%20chapter.pdf

[23] Internet Security Systems (2002). *System Requirements RealSecure® Protection System for Networks and Servers* [WWW document]. URL http://documents.iss.net/literature/RealSecure/rs_sysreqs.pdf

[24] Townsend & Taphouse (2003). Cisco *NetRanger system version 2.2* [WWW document]. URL http://www.itsecurity.com/products/prod9.htm

[25] Porras, P. and Neumann, P. (1997). *Emerald: Event Monitoring Enabling Responses to Anomalous Live Disturbances* [WWW document]. URL http://www.sdl.sri.com/projects/emerald/emerald-niss97.html

[26] STAT. *Projects* [WWW document]. URL http://www.cs.ucsb.edu/~rsg/STAT/projects.html#NetSTAT

[27] CISCO. *Introducing Cisco Secure Intrusion Detection System* [WWW document]. URL: http://www.cisco.com/en/US/products/sw/secursw/ps2113/products_configuration_guide_chapter09186a008007f36d.html

[28] Rahman, S. (2000). *Network Intrusion Detection Systems* [WWW document]. URL

http://www.cs.utk.edu/~abdulrah/netsecurity/paper.html

[29] Security Consulting Company. *Intrusion Detection Systems List and Bibliography* [WWW document]. URL http://www-rnks.informatik.tu-cottbus.de/en/security/idsbody.html

[30] Barbara, D., Couto, J., Jajodia, S., Popyack, N., and Wu, N. (2001). *ADAM: Detecting Intrusions by Data Mining* [WWW document]. URL http://www.itoc.usma.edu/Workshop/2001/Authors/Submitted_Abstracts/paperT1A3(21).pdf

[31] Bambroo, M. (2003). *Learning Classifier Systems to Intrusion Detection Systems* [WWW document]. URL http://web.umr.edu/~tauritzd/courses/cs401/fs2003/project/Bambroo.ppt

[32] Computer Times - Technology (2003). *Smarter spam slayers* [WWW document]. URL http://computertimes.asiaone.com.sg/news/story/0,5104,1453,00.html

[33] backdoor [WWW document]. URL http://www.webopedia.com/TERM/b/backdoor.html

[34] *Trapezoid Rule* [WWW document]. URL http://www.np.edu.sg/mscIntMaths/Integrat/5_Trap.htm

[35] Cheney, W. and Kincaid D (1999). *Numerical Mathematics and Computing.* Brooks Cole

[36] Branch, J., Bivens, A., Chan, C., Lee, T., and Szymanski, B. (2002). *Denial of Service Intrusion Detection Using Time Dependent Deterministic Finite Automata.* [WWW document] URL citeseer.ist.psu.edu/branch02denial.html

[37] IOS. *Euclidean and Euclidean Squared.* [WWW document] URL http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Euclidean_and_Euclidean_Squared_Distance_Metrics.htm

APPENDIX A

METHOD OF LEAST SQUARES

In the profiling method based on Traffic Shape, method of Least Squares is applied to a traffic sample for each candidate function. The error is calculated in each case, and the function with the least error is selected to represent the traffic sample.

For example, the following three functions are utilized to categorize the shape of the traffic sample:

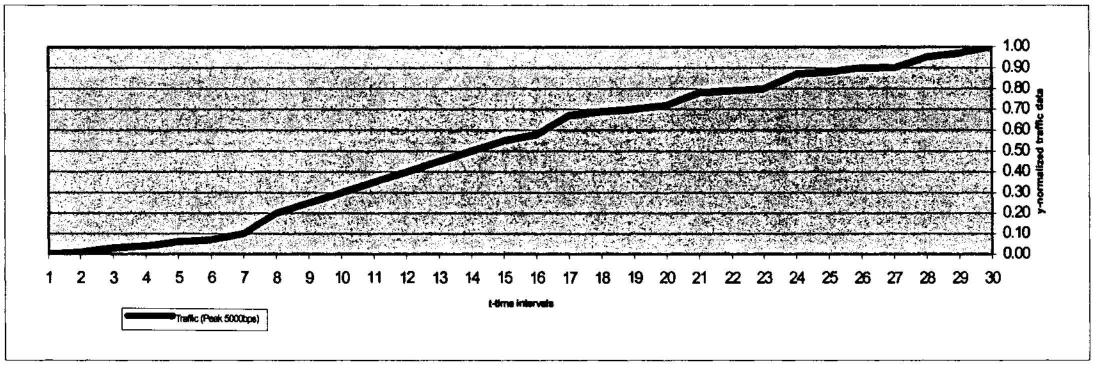
Function 1: $y = at + b$

Function 2: $y = at^2 + b$

Function 3: $y = at^3 + b$

where *y-values* are the data set values and *t-values* are the time intervals.

The method of Least Squares is applied to each of the above functions to find the coefficients and calculate the error of each data set using $\ell 2$ approximation. Following is an example of how the method of Least Squares is used to determine the shape of a traffic sample:



t:[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30]
y:[0 .01 .03 .04 .06 .07 .10 .20 .25 .30 .35 .40 .45 .50 .55 .58 .67 .69 .70 .72 .78 .79 .80 .87 .88 .90 .90 .95 .97 1]

Figure 9 A Traffic Sample

Function 1 (y=at + b) was found to be the best candidate to match the shape of the above traffic sample. The following steps explain how Function 1 was determined to be the best candidate for Figure 9:

1. Calculation for the total error of Function 1: y = at + b

1a. ℓ 2 approximation

The minimization of total error Ø(a,b) produces a best estimate of *a* and *b* values. The coefficients, a and b, are determined by differentiating the error function with respect to each parameter, and setting the result equal to zero [35].

$$\partial Ø/\partial a = 0, \quad \partial Ø/\partial b = 0$$

$$Ø(a,b) = \sum_{k=1}^{m} (at_k + b - y_k)^2$$

$$\partial Ø(a,b)/\partial a = 2 \sum_{k=1}^{m} (at_k + b - y_k) * t_k = 0$$

$$\partial Ø(a,b)/\partial b = 2 \sum_{k=1}^{m} (at_k + b - y_k) = 0$$

Let $c1 = \sum_{k=1}^{m} (t_k)^2, \quad c2 = \sum_{k=1}^{m} t_k, \quad c3 = \sum_{k=1}^{m} y_k * t_k, \quad c4 = \sum_{k=1}^{m} t_k, \quad c5 = \sum_{k=1}^{m} 1, \quad c6 = \sum_{k=1}^{m} y_k;$

$$\mathbf{ac_1 + bc_2 - c_3 = 0}$$

$$\mathbf{ac_4 + bc_5 - c_6 = 0}$$

Calculating for *c1, c2, c3,…, c6* yields:

$\mathbf{c_1}$=9455, $\mathbf{c_2}$=465, $\mathbf{c_3}$=326.55, $\mathbf{c_4}$=465, $\mathbf{c_5}$=30, and $\mathbf{c_6}$=15.51

1b. Solving for *a* and *b* from the above simultaneous equations yields:

$$\mathbf{a = ((c3 * c5) - (c2 * c6))/((c1 * c5) - (c2 * c4))}$$

$$a = ((\sum_{k=1}^{m} (y_k * t_k) * \sum_{k=1}^{m} 1) - (\sum_{k=1}^{m} t_k * \sum_{k=1}^{m} y_k)) / ((\sum_{k=1}^{m} (t_k)^2 * \sum_{k=1}^{m} 1) - (\sum_{k=1}^{m} t_k * \sum_{k=1}^{m} t_k))$$

$$\mathbf{b = ((c1 * c6) - (c4 * c3))/((c1 * c5) - (c2 * c4))}$$

$$b = ((\sum_{k=1}^{m} (t_k)^2 * \sum_{k=1}^{m} y_k) - (\sum_{k=1}^{m} t_k * \sum_{k=1}^{m} (y_k * t_k)) / ((\sum_{k=1}^{m} (t_k)^2 * \sum_{k=1}^{m} 1) - (\sum_{k=1}^{m} t_k * \sum_{k=1}^{m} t_k))$$

Results of *a* and *b* calculations

a=.0383 and **b**=-0.0771

1c. Total error for Function 1:

Total error for Function 1 = ∅(a,b) = $\mathbf{\sum(a * t_k + b - y)^2}$

Results of the calculation of the total error, ∅(a,b)

Total error of Function 1 = 0.0676

2. Calculation of the total error for Function 2: $y = at^2 + b$

2a. ℓ 2 approximation

$$\partial\varnothing/\partial a = 0, \quad \partial\varnothing/\partial b = 0$$

$$\emptyset(a,b) = \overset{m}{\underset{k=1}{\blacksquare}} (at^2_k + b - y_k)^2$$

$$\partial\emptyset(a,b)/\partial a = 2 \overset{m}{\underset{k=1}{\blacksquare}} (at^2_k + b - y_k) * t^2_k = 0$$

$$\partial\emptyset(a,b)/\partial b = 2 \overset{m}{\underset{k=1}{\blacksquare}} (at^2_k + b - y_k) = 0$$

Let $c1 = \sum_{k=1}^{m} (t^2{}_k)^2, \quad c2 = \sum_{k=1}^{m} t^2{}_k, \quad c3 = \sum_{k=1}^{m} y_k * t^2{}_k, \quad c4 = \sum_{k=1}^{m} t^2{}_k, \quad c5 = \sum_{k=1}^{m} 1, \quad c6 = \sum_{k=1}^{m} y_k;$

$$ac_4 + bc_5 - c_6 = 0$$

$$ac_1 + bc_2 - c_3 = 0$$

Calculating for *c1, c2, c3,…, c6* yields:

c1=5273999, c2=9455, c3=7502.77, c4=9455, c5=30, and c6=15.51

2b. Solving for *a* and *b* from the above simultaneous equations yields:

a = ((c3 * c5) - (c2 * c6))/((c1 * c5) - (c2 * c4))

$$a = ((\overset{m}{\underset{k=1}{\blacksquare}} (y_k * t^2_k) * \overset{m}{\underset{k=1}{\blacksquare}} 1) - (\overset{m}{\underset{k=1}{\blacksquare}} t^2_k * \overset{m}{\underset{k=1}{\blacksquare}} y_k)) / ((\overset{m}{\underset{k=1}{\blacksquare}} (t^2_k)^2 * \overset{m}{\underset{k=1}{\blacksquare}} 1) - (\overset{m}{\underset{k=1}{\blacksquare}} t^2_k * \overset{m}{\underset{k=1}{\blacksquare}} t^2_k))$$

b = ((c1 * c6) - (c4 * c3))/((c1 * c5) - (c2 * c4))

$$b = ((\overset{m}{\underset{k=1}{\blacksquare}} (t^2_k)^2 * \overset{m}{\underset{k=1}{\blacksquare}} y_k) - (\overset{m}{\underset{k=1}{\blacksquare}} t^2_k * \overset{m}{\underset{k=1}{\blacksquare}} (y_k * t^2_k)) / ((\overset{m}{\underset{k=1}{\blacksquare}} (t^2_k)^2 * \overset{m}{\underset{k=1}{\blacksquare}} 1) - (\overset{m}{\underset{k=1}{\blacksquare}} t^2_k * \overset{m}{\underset{k=1}{\blacksquare}} t^2_k))$$

Results of *a* and *b* calculations

a=0.0011 and b=0.1578

2c. Total error for Function 2:

Total error for Function 2 = $\emptyset(a,b) = \sum(a * t^2_k + b - y)^2$

Results of the calculation of the total error, Ø(a,b)

Total error of Function 2= 0.3897

3. Calculation of total error for Function 3: $y = at^3 + b$

3a. ℓ 2 approximation

$$\partial\mathcal{E}/\partial a = 0, \quad \partial\mathcal{E}/\partial b = 0$$

$$\mathcal{E}(a,b) = \sum_{k=1}^{m} (at^3_k + b - y_k)^2$$

$$\partial\mathcal{E}(a,b)/\partial a = 2 \sum_{k=1}^{m} (at^3_k + b - y_k) * t^3_k = 0$$

$$\partial\mathcal{E}(a,b)/\partial b = 2 \sum_{k=1}^{m} (at^3_k + b - y_k) = 0$$

Let $c1 = \sum_{k=1}^{m} (t^3_k)^2$, $c2 = \sum_{k=1}^{m} t^3_k$, $c3 = \sum_{k=1}^{m} y_k * t^3_k$, $c4 = \sum_{k=1}^{m} t^3_k$; $c5 = \sum_{k=1}^{m} 1$, $c6 = \sum_{k=1}^{m} y_k$;

$$ac_4 + bc_5 - c_6 = 0$$

$$ac_1 + bc_2 - c_3 = 0$$

Calculating for *c1, c2, c3,..., c6* yields:

c_1=3500931215, c_2=216225, c_3=182462.55, c_4=216225, c_5=30, and c_6=15.51

3b. Solving for *a* and *b* from the above simultaneous equations yields:

$$a = ((c3 * c5) - (c2 * c6))/((c1 * c5) - (c2 * c4))$$

$$a = ((\sum_{k=1}^{m} (y_k * t^3_k) * \sum_{k=1}^{m} 1) - (\sum_{k=1}^{m} t^3_k * \sum_{k=1}^{m} y_k)) / ((\sum_{k=1}^{m} (t^3_k)^2 * \sum_{k=1}^{m} 1) - (\sum_{k=1}^{m} t^3_k * \sum_{k=1}^{m} t^3_k))$$

$$b = ((c1 * c6) - (c4 * c3))/((c1 * c5) - (c2 * c4))$$

$$b = ((\sum_{k=1}^{m} (t^3_k)^2 * \sum_{k=1}^{m} y_k) - (\sum_{k=1}^{m} t^3_k * \sum_{k=1}^{m} (y_k * t^3_k)) / ((\sum_{k=1}^{m} (t^3_k)^2 * \sum_{k=1}^{m} 1) - (\sum_{k=1}^{m} t^3_k * \sum_{k=1}^{m} t^3_k))$$

Results of *a* and *b* calculations:

a=0.0000364 and **b**=0.7981

3c. Total error for Function 3:

Total error for Function 3 = $\varnothing(a,b) = \sum(a * t^3_k + b - y)^2$

Results of the calculation of the total error, $\varnothing(a,b)$

Total error of Function 3 =0.7981

4. Select the function with the minimum total error

The results given by the method of Least Squares for each candidate function using Figure 9 sample are as follows:

Function #	**$\varnothing(a,b)$**
1	0.06756
2	0.3897
3	0.7981

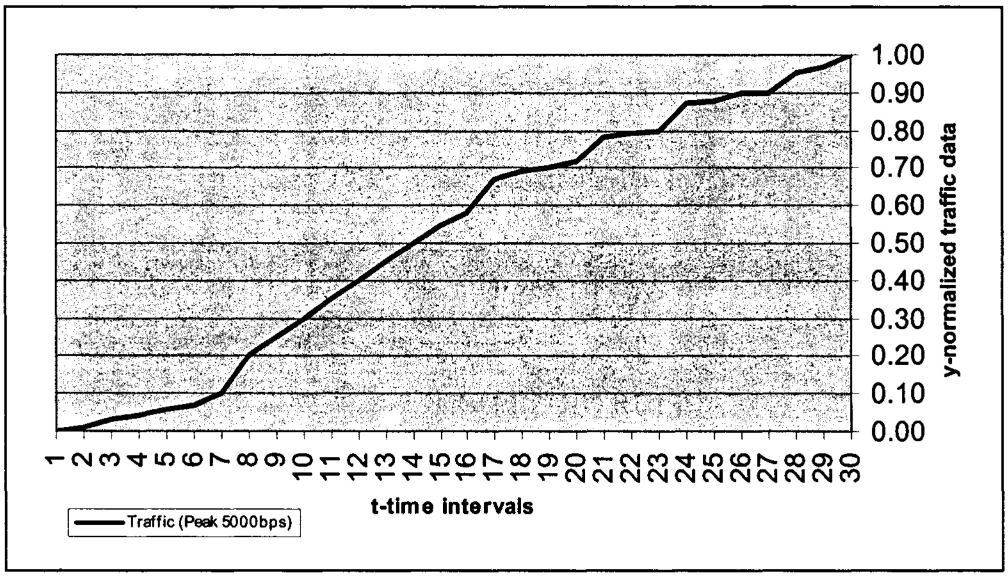
⇐ **Minimum $\varnothing(a,b)$**

As demonstrated above, the minimum total error $\varnothing_{min}(a,b)$ corresponds with Function 1. Therefore, Function 1 is selected to model the traffic sample shown in Figure 9.

APPENDIX B

TRAPEZOID RULE

The Trapezoid Rule of integration is applied on a traffic sample in the method of profiling based on traffic volume. The process of determining the volume of a traffic sample is explained in this appendix [34]:



t:[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30]

y:[0 .01 .03 .04 .06 .07 .10 .20 .25 .30 .35 .40 .45 .50 .55 .58 .67 .69 .70 .72 .78 .79 .80 .87 .88 .90 .90 .95 .97 1]

Figure 10 Graph of an Inbound Bandwidth Traffic Sample

1. The area of a trapezoid is given by:

   Height * (Base1 + Base2) /2

2. The approximate area under the curve is found by adding the area of the trapezoids.

$$\tfrac{1}{2}(y_0 + y_1)\,\Delta t + \tfrac{1}{2}(y_1+y_2)\,\Delta t + \tfrac{1}{2}(y_2 + y_3)\,\Delta t\ldots$$

where
number of values = $n+1$
$\Delta t = (t_n-t_0)/n$ where *t-values* are the time intervals of one minute each
n is the number of time intervals
$y_0, y_1, \ldots, y_n$ are the values that constitute the traffic sample

3. The above formula is simplified to give us the Trapezoidal Rule, for n number of trapezoids:

Trapezoid Rule ≈ $$\Delta t((1/2)*[y_0 + y_n] + \sum_{k=1}^{n-1} y_k)$$

Using Trapezoid Rule, the volume of the traffic sample of Figure 10 is calculated as follows:

Δt = 60 seconds
n is 29 time intervals
$y_0 = 0$
$y_n = 1$
$y_1, \ldots, y_{n-1}$ = .01, .03, .04, .06, .07, .10, .20, .25, .30, .35, .40, .45, .50, .55, .58, .67, .69, .70, .72, .78, .79, .80, .87, .88, .90, .90, .95, .97
peak= 5000 bits per second

Normalized Volume = 60s[0.5 * (0+1) + 14.51] = 900.6s
Traffic Volume = 900.6s * 5000bps = 4503000 bits

APPENDIX C

30-MINUTE PERIOD SLIDING WINDOW

A 30-minute period sliding window is taken to represent traffic behavior of a local host. As documented before, sliding windows consist of the collected traffic parameters for every host on the network. There are three traffic parameters: (a) bandwidth usage, (b) number of remote hosts that a local host is connecting and vice versa, and (c) number of ports used by the local host. Each sliding window consists of an ordered set of values of a traffic parameter measured in 1-minute intervals. Therefore, the traffic sample is over 30 minutes consisting of 30 normalized members. The following are six different Sliding Windows use to analyze an anomaly on the network:

- Inbound Bandwidth (IB)
- Outbound Bandwidth (OB)
- Number of Remote Hosts being contacted by a Local Host (LHRH)
- Number of Remote Hosts contacting a Local Host (RHLH)
- Number of Source Ports (SP)
- Number of Destination Ports (DP)

For example, the following 30-minute period sliding window on *bandwidth usage traffic parameter* portrays a high download. One can detect this type of traffic on the network by analyzing a 30-minute period sliding window of the local host.

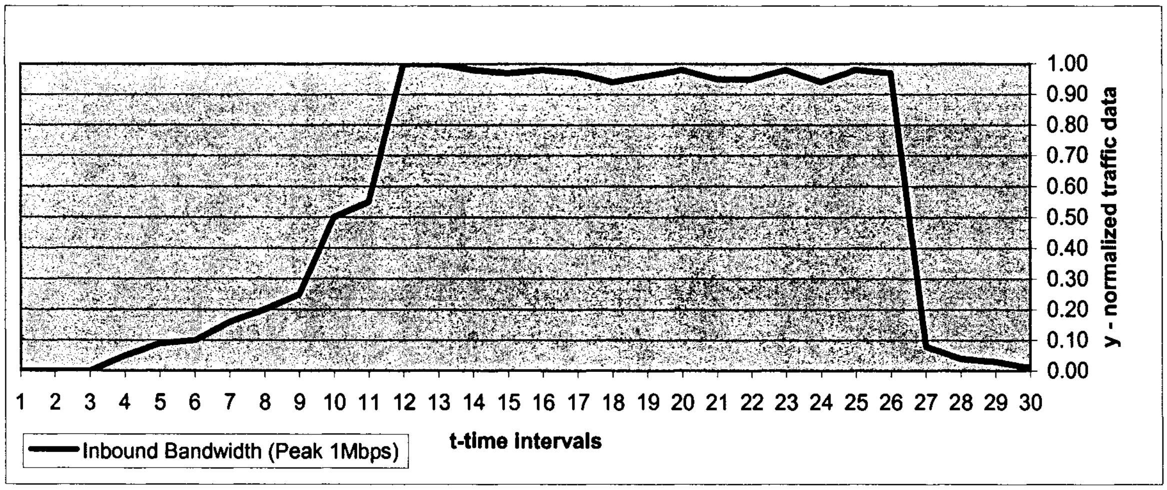


Figure 11 A Traffic Sample of a download

Also, a 30-minute period sliding window on the *number of hosts traffic parameter* can determine an anomaly on the network. For example, the following window on the number of remote hosts contacted by a local host provides enough detail to determine IP scanning characteristics on the local host behavior.

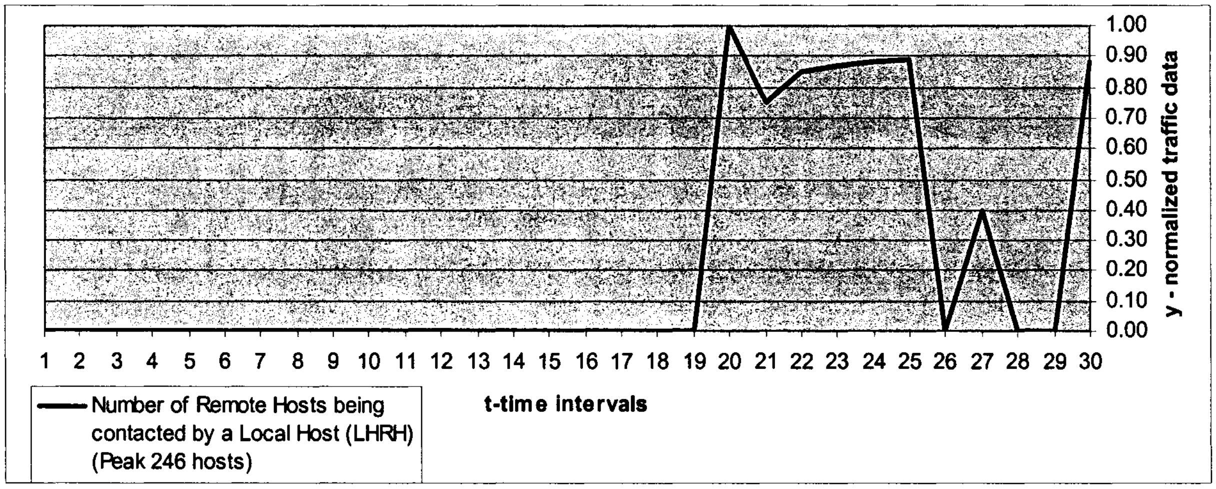


Figure 12 A Traffic Sample of an IP Scan

In addition, a 30-minute period sliding window on the *number of ports traffic parameter* is capable of detecting abnormal behavior on the network. For instance, the next sliding window on the number of source ports is a portrait of a port scanning behavior.

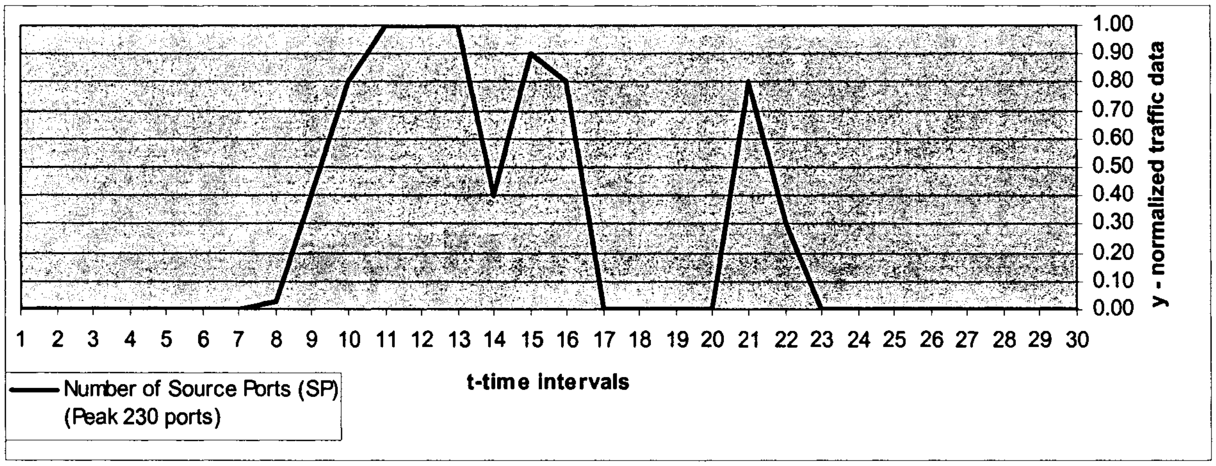


Figure 13 A Traffic Sample of a Port Scan

APPENDIX D

EUCLIDEAN DISTANCE

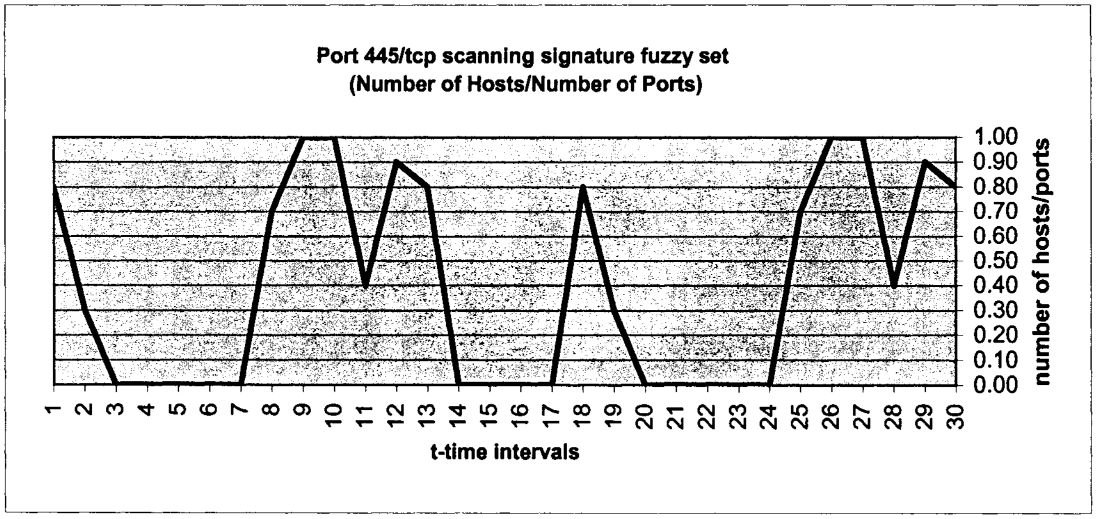
Euclidean distance is applied on the method of profiling based on Fuzzy Sets to find a known signature that matches the fuzzy set representing a traffic sample. The Euclidean distance examines the root of square differences between coordinates of a pair of objects. The formula of this method is the following [37]:

Euclidean Distance

$$d_{ij} = \sqrt{} \sum_{k=1}^{n} (y_{ik} - y_{jk})^2$$

where
n is the number of time intervals
$y_{i1} \ldots y_{in}$ are the values that constitute the fuzzy set of the traffic sample
$y_{j1} \ldots y_{jn}$ are the values that constitute the fuzzy set of the signature

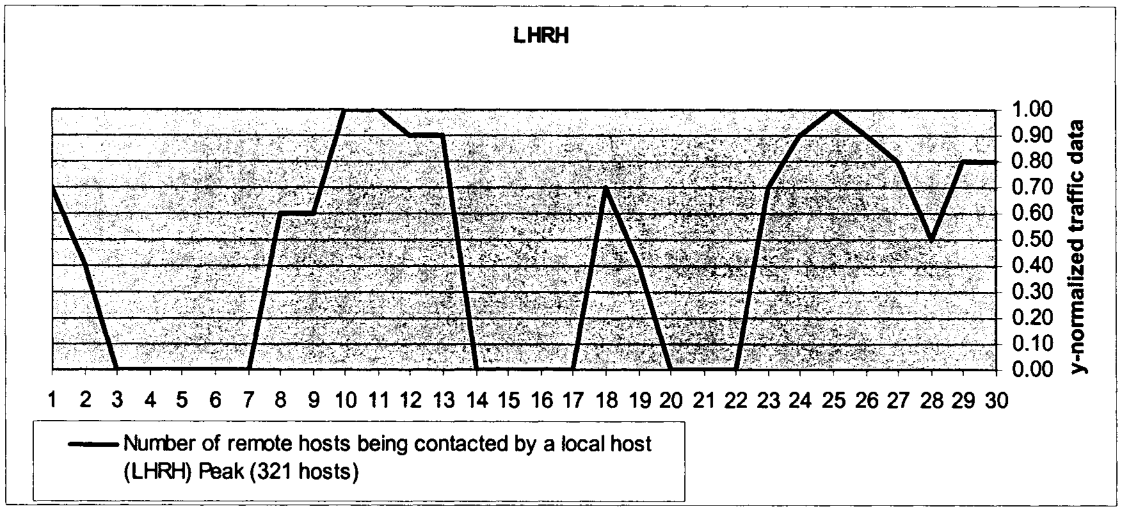
For example, the following signature is for 445/tcp port scanning behavior. It is a signature used for number of hosts and number of ports parameters. As documented before, a host scanning the network is going to try to connect to a large amount of hosts using a large amount of source ports. Taking this into consideration, the behavior on both parameters should be similar in this type of traffic.



Signature-TCP445_NH_NP (0.8 0. 3 0.0 0.0 0.0 0.0 0.0 0.71.0 1.0 0.4 0.9 0.8 0.0 0.0 0.0 0.0 0.8 0.3 0.0 0.0 0.0 0.0 0.0 0.7 1.0 1.0 0.4 0.9 0.8)

Figure 14 TCP Port 445 Scanning Signature Fuzzy Set for *Number of Hosts* and *Number of Ports* Parameters

The following graph displays a traffic sample fuzzy set with TCP port 445 scanning characteristics:



LHRH (0.7 0.4 0.0 0.0 0.0 0.0 0.0 0.6 0.6 1.0 1.0 0.9 0.9 0.0 0.0 0.0 0.0 0.7 0.4 0.0 0.0 0.0 0.7 0.9 1.0 0.9 0.8 0.5 0.8 0.8)

Figure 15 TCP Port 445 Scanning Traffic Sample of *Number of Hosts Traffic Parameter*

The calculation of the Euclidean distance between the fuzzy set of the above traffic sample and 445/tcp port scanning signature gives the following result:

Euclidean Distance=$\sqrt{2.04}$ =1.43

As documented in this thesis, profiling method based on Fuzzy Sets selects the signature with the minimum distance as the signature that best models the behavior of the traffic sample fuzzy set. Similarly, Euclidean distances of the traffic sample for other signatures are calculated.

VITA

Angelica M. Delgado

1728 Redbud Dr.
Brownsville, Texas 78526

Education:

The University of Texas at Brownsville and Texas Southmost College, B.S. Major in Computer Science, 2001

Work Experience:

2004 – Present	Network Analyst, The University of Texas at Brownsville and Texas Southmost College
2001 – 2004	Network Support Specialist, The University of Texas at Brownsville and Texas Southmost College
2000 – 2003	Research Assistant for Laser Interferometer Gravitational Wave Observatory (LIGO), The University of Texas at Brownsville and Texas Southmost College