The 1998 Lincoln Laboratory IDS Evaluation
A Critique*

John McHugh

CERT® Coordination Center
Software Engineering Institute
Carnegie Mellon University
jmchugh@cert.org

Abstract. In 1998 (and again in 1999), the Lincoln Laboratory of MIT conducted a comparative evaluation of Intrusion Detection Systems developed under DARPA funding. While this evaluation represents a significant and monumental undertaking, there are a number of unresolved issues associated with its design and execution. Some of methodologies used in the evaluation are questionable and may have biased its results. One of the problems with the evaluation is that the evaluators have published relatively little concerning some of the more critical aspects of their work, such as validation of their test data. The purpose of this paper is to attempt to identify the shortcomings of the Lincoln Lab effort in the hope that future efforts of this kind will be placed on a sounder footing. Some of the problems that the paper points out might well be resolved if the evaluators publish a detailed description of their procedures and the rationale that led to their adoption, but other problems clearly remain.
Keywords: Evaluation, IDS, ROC Analysis

1 Introduction

The most comprehensive evaluation of research Intrusion Detection Systems that has been performed to date is an ongoing effort by MIT's Lincoln Laboratory, performed under DARPA sponsorship. While this work is flawed in many respects, it is the only large scale attempt at an objective evaluation of these systems of which the author is aware. As such, it does provide a basis for making a rough comparison of existing systems under a common set of circumstances and assumptions.

It is important to note that the present paper is a critique of existing work, not a direct technical contribution or a proposal for new efforts, *per se*. Its purpose is to examine the work done by the Lincoln Laboratory group in a critical but scholarly fashion, relying on the public (published) record to the greatest extent possible. The role of the critic is to ask questions and to point out failings and omissions, but not necessarily to provide solutions to all the issues raised. Indeed, the problem is large and complex and one of the likely reasons

* This work was sponsored by the U.S. Department of Defense.

for many of Lincoln's failures that its size and complexity clearly outstripped the resources available to apply to it. Many of the issues raised in this paper will require substantial resources and effort to resolve, and, at the time of writing, these resources were not available. Still, it is to be hoped that the community as a whole will be able to address these problems in connection with future efforts.

The analysis given here is presented with the goal of promoting a discussion of the difficulties that are inherent in performing objective evaluations of software. Although the software in question performs security related functions, the questions raised in its evaluation should be of interest to the broader software engineering community, particularly to that portion of the community that deals with software testing or evaluation and reliability estimation. As far as we have been able to determine, no comparable efforts have been reported elsewhere in the software evaluation and testing community. Only the usage modeling and statistical testing used by the Cleanroom methodology [13, Ch. 10] seems to come close.

We concentrate on the 1998 evaluation. The 1999 evaluation was under way when the original version of this paper was written and its results, though presented in a number of meetings (including RAID 2000) have not been published in detail. Many of the changes made during the 1999 evaluation do not affect the observations or conclusions of this paper. The data used in 1999 was similar in form to that of 1998. Sessions were not identified in the training data, making the unit of analysis problem described in section 5.1 more difficult. TCP dump data was sensed both inside and outside the target system and a Windows NT victim was added. A relatively permissive security policy was described for the targets. A wider variety of attacks were represented in the data, including several insider attacks. Initial presentations of the 1999 results relied heavily on ROC analysis (see section 5.2), but more recent presentations have dropped this approach entirely. Missed attacks were analyzed in some detail with investigators being asked to explain why their system did not detect them. In many cases, especially for rule based systems, the misses were due to decisions by the investigators (encouraged by the sponsor) to concentrate on detection technique at the expense of complete rule bases. It would be interesting to rescore the 1999 results, using the optimistic assumption that such misses are correctable in a production version of the system.

The discussion begins with a consideration of the methods used to generate the data used for the evaluation. There are a number of questions that can be raised with respect to the use of synthetic data to estimate real world system performance. We concentrate on two of these; the extent to which the experimental data is appropriate for the task at hand and the possible effects of the architecture of the simulated test environment. This is followed by a discussion of the taxonomy developed to categorize the exploits involved in the evaluation. The taxonomy used was developed solely from the attacker's point of view and may introduce a bias in evaluating manifestations seen by the attacked.

The Lincoln Lab evaluation uses the ROC, variously known as the receiver operating curve or relative operating characteristic as the primary method for

presenting the results of the evaluation. This form of analysis has been used in a variety of other fields, but it appears to have some unanticipated problems in its application to the IDS evaluation. These involve problems in determining appropriate units of analysis, bias towards possibly unrealistic detection approaches, and questionable presentations of false alarm data.

2 Evaluation Overview

The descriptions of the evaluation that have appeared in print leave much unsaid and it may be that a more detailed exposition of the work will alleviate some of the criticisms contained in this paper. The most detailed descriptions of the work available at the present time are Kristopher Kendall's BS/MS Thesis [6] and a paper [9] presented at DISCEX in January, 2000. In addition, the Lincoln Lab team has made presentations on the experiment at various meetings attended by the author. These include the August 1999 DARPA PI meeting in Phoenix, AZ and RAID 99. Presentations [8, 4] similar to ones given at those meetings also appear at the Lincoln Lab experiment site, http://ideval.ll.mit.edu[1].

According to the DISCEX paper [9], "The primary purpose of the evaluations is to drive iterative performance improvements in participating systems by revealing strengths and weaknesses and helping researchers focus on eliminating weaknesses." The experiment claims to provide "unbiased measurement of current performance levels." Another objective is to provide a common shared corpora of experimental data that is available to a wide range of researchers.

While these goals are laudable, it is not clear that the way in which the evaluation has been carried out is consistent with the goals. In section 3 we will discuss the adequacy of the data set used during the evaluation, suggesting that, at best, its suitability for this purpose has not been demonstrated. The way in which the results of the evaluation have been presented (through the use of ROC and ROC like curves as discussed in section 5.2) seems to demonstrate a bias towards systems that can be tuned to a known mix of signal and noise, even though the appropriate tuning parameters may not be possible to discover in the wild. Each of these factors will be discussed further in the appropriate sections.

Many of the systems evaluated by the Lincoln Lab group have been described in a variety of technical publications, some of which are cited in the DISCEX paper [9]. Each system under test was evaluated by its developers who adapted the data as necessary to fit the system in question [9, Section 7]. It is highly likely the disparate behaviors of the individual investigators introduced unintentional biases into the results of the evaluation, but there has been no discussion of this possibility in any of the presentations or in the DISCEX paper.

[1] This site is password protected. For information concerning access, contact `intrusion@sst.ll.mit.edu`.

3 The Evaluation Data

For reasons having to do with privacy and the sensitivity of actual data content, the experimenters chose to synthesize both the background data and the attack data used during the evaluation. There are problems with both components. The data also reflects problems that are inherent in the architecture used to generate it. The generated data is intended to serve as corpora for present and future experimenters in the field. As such, it may have a lasting impact on the way IDS systems are constructed. Unless the performance of an IDS system on the corpora can be related accurately to its performance in the wild, there is a risk that systems may be biased towards unrealistic expectations with respect to true detections, false alarms, or both. It is also necessary to ensure that the corpora are sufficiently large so that deviations from the desired norm do not alter evaluation results.

The data generated for the evaluation consists of two components, background data that is intended to be completely free of attacks and attack data that is intended to consist entirely of attack scenarios. The test stream results from simultaneously generating and interleaving of the two components. If we view background data as noise and attack data as signal, the IDS problem can be characterized as one of detecting a signal in the presence of noise. The evaluation produces two measures, one primarily a function of the noise, the other primarily a function of the signal embedded in noise. Given this approach, it is necessary to ensure that both the signal and the noise used for the evaluation affect the systems under test in a manner related to signals and noise that occur in real deployment environments.

3.1 Background Data

The process used to generate background data or noise is only superficially described in the thesis and presentations. The data is claimed to be similar to that observed during several months of sampling data from a number of Air Force bases, but the statistics used to describe the real traffic and the measures used to establish similarity are not given, except for the claim that word and word pair statistics of email messages match those observed. The DISCEX paper [9, Sections 3 and 4] devotes approximately a page to a discussion of this issue and makes a broad claim that the data is similar to that seen on operational Air Force bases. It has been observed that internet site behaviors differ greatly[2] and, while it is possible that Air Force bases form an exception, the notion of a typical mix of background traffic should be viewed with some skepticism. If this skepticism is justified, and if the nature of the background traffic is shown to have a substantial impact on IDS performance, we see no alternative to the provision of much more extensive corpora of evaluation data.

[2] This was pointed out by one of the anonymous reviewers for RAID 2000. Although it may not hold for Air Force bases, it is a factor to consider in extending the results of the evaluation to more general environments.

As far as can be determined from the record, neither analytical nor experimental validation of the background data's adequacy was undertaken prior to the evaluation. No rationale is given that would allow a reader to conclude that the systems under test should exhibit false alarm behaviors when exposed to the artificial background data that are similar to those that they exhibit when exposed to "natural" data. This is particularly troublesome since the metric used for the evaluation of the IDS systems under test is an operating point characterized by the percentage of detected intrusions at a given false alarm rate or percentage. False alarms should arise exclusively from the background data, and it would appear incumbent upon the evaluators to show that the false alarm behavior of the systems under test is not significantly different on real and synthetic data.

Real data on the internet is not well behaved. Bellovin reported on anomalous packets [2] some years ago. Observations by Paxson [12] indicate that the situation has become worse in recent years with significant quantities of random garbage being frequently observed on the internet. This internet "crud" consists of legitimate but odd looking traffic. Poor implementations of protocols often result in spontaneous packet storms that are indistinguishable from malicious attempts at flooding. Many of the packets that Bellovin and Paxson observe could (and probably should) be interpreted as suspicious. As far as we can tell, such packets were not included in the background traffic.

None of the sources that we have examined contain any discussion of the data rate and its variation with time is not specified. This may be another critical factor in performing an evaluation of an IDS system because it appears that some systems may have performance problems or may be subject to what are, in effect, denial of service attacks when deployed in environments with excessive data rates[3]. We have performed a superficial examination of several days of the tcpdump training data. The results indicate averages in the 10 to 50 kilobit per second range over the 22 hour period. Given that most of the activity occurs during working hours, the daylight rate may be 2 or 3 times this. In contrast, data rates at the Portland State University Computer Science department ($\approx$100 workstations) and Engineering school (several hundred) are 1 and 10 megabits per second respectively. Paxson indicates sustained data rates in excess of 30 megabits per second [12] on the FDDI link monitored by the Bro IDS. Since one would expect false alarm rates to be proportional the background traffic rate for a given mix, the false alarm rates reported by Lincoln Lab may need to be adjusted.

3.2 Attack Data

Similar arguments can be made about the synthetic attack data. The attacks used were implemented via scripts and programs collected from a variety of sources. As far as can be determined from the available descriptions, no attempt was made to ensure that the synthetic attacks were realistically distributed in

[3] This factor may not be relevant for an offline evaluation, but we would expect the evaluators to consider timing.

the background noise. This may or may not be significant, depending on several factors, including the use to be made of the evaluation results, but it raises several issues. Reporting an aggregate result over any mix requires a strong caveat to the effect that the results may not apply to other mixes. This is more important if the mix is atypical[4]. In addition, some systems that require training on a known mix of attack and background data may be sensitive to the mix and fail to perform as well on substantially different mixes.

Kendall [6, Section 12.2] describes the total number of attacks in various categories that were included in the training and test data sets. Some 300 attacks were injected into 10 weeks of data, an average of 3 to 4 attacks per day. Kendall gives a tabulation of the attack data in [6, Table 12.1]. In each of the major categories of the attack taxonomy (User to Root, Remote to Local User, Denial of Service, and Probe/Surveillance) the number of attacks is of the same order (114, 34, 99, and 64). This is surely unrealistic as current experience indicates that Probe/Surveillance actions are by far the most common attack actions reported.

An aggregate detection rate based on the experimental mix represented in the corpora is highly unlikely to reflect performance in the field. If a more detailed analysis and presentation of the data were to be used, the attack mix would be less significant, although the user of the evaluation results would have to invest more time and effort in understanding the results and their significance. Particular care would be needed to ensure that the presentation does not obscure the characteristics of the evaluated systems in this case. For example, the attack taxonomy used combines attacks that have widely differing manifestations. Reporting, for example, that a given system detected 60% of the denial of service attacks may reflect the system's ability to detect some kinds of manifestations and not others rather than reflecting on its ability to detect that taxonomic category of attack. Thus, even reporting performance by taxonomic attack category may be misleading if the distribution of attack manifestations is not explicitly considered.

The evaluation data represents an attempt at creating a somewhat realistic test environment in which known attacks are executed in a background of normal activity. A number of researchers have said that it would also be useful to present a variety of attacks under ideal conditions without background traffic so as to separate detection characteristics from the confounding effects of the background traffic. Providing attack data in this form was not one of the goals of the Lincoln effort.

[4] While it is clear that there is no such thing as a typical mix of attacks, experience over the past few years indicates that hacker activity on the internet (the attack population represented in the experimental mix) consists primarily of probe activities followed by fairly large numbers of the most recently popular attack *du jour*, followed by a sprinkling of less recently publicized attacks against well known vulnerabilities.

3.3 Eyrie AFB

The simulated data is said to represent the traffic to and from a typical Air Force Base, referred to as Eyrie AFB. The thesis [6, Figure 3-1] and the information available from the Lincoln Lab web site seem to differ on the details of the configuration. The host list for weeks 3-7 lists additional hosts linux1 – linux10 which are probably implemented on the additional Linux target mentioned in the thesis, but the week 3-7 network diagram does not show this host. The DISCEX paper [9, Section 3] is less specific.

The thesis contains a list of the attacks [6, Appendix A] from the test phase of the evaluation. 45 attacks target Pascal, 28 target Marx, 12 target Zeno, 10 target one of the virtual Linux machines, and 5 (all the same scenario) target the router. The only attacks that attempt to access any of the other simulated machines at Eyrie are probes or scans for which no response is necessary. The skewed nature of the attack distribution may affect the evaluation. By the end of the training period, it should have been clear to the testers that only a small subset of the systems are actually subject to interactive attacks. Tuning or configuring the IDS under evaluation to look only at these systems would be an effective way to reduce false alarms and might raise the true alarm rate by reducing noise. This appears to fall within the letter, if not the spirit, of the 1998 rules though there is no evidence that it was done by any of the participants.

Although it is claimed that the traffic used in the evaluation is similar to that of a typical Air Force base, no such claim is made for the internal network architecture used. The unrealistic nature of the architecture is implicitly acknowledged by Kendall [6, Section 6.8] where it is noted that the flat structure of the simulation network precluded direct execution of a "smurf" or ICMP echo attack. It is not known whether the flat network structure used in the experiment is typical of Air Force bases, but this seems doubtful as does the relatively small host population. Investigation of whether this as well as the limited number of hosts attacked affect the evaluation is needed. Certainly, intrusion detection systems that make a stateful evaluation of the traffic stream are less likely to suffer from resource exhaustion in such a limited environment.

3.4 Does it matter?

Perhaps and perhaps not. Many experiments and studies are conducted in environments that are contrived. Usually, this is done to control for factors that might confound the results. When it is done, however, the burden is on the experimenter to show that the artificial environment did not affect the outcome of the experiment. A fairly common method of demonstrating that the experimental approach being used is sound is to conduct a controlled pilot study to collect evidence supporting the proposed approach. As far as we can tell, no pilot studies were performed either to validate the use of artificial data or to ensure that the data generation process resulted in reasonably error free data. The evaluators at Lincoln Lab have not shown that the test environment that they created does not confound the evaluation in ways that would affect its objectives.

3.5 Training and Test Data Presentation

The evaluators prepared datasets for the purposes of "training" and "test." The training set consists of seven weeks of data covering 22 hours per day, 5 days per week. As discussed in section 5.1 below, the training data contains attacks that are identified in the associated lists. It also contains examples of anomalies, here defined rather restrictively as departures from the normal behaviors of individual system users rather than the more common usage of abnormal or unusual events.

The apparent purpose of this data was to provide the researchers being evaluated with corpora containing known and identified attacks that could be used to tune their systems. For the systems based on the detection of anomalies, the training data was intended to provide a characterization of "normal," although the presence of attacks in the data renders it questionable from this standpoint. The question of the adequacy of this data for its intended purpose does not seem to have been addressed. There is no discussion, for example, of whether the quantity of data presented is sufficient to train a statistical anomaly system or other learning based system. Similarly, there is no discussion of whether the rates of intrusions or their relationship to one another is typical of the scenarios that such detectors might expect.

For systems using *a priori*, non parametric, rules for detecting intrusion manifestations, the training data provides a sanity check, but little more. If there are background manifestations that trigger the same rule as an identified intrusion in the training data, and the developer wishes to use the training data to guide development of his system he might attempt to refine the rules to be more discriminatory. The user could also change the way in which the system operates to make detections probabilistic, based on the relative frequencies of identified intrusion manifestations and background manifestations that trigger the same rule. As we will see later, the ROC analysis method is biased towards detection systems that use this kind of approach.

For systems that can be tuned to the mix of background and intrusions present in the training data, this bias may be inherent depending on whether the detection methods result in probabilistic recognitions of intrusions or whether internal thresholds are adjusted to achieve a similar effect. The problem with tuning the system to the data mix present in the training data is that transferring the system experience to the real world either requires demonstrating that the training mix is an accurate representation of real world data with respect to the techniques used by each system or it requires that accurate real world training data be available for each deployment environment. We claim that the former conditions have not been met and that the latter may not be possible. As far as we are aware, existing studies of network traffic patterns show a high degree of variability among sites as well as substantial changes with time at a given site. As we have noted earlier, unless the target environments, i.e. military installations, are atypical, it may be the case that there is no such thing as a "typical" traffic mix that is suitable for background data. If each deployment environment is characterized by a unique traffic mix and if the ability of an IDS to detect intrusions effectively depends on tuning it to match the mix under

controlled conditions, the problem may well be intractable. More work on traffic characterization and the effects of traffic variability on the IDSs is clearly needed.

If one views the corpora of training data as a form of benchmark against which present and future IDS systems might be evaluated, there is also a risk that systems might be optimized for the benchmark at the expense of normal case behavior. This is a well known problem in the software evaluation field.

4 The Taxonomy of Attacks

Kendall's thesis uses a taxonomy of attacks that was originally developed by Weber [15]. The taxonomy describes intrusions from an intruder centric viewpoint based loosely on a user objective. For the purposes of the evaluation, the attacks used were characterized as

1. Denial of Service,
2. Remote to user,
3. User to Superuser, or
4. Surveillance/Probing

and were further characterized by the mechanism used. The mechanisms were characterized as

m Masquerading (stolen password or forged IP address)
a Abuse of a feature
b Implementation bug
c System misconfiguration
s Social engineering

While this taxonomy describes the kinds of attacks that can be made on systems or networks, it is not useful in describing what an intrusion detection system might see. For example, in the denial of service category, we see attacks against the protocol stack, against protocol services, against the mail, web, and syslog services, and against the system process table. The effects range from machine and network slowdowns to machine crashes. From the standpoint of a network or host observer (i.e. most intrusion detection systems), the attack manifestations have almost nothing in common. From this, it can be seen that the taxonomy used in the evaluation offers very little support for developing an understanding of intrusions and their detection. We suggest that the taxonomy used is not particularly supportive of the stated objectives of the evaluation and that one or more of the potential taxonomies discussed in the following section could be more useful in guiding the process.

The attacker centric taxonomy poses an additional problem. By tying attacks to overt actions on the part of a putative attacker, it creates a highly unrealistic evaluation bias. The treatment of probes is a case in point. Not all probes are hostile. They are a standard way of attempting to initiate internet communication, but communication does not always occur even when the probed host

acknowledges that it provides the probed for service. As far as we have been able to tell, the 1998 background data does not contain this kind of benign probe activity, but the evaluation data contained at least one "attack" that consisted of a very small number of probes. We claim that, had the background data contained a typical mix of normal or benign probe data, these probes would have been distinguishable as attacks only if the intent of the prober were known. While this is possible in the evaluation context, it is generally not possible in the field.

4.1 An Alternative Taxonomy

Attacks could be classified based on the protocol layer and the particular protocol within the layer that they use as the vehicle for the attack. Under this approach, attacks such as "Land," "Ping of Death," and "Teardrop" are related because they never get out of the protocol stack. They are also similar in being detectable only by an external observer looking at the structure of the packets for the identifying characteristics. Smurf and UDPStorm attacks are even lower in the hierarchy because they affect the network and interface in the neighborhood of the victim. Also, they are detectable based on counting of packet occurrences which could be considered a lower level operation than examining packet structure. Attacks that involve altering the protocol stack state such as "SYNFlood" are higher since their detection either involves monitoring the state of the protocol stack internally, or modeling and tracking the state based on an external view. Attacks that require the protocol stack to deliver a message to an applications process (trusted or not) are still higher. Detecting such attacks requires either monitoring the messages within the host (between the stack and the application or within the application) or modeling the entire stack accurately, assembling messages externally and examining the interior data with respect to the view of the attacked application to determine the attack. Probes can take on a variety of forms, but are usually handled either within the stack (especially if the service sought is not supported) or via interaction with the application that supports the probed for service.

A strength of this taxonomic approach is that it leads to an understanding of what one must do to detect attacks. Within a particular higher level protocol or service this view may group attacks that exploit common vulnerabilities together, for example "Appache2" and "Back" exploit pathologies in the http specification while "phf" exploits a bug in the web server's implementation of CGI bin program handling.

Many other taxonomies are possible. The point is that the taxonomy must be constructed with two objectives in mind; describing the relevant universe and applying the description to gain insight into the problem at hand. Weber's taxonomy serves the first purpose fairly well, but fails to provide insights useful to understanding the detection of intrusions.

5 The Evaluation

The results of the evaluation and the way in which they have been presented by Lincoln Lab present a number of difficulties. We examine several of these, notably the problem of determining an appropriate "unit of analysis" and problems associated with the use of the ROC method of analysis. The unit of analysis problem arises whenever experimental results are reported as percentages. The evaluated IDS systems report detections which can be characterized as either correct, i.e. an attack was reported when one *was* present, or incorrect, i.e. a false alarm, an attack that was reported when one *was not* present. The ROC method requires both correct and incorrect detections to be reported as percentages of the possible cases in which the detection could have been made. In the case of the evaluation, successful detections can be reported as (number of attacks detected) / (number of attacks made), but no comparable denominator exists for reporting false alarms. The unit of analysis problem is well known in other fields [16] where it often results in ascribing more power than is appropriate to the results of certain statistical tests. While this is not the case here, the problem exists and its solution is a necessary prerequisite to performing meaningful comparisons among systems. For example, two sustems may raise the same number of false alarms and have different false alarm percentages if one bases its decisions on the examination of entire protocol sessions while the other examines individual packets.

ROC analysis is a powerful technique for evaluating detection systems, but there are a number of underlying assumptions that must be satisfied for the technique to be effective. It is not clear that these assumptions are or can be satisfied in the experimental context. In addition, ROC analysis is biased towards a classical detection approach not commonly used in IDS systems.

5.1 TCPdump Data and the Unit of Analysis Problem

The largest data set[5] made available to investigators for evaluating their systems consists of raw TCPdump data collected with a sniffer positioned on the network segment external to the Eyrie AFB router. This dataset should contain all the data generated inside the simulated base destined for the outside world and all the data generated outside the base destined for an inside location. Experience with TCPdump indicates that it can become overloaded and drop packets although the possibility of this is reduced by the apparently low data rates used. The thesis indicates that attacks were "verified" by hand and that this process was very labor intensive [6, Section 13.2.2], but it is unclear what verification means here.

Training data is accompanied by a list of the "sessions" that are present in the TCPdump data where a session is characterized by a starting time, duration, source, destination, and a protocol. If the session contained an attack, the list

[5] Solaris BSM audit data and file system dump data were also available. We have not looked at them.

identifies the attack. Examination of a sample of the TCPdump data indicates that it contains additional traffic, e.g. messages from ethernet hubs, that is not in the list.

The association of alarms with sessions is an instance of a more general unit of analysis problem. The question of an appropriate denominator for presenting the evaluation results is only superficially addressed. It may not be appropriate to use the same denominator for all systems and the choice of a denominator may vary from system to system or even from attack to attack within the same system. The appropriate unit of analysis is that body of information on which the system based its decision to raise or not raise an alarm. The denominator for the expression giving the percentage of true alarms is the number of cases when this decision point was reached and the body of data used to make the decision contained a manifestation of a real intrusion. Similarly, the appropriate denominator for false alarms is then the number of times that the system reached this decision point when the data on which the decision was based did not contain a manifestation of a real intrusion. These numbers are a function of the detection process and cannot be externally imposed unless the decision criteria are externally specified. Sessions may be the natural unit on which to base decisions in some systems and not for others and their use will bias the results when they are used as the unit of analysis where they are not appropriate.

The use of sessions as the unit of analysis presents other potential problems. Attacks are, of necessity, associated with a single session under this model, precluding the injection of coordinated attack behavior involving multiple sources and/or protocols. For example one could envision probes carried out from a large number of locations so that no single source address appears more than once. The session model seems to preclude this. Although the injected attacks are associated with sessions in the test data, nothing constrains the evaluated systems to use the session concept, and it is possible that alarms may be raised as a result of events contained in more than one session.

5.2 Scoring and the ROC

The Lincoln Lab team decided to use a technique known as the ROC[6] as the method for presenting their results and the use of this technique is claimed as one of the major contributions of their effort in the DISCEX paper [9, Section 2]. The ROC has its origin in radar signal detection techniques developed during World War II and was adopted by the psychological and psychophysical research communities during the early post war era [14]. Its adoption by the Lincoln Lab group is not surprising given that their background is in speech recognition (word spotting in particular). Much of the discussion that follows is due to Egan [3]. Signal detection theory was developed during the two decades following World War II to give an exact meaning, in a probabilistic sense, to the process of

[6] The term ROC originally stood for Receiver Operating Curve. Since the technique has been widely used to evaluate systems that do not have a recognizable receiver, ROC is commonly interpreted as Relative Operating Characteristic.

recognizing a wanted signal that has been degraded by noise. The methods took into account the relationship between the physical characteristics of the signal and the theoretically achievable performance of the observer. Later the concepts of signal detection theory were adapted to provide a basis for examining some problems in human perception. The basis for the ROC is given by Egan [3, P. 2]

> When the detection performance is imperfect, it is never assumed that the observer "detects the signal." Rather, it is assumed that the observer receives an input, and this input corresponds to, or is the equivalent of, the unique value of a likelihood ratio. Then, given other factors, such as the prior probability of signal existence, the observer makes the decision "Yes, the odds favor the event *signal plus noise*," or "No, the odds favor the event *noise alone*."

Egan goes on to note that signal detection theory consists of two parts, decision theory, which deals with the rules to be used in making decisions that satisfy a given goal, and distribution theory, dealing with the way in which the signals and noise are distributed. When the distributions are known (or can be assumed) the relationship between the distributions and possible performances is called ROC analysis.

A typical ROC curve is a plot on two axes as seen in Figure 1. The vertical axis measures the true positive rate of the system (i.e. the Bayesian detection rate or the probability of a recognition given that signal plus noise is present). The horizontal axis gives the false positive rate (i.e. the probability that an alarm is raised given that only noise is present). An evaluation of a system provides estimates of these probabilities as the percentage of accurate and inaccurate recognitions in a series of trials under fixed conditions. By fixed conditions here, we mean constant distributions of signal plus noise and noise.

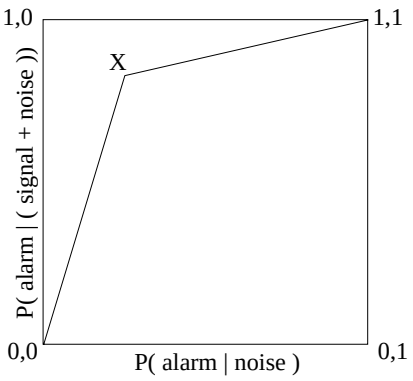

Fig. 1. A single point ROC

Note that there are two crucial aspects of the process. First, the observer receives an input and second, the observer makes a decision concerning that input. The observer thus controls the unit of analysis problem by defining the unit of analysis as the quantity of input on which a decision is made. Both positive and negative decisions must be recorded so that event counts for the denominators of the percentages used in ROC analysis will be available. Unless all the systems under evaluation are based on the same notion of an event on which a decision is to be made, choosing an arbitrary division in the input such as a packet or a session does not supply the necessary denominator.

Both parametric and non-parametric IDS detectors exist. Non parametric detectors have no provision for adjusting the sensitivity of the detection mecha-

nism to effect a tradeoff between detection rates and false alarm rates. Examples include signature systems in which the attack signature is matched or it isn't and finite state approaches that raise an alert only if the underlying automata reaches an accepting state. Parametric systems have adjustable thresholds or are able to assign probabilities to alerts based, e.g., on a priori knowledge of signal and noise distributions[7] or on quantifiable uncertainties in the detection process. The later is more likely to be a property of anomaly detectors, especially those based on population or individual statistical properties.

If the ROC is an appropriate mechanism for presenting the results of an IDS evaluation in which non parametric, binary, decisions are made, the curve will consist of a single point that expresses the true positive and false positive percentages for the entire evaluation. The justification for drawing lines from the (0,0) coordinate to the point and from the point to the (1,1) coordinate is counterintuitive, imposing a probabilistic model where none is present. Nonetheless, the lines are usually presented as shown, and we follow the tradition in our presentation. In the environment in which most IDS systems operate, the signal percentage is very small[8] requiring very low false positive rates for useful detection as discussed in a recent paper by Axelsson [1].

As far as we are able to tell, none of the IDSs under evaluation use a likelihood ratio estimator that considers both the signal distribution and the noise distribution as their decision criteria and little is known about the *in vitro* distributions of intrusions and background activity that would make this fruitful. Most of the systems use only signal plus noise characteristics (signature based systems) or only noise characteristics (anomaly detection systems). The issue of tuning systems that use *a priori* distributions implicitly by learning or training procedures has been discussed above.

5.3 Errors per unit time

The DISCEX paper uses a non-standard variation of the ROC presentation [9, Figure 4] that labels the horizontal axis with false alarms per day rather than

[7] Suppose that we know that 0.1% of the probes for finger service are precursors to an attack, while 99.9% are benign. How should we deal with this situation? Assuming that we can detect the probe 100% of the time, we can raise an alert with a 0.1% probability that it represents an attack every time a finger probe occurs. The ROC method requires us to classify each alert as either a successful detection or as a false alarm, but allows us to vary the threshold for the decision. As we vary the threshold from 0.0% to 0.1% the curve will show 100% detection rate and 100% false alarm rate since both attacks and false alarms are assigned a probability above the threshold. Above a threshold of 0.1%, both the detection rate and false alarm rate drop to 0.0%. The problem here is that there is very little signal (attack instances) and a lot of noise (benign use of the finger service). In the absence of other factors that allow us to refine the probability assigned to a given probe, the *a priori* distribution does not help and we are left with two choices; ignore finger probes (missing a small number of attack indicators) or raise a large number of false alarms.

[8] This assumes that a small unit of analysis is chosen for computing the denominator of the false alarm rate.

percent false alarms. A search of the traditional ROC literature [14, 3] shows no mention of this formulation. It does appear, without comment or justification in the word spotting literature where it is usually [7], but not always [5], referred to as a ROC curve.

Many of the corpora used for word spotting evaluations come from NIST, but researchers at NIST disavow the origin of the formulation saying that it was already in use when they entered the field. According to Alvin Martin of NIST, the earliest use of the formulation of which he is aware appeared in technical reports from Verbex Corporation in the late 1970s [10]. We were able to locate Stephen L. Moshier, one of the founders of Verbex and an author of some of the reports mentioned by Martin. He reported [11] that

> The military customer perceived that the user of a word spotter could cope with alarms (true or false) happening at a certain average rate but would becomes overloaded at a higher rate. So that is a model of the user, not a model of the incoming voice signals.

One of the more powerful features of the ROC analysis is its ability to abstract away certain experimental variables such as the rates at which detections are performed. The primary factors that influence ROC results are the detector characteristics and the distributions of signals and noise. If the latter are realistic, the ROC presentation of the detector characteristics should have good predictive power for detector performance in similar environments.

The pseudo-ROC, as we choose to call word spotting form, breaks these abstractions. By using incomparable units on the two axes, the results are strongly influenced by factors, such as data rate, that ought to be irrelevant. The form shown in the DISCEX paper is misleading for a number of reasons, notably because of its failure to present the relevant information. Using the data set as provided for the evaluation, but reassigning values to the time stamps attached to the data items, the false alarm rate per unit time can be manipulated to any degree desired by varying the total duration represented by the dataset[9]. At the very least, the pseudo-ROCs presented by Lincoln Lab [9, Figure 4] should be labeled with the data rate on which the false alarm axis is based. This is especially true given that the data rates used in the evaluation appear to be unrealistically low. Using the evaluated systems on data streams with megabit rates might result in a ten to hundredfold increase in the false alarm rate when reported per unit time.

[9] Changing the timestamps so as to give the appearance that a five day dataset represented a single day would raise the false alarms per day by a factor of five. Similarly, increasing the generated background traffic from the average of 10Kb/s – 50Kb/s used in the evaluation to an average in the 50Kb/s – 250Kb/s rate should have the same effect.

6 Conclusions

The Lincoln Lab evaluation is a major and impressive undertaking, but its benefits seem to be far out of proportion with its costs and impacts on research programs. It is not clear that the results of the evaluation predict deployed performance. Reducing the performance of these systems to a single number or to a small group of numbers is not particularly useful to the investigators since the numbers have no explanatory power. While detection and false alarm rates are important at a gross level and might be a basis for comparing commercial products, the research community would benefit from an evaluation approach that would provide constructive advice for improvement.

It is hoped that this critique will either lead to a rethinking of the evaluation process and a recreation of it in a form that will help IDS development move forward. If the evaluation process cannot be modified so that it makes a substantial contribution to the improvement of the IDS state of the art, it would be better to abandon the evaluations for the present. Indeed, it appears that DARPA is currently rethinking its approach to evaluation in response to this and the other criticism[10] that it has received from other members of the IDS community.

7 Acknowledgments

I want to thank many members of the intrusion detection community for helpful comments on earlier versions of this paper. Some of them wish to remain anonymous and I respect those wishes. I also want to thank one of the anonymous reviewers for RAID 2000 who also provided numerous insightful and valuable comments. Roy Maxion of CMU has been a constant source of inspiration and support. Jim Jenkins of the Psychology Department of the University of South Florida provided help in tracking down the origins of the pseudo ROC curves as did Alvin Martin at NIST. Martin's efforts led me to Stephen L. Moshier, one of the founders of Verbex who was, in part, responsible for its introduction into the word spotting arena. Stefan Axelsson provided valuable critiques of earlier versions of this paper. Julia Allen and Tom Longstaff of CERT have been most supportive and without their efforts, the paper would not exist. Jim Binkley of Portland State provided data on typical network traffic levels at the PSU School of Engineering.

References

[1] Stefan Axelsson. The base-rate fallacy and its implications for the difficulty of intrusion detection. In *6th ACM Conference on Computer and Communications Security*, pages 1–7, 1999.

[10] Criticisms similar to those presented in this paper were made in a presentation given by Brad Wood of Sandia National Laboratory at the DISCEX 2000 conference in January, 2000. Unfortunately, there is no corresponding paper in the DISCEX proceedings.

[2] Steven M Bellovin. Packets found on an internet. *Computer Communications Review*, 23(3):26–31, July 1993.

[3] James P. Egan. *Signal detection Theory and ROC Analysis.* Academic Press, 1975.

[4] Isaac Graf et al. Results of DARPA 1998 offline intrusion detection evaluation. Presentation at MIT Lincoln Laboratory PI Meeting (available at http://ideval.ll.mit.edu/results-html-dir/, 15 December 1998.

[5] D. A. James and S. J. Young. A fast lattice-based approach to vocabulary independent wordspotting. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 337–380, 1994.

[6] Kristopher Kendall. A database of computer attacks for the evaluation of intrusion detection systems. BS/MS thesis, Massachusetts Institute of Technology, June 1999.

[7] Richard P. Lippmann, Eric I. Chang, and Charles R. Jankowski. Wordspotter training using figure-of-merit back propagation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 385–388, 1994.

[8] Richard P. Lippmann et al. MIT Lincoln Laboratory offline component of DARPA 1998 intrusion detection evaluation. Presentation at MIT Lincoln Laboratory PI Meeting (available at http://ideval.ll.mit.edu/intro-html-dir/, 14 December 1998.

[9] Richard P. Lippmann et al. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In *DISCEX 2000.* IEEE Computer Society Press, January 2000.

[10] Alvin Martin. Personal communications, January 2000.

[11] Stephen L. Moshier. Personal communications, January 2000.

[12] Vern Paxson. Bro: A system for detecting network intruders in real–time. *Computer Networks*, 31(23-24):2435–2463, December 1999.

[13] Stacy J. Prowell, Carmen J. Trammell, Richard C. Linger, and Jesse H. Poore. *Cleanroom Software Engineering: Technology and Process.* Addison–Wesley, Reading, Mass., 1998.

[14] John A. Swets. Measuring the accuracy of diagnostic systems. *Science*, 24(48):1285–1293, 3 June 1988.

[15] Daniel Weber. A taxonomy of computer intrusions. MS thesis, Massachusetts Institute of Technology, 1998.

[16] Q. E. Whiting-O'Keefe, Curtis Henke, and Donald W. Simborg. Choosing the correct unit of analysis in medical care experiments. *Medical Care*, 22(12):1101–1114, December 1984.