Needles in Haystacks: Practical Intrusion Detection from Theoretical Results*

Gerald A. Marin and William H. Allen
Florida Institute of Technology
gmarin@fit.edu wallen@fit.edu

Abstract

Many researchers are working towards discovering techniques that can alert network administrators to the presence of previously unseen attacks in their networks. Here we focus on attacks, such as denial-of-service attacks, that depend on multiple packets being sent over minutes or, at least, several seconds. No definitive technique has been demonstrated that can guarantee a substantial probability of detection while keeping probability of false alarm at an acceptable level. However, theoretical work by Li, Jia, and Zhao (referenced below) describes an interesting approach based on observing changes to autocorrelations obtained over time from measured traffic. Their work provides a theoretical way of estimating probability of detection vs. probability of false alarm. They make assumptions concerning availability of a background template and normality of residuals that bear examining with real traffic and attacks. This paper attempts a practical approach.

1. Introduction

The problem of detecting intrusive (or malicious) traffic in a network includes two principal cases:

- Detecting attacks that have been analyzed previously (signature-based detection)
- Detecting previously unseen attacks.

Although the first case continues to challenge, especially with regard to false alarms and scalability, it is the second case that is the focus here – as it is in much of the current research. Numerous papers have been written on approaches to this problem using techniques including data mining, statistical analysis, artificial intelligence, neural networks, Markov modeling, sensor correlation, and analysis of management information data.

In theory, whatever the approach, each intrusion detection (ID) technique can be characterized by curves showing probability of detection, ***pd***, versus probability of false alarm, ***pfa***. These, or their ratio, are commonly referred to as Receiver Operating Curves (from radar detection literature). However, most of the current research proposes novel ID techniques and evaluates them empirically with no clear approach given for generalizing the work to establish dependable *pd* vs. *pfa* characteristics.

A theoretical paper by Li, Jia, and Zhao [1] offers a mathematical approach for intrusion detection that does yield such a characterizing of *pd* vs. *pfa*. The result raises a number of questions regarding practical implementation that go unanswered. Motivated by this previous work (but not limited to it) we describe efforts to develop a practical Intrusion Detection System (IDS) whose sensitivity is quantified by RoC graphs.

2. Overview of the Theory

Suppose that the time series X_t, $t > 0$ represents packet arrivals at a networking appliance, such as a router or switch. Assuming wide-sense stationarity of the process, one can estimate the normalized sample autocorrelation process using

$$r_X(k) = \frac{\sum_i (x_i - \bar{x})(x_{i+k} - \bar{x})}{\sqrt{\sum_i (x_i - \bar{x})^2}}$$

from any realization $x_1, x_2, \ldots, x_n$ where the x_i values are arrival counts over sequential time intervals of constant length, Δt. Suppose that the process Y_t is formed by adding to the background process, X_t, the arrival counts from a particular attack. The work by Li et al. suggests that the norms $\| r_X - r_Y \|$ will be normally distributed (we did not find this) and that differences in their distribution can be used to detect the presence of an attack (with some adaptation we did confirm).

3. Experimental Results

For our experiments we used the datasets described in Table 1. The first two datasets (BC and LBL) were captured from real networks (one from BellCore Research and the other from the Lawrence Berkeley Laboratory). The third dataset (CS) was captured from the Computer Science Department LAN at a large metropolitan university.

* Supported, in part, by a grant from the Office of Naval Research

Table 1: Background Datasets

Dataset Name	Source & Date	Packet Count	Data Length	Avg Pkts/Sec
BC	BellCore August 29, '89	941,299	50 minutes	313.8
LBL	LBL-4 January 21, '94	862,945	60 minutes	239.7
CS	CS LAN February 5, '03	1,966,418	90 minutes	364.2

In 1998 and 1999 researchers at the Massachusetts Institute of Technology (MIT) Lincoln Labs created a test environment that included synthetically generated background traffic plus live attack traffic [2]. For the research reported here several of the Lincoln Lab attacks were extracted and added to the backgrounds listed in Table 1.

If an attack includes only a few packets (intended, for example, to cause a system crash by exploiting a specific operating system vulnerability at the target), then one cannot expect that a technique based on changes in packet arrival patterns will be successful. Thus, we tested against attacks whose negative effects persist only while the attack stream continues. For clarity, we refer to these as traffic exploits (TE's). Table 2 lists three TE attacks from Lincoln Labs data that we used here for tests in all three backgrounds. It includes the attack peak to background average ratio as these particular attacks were found in the original Lincoln Lab data.

Table 2: Attack Characteristics

Attack Name	Duration (sec)	AtkPeak/BkgAvg (ratio)
Apache	519	24.8
Neptune	410	2.9
Smurf	4	29.0

Table 3 presents selected results for each attack type in each of the backgrounds. The integers in attack names, i.e. Apache4, represent the ratio of the attack "peak" to background "average." Note that both Apache4 and Neptune 1 were below the ratio at which the attacks were encountered in the original data. Neptune1 can readily be detected, but Smurf16 has a low *pd*. These results are best understood by examining the "shape" of the attack in Figure 1. The Apache, Neptune, and Smurf attack are depicted at three different scales, with and without background traffic. Given that Smurf appears as a single needle in the background haystack, it is not terribly surprising that the results of Smurf are not promising.

Table 3: Probability of Attack Detection

Name	Back-ground	Attack Windows	PD* $\rho = 0.9$
Apache4	BC	8	0.97
	LBL	8	0.99
	CS	8	0.99
Neptune1	BC	7	0.95
	LBL	7	0.99
	CS	7	0.98
Smurf16	BC	1	0.57
	LBL	1	0.52
	CS	1	0.02

*Applies only when attack occupies multiple sliding windows; thus, does not apply to Smurf attack.

In summary we note the following:

- The shape of the attack plays a major role in the effectiveness of this technique.
- Though not illustrated here, it is especially noteworthy that even the 4-second Smurf attack may be detected by this technique in the classical Internet backgrounds compared to the extremely variable CS LAN background.
- A change in attack intensity (packets-per-second) has a significant effect on detection probability depending on duration. It does appear from our experiments that the more robust (and therefore more threatening) the attack, the more likely its detection.

4. References

[1] Ming Li, Weija Jia, and Wei Zhao, "Decision Analysis of Network Based Intrusion Detection Systems for Denial-of-Service Attacks", Proceedings IEEE Conferences on Info-tech and Info-net, 2001.

[2] Richard Lippmann, Joshua Haines, David Fried, et. al. "The 1999 DARPA Off-line Intrusion Detection Evaluation", Computer Networks, 34, 2000.

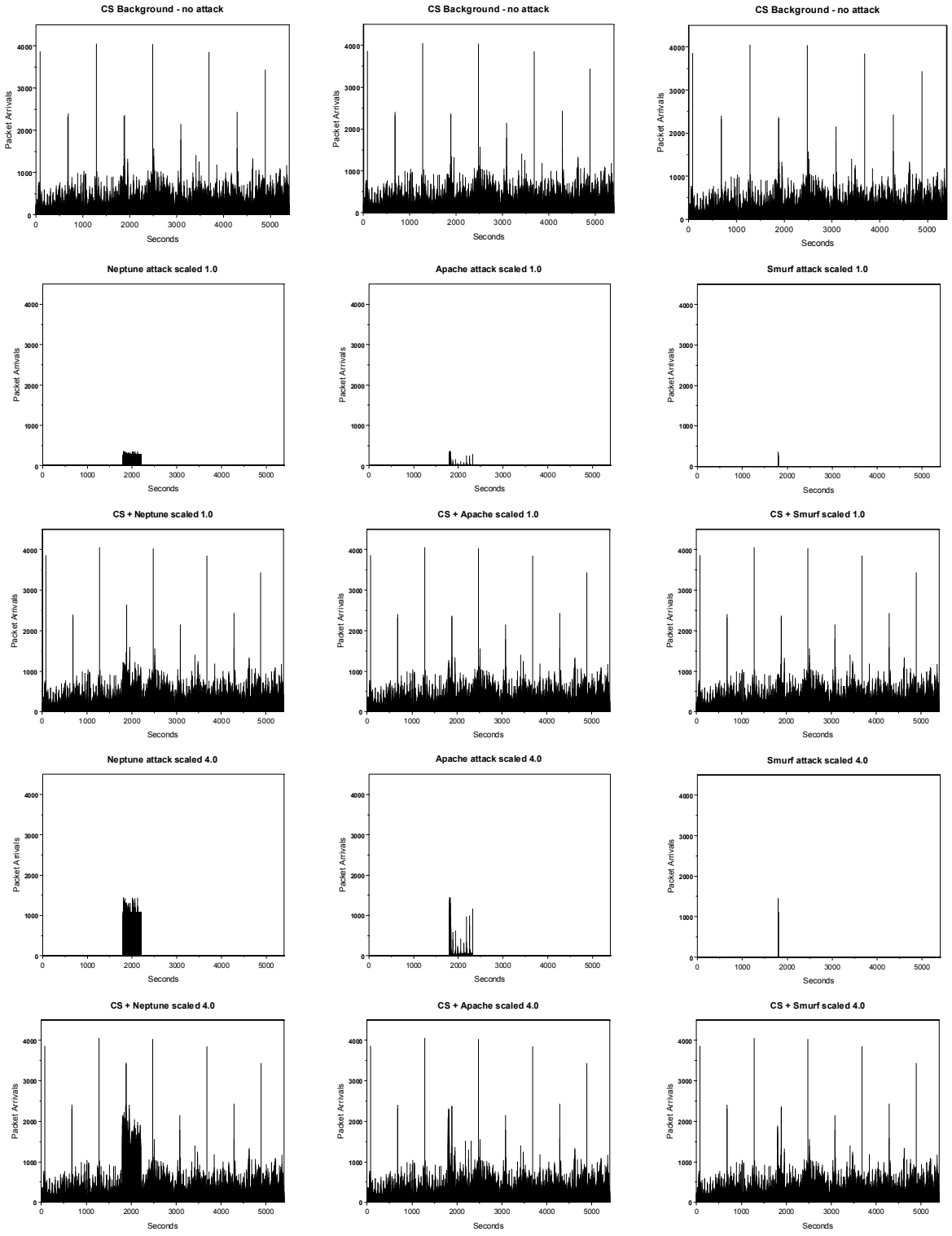

Figure 3. Finding needles in haystacks: each column shows the CS background, the attack traffic shape (scale 1), the background plus attack (scale 1), the attack traffic shape (scale 4), the background plus attack (scale 4). Column 1 depicts the Neptune cases, column 2 depicts the Apache cases, and column 3 depicts the Smurf cases.