Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory

JOHN McHUGH
Carnegie Mellon University

In 1998 and again in 1999, the Lincoln Laboratory of MIT conducted a comparative evaluation of intrusion detection systems (IDSs) developed under DARPA funding. While this evaluation represents a significant and monumental undertaking, there are a number of issues associated with its design and execution that remain unsettled. Some methodologies used in the evaluation are questionable and may have biased its results. One problem 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 appropriateness of the evaluation techniques used needs further investigation. The purpose of this article 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 article points out might well be resolved if the evaluators were to publish a detailed description of their procedures and the rationale that led to their adoption, but other problems would clearly remain.

Categories and Subject Descriptors: K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Invasive software* (e.g., viruses, worms, Trojan horses)

General Terms: Security

Additional Key Words and Phrases: Computer security, intrusion detection, receiver operating curves (ROC), software evaluation

1. INTRODUCTION

The most comprehensive evaluation of research on 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

This work was sponsored by the U.S. Department of Defense.
Author's address: CERT® Coordination Center, Software Engineering Institute, Carnegie Mellon University, 4500 Fifth Ave., Pittsburgh, PA 15213.

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.

1.1 What is this Article and Why was it Written?

It is important to note that the present article 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. The critic need not provide solutions to the problems raised although it is to be hoped that the community as a whole will be able to address them in connection with future efforts. This article is offered in the spirit of Harvey Einbinder [1964], whose book criticizing the Encyclopedia Britannica was based on the premise that, flawed as it was, the Britannica was the only work of its kind worthy of the critical effort.

We note that criticism and review[1] are well-established practices in many scientific fields as well as in the social sciences and humanities. In many scientific areas, results are not accepted until they have been replicated by independent investigators, placing a premium on the completeness of the original published description of the result. In the humanities, careers may be based on criticism rather than on original contributions. In computer science, replication, criticism, and review are rare. One of the few efforts at review of which we are aware uncovered substantial problems with the reviewed project [NASA 1985; Research Triangle Institute 1984] and had a lasting impact on the area.

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.

For the most part, we concentrate on the 1998 evaluation, only briefly discussing the 1999 evaluation that was under way when the original version of this article was written. Many of the problems noted in 1998 remain in 1999 although some specific improvements, along with a few additional problems, are noted. The scope of the article is limited. We have primarily considered issues associated with one of the datasets provided by Lincoln, the network data recorded at the boundary of the monitored network, and have not examined in detail issues associated with either the Solaris BSM (Basic Security Module) audit data or the file system snapshot

[1] By review, we mean peer review of the form in which the reviewers conduct an in-depth examination of a completed or ongoing project. This is in contrast to publication review, which is common but seldom able to pose the kinds of questions that lead to deep insights.

data. The real-time evaluation of IDSs conducted by AFRL (Air Force Research Laboratory) [Durst et al. 1999] has not been considered. We have not attempted to evaluate the overall DARPA research program of which the Lincoln evaluation is a part, although a review of the program, similar to that carried out for an earlier ARPA program in speech understanding [Klatt 1977], might be useful.

While the present work might be criticized for not giving more weight to restricted information sources that may alleviate some of the problems noted, we argue that relying on such sources would be bad scholarship. Large scale experiments of this type are likely to have an influence on the field that outlasts the involvement of the present participants. We feel strongly that the public record should contain sufficient information to permit replication of the results and to permit other investigators to understand the rationale behind many of the decisions made by the investigators. Relying on folklore, unpublished presentations, and personal communications should be a last resort in scholarly publications.

1.2 Overview

After a brief look at the (limited) prior attempts to evaluate IDSs, the remainder of the article is divided into several sections, each addressing some critical aspects of the problem. First, we describe the overall experiment to provide an appropriate framework for further discussion.

We begin 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 receiver operating curve (ROC) 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. Finally, the article concludes with suggestions for performing future evaluations that might overcome the deficiencies of the Lincoln Lab work. Because of the difficulties involved in performing experimentation of this type, it is suggested that a more cautious and systematic approach be adopted.

2. ANTECEDENT AND CONCURRENT EFFORTS

There is relatively little prior work in the field of evaluating intrusion systems. The work of Puketza and others at the University of California at

Davis [Puketza et al. 1997; 1996] is the only reported work that clearly predates the Lincoln effort. These papers describe a methodology and software platform for the purpose of testing intrusion detection systems. The methodology consists of using scripts to generate both background traffic and intrusions with provisions for multiple interleaved streams of activity. These provide a (more or less) repeatable environment in which real-time tests of an intrusion detection system can be performed. Only a single IDS, the network security monitor (NSM) [Heberlein et al. 1990], seems to have been tested, and the tests reported could not be seen as any sort of a systematic evaluation. The earlier work [Puketza et al. 1996], dating from 1993, reports the ability of NSM to detect several simple intrusions, both in isolation and in the presence of stresses. One form of stress is induced by system loading. Load is measured in terms of the number of concurrent jobs running on the host supporting NSM and NSM is reported to drop packets under high load averages (42% byte stream loss at a load average of about 14.5). Other forms of stress include background noise (nonintrusive network activity), session volume (the number of commands issued during an intrusive session), and intensity (number of concurrent sessions on the link being monitored). No experimental results are given for these forms of stress. In their later paper [Puketza et al. 1997], the Davis group concentrates on the ability of the test facility to support factoring of sequential attacks into a number of concurrent or overlapping sessions. They report that NSM assigns lower scores to some attacks that have been factored, noting that NSM's independent evaluation of individual network connections may allow attacks to be hidden in this way.

In 1998, while the Lincoln group was developing and carrying out its test methodology, a group at the IBM Research Division in Zurich issued a technical report [Debar et al. 1998] describing another experimental facility for comparing IDSs. Like the previous work, the Zurich group reports on the design and implementation of a real-time testbed. The Zurich testbed consists of several client machines and several server machines, under the control of a workstation used as the workbench controller. The report discusses a number of issues associated with the generation of suitable background traffic, noting the difficulties associated with alternatives including developing accurate models of user and server behavior, using test suites designed by operating system developers to exercise server behavior, and using recorded "live" data. The authors tend to favor the test suite approach, but recognize that it may bias results with respect to false alarms. Attacks are obtained from an internally maintained vulnerability database that makes hundreds of attack scripts available although only a few are applicable to the initial workbench configuration which only supports FTP services. The article describes several of the attacks on FTP. Considerable attention is given to the controller component of the workbench which allows the systems under evaluation to be configured and administered from a single console. The controller also allows the results from several IDSs to be compared. Unfortunately, the report does not present any results obtained from the workbench. Several observations from the article are worth noting. The first is that "Generating realistic, normal behavior is a time-consuming task when working in a heterogeneous environment." Another is that "it is worth noting that the definition of a set of criteria to evaluate the effectiveness of existing IDSs remains an open issue which we will have to address if we want to perform pertinent comparative studies."

Recent work at Zurich [Alessandri 2000] addresses the potential of IDS systems to detect certain classes of intrusions using an analysis of design principles rather than an evaluation based on an actual implementation. The article develops a technique for describing activities that may be either intrusive or benign and describes the features that an IDS must have in order to successfully detect the intrusive activities while rejecting benign ones. Although this work is in its early stages, the author claims that it is generic and easily extends to a wide variety of intrusive activities, including those for which signatures have not yet been developed.

Reviews and comparisons of commercial IDS systems appear from time to time, usually at the Web sites of on-line publications.[2] The reviews are generally superficial and lack details concerning the test methods used. The rapid rate at which new products are introduced and existing products modified gives these reviews a limited window of utility. This is discussed in a recent SEI technical report [Allen et al. 2000].

3. THE EVALUATION

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 article. The most detailed descriptions of the 1998 work available at the present time are Kristopher Kendall's BS/MS dissertation [Kendall 1999] and a paper by Lippmann et al. [2000] in the proceedings of the DARPA-sponsored DISCEX conference. 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 the Recent Advances in Intrusion Detection Workshop (RAID 99) at Purdue University in September of 1999. Presentations [Lippmann et al. 1988; Graf et al. 1998] similar to ones given at those meetings also appear at the Lincoln Lab experiment site, http://ideval.ll.mit.edu.[3] A paper describing the 1999 evaluation results[Lippmann et al. 2000] was presented at RAID 2000 and appears in its proceedings. The reader is referred to these sources for a description of the evaluations, their goals, objectives, and results. Note that references to the systems that were evaluated are contained in many of the publications cited above, notably the DISCEX and RAID 2000.

[2] Unfortunately, these reports trend to be transient in nature. One such source cited by a reviewer of the present work was invalid by the time the review reached the author. Only two of the three sites mentioned in the SEI report were still valid as of November 2000.

[3] This site is password protected. Contact <intrusion@sst.ll.mit.edu> for information concerning access.

According to the DISCEX article [Lippmann et al. 2000], "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 corpus 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 4, we discuss the adequacy of the dataset used during the evaluation, suggesting that, at best, its suitability for this purpose has not been demonstrated and that, at worst, it contains biases that may be reflected in the performance of the evaluated systems. 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 6.3) 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 is discussed further in the appropriate section.

Many of the systems evaluated by the Lincoln Lab group have been described in a variety of technical publications. In some cases, these articles mention the evaluation; however, none that we are aware of and that appear as part of the public record provide significant additional information concerning the evaluation process and we do not cite them here.

Each system under test was evaluated by its developers who adapted the data as necessary to fit the system in question, as described in the DISCEX article [Lippmann et al. 2000]. [4] 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 article.

Lincoln Lab implies [Lippmann et al. 2000, Fig. 4] that the research systems it evaluated are demonstrably superior to a baseline system said to be typical of commercial IDSs [Lippmann et al. 2000, Sect. 7]. This claim is apparently based on an assumption that all commercial products use a very naive concrete string matching algorithm rather than on the actual evaluation of commercial systems.[5] While there is no evidence that commercial

[4] Some of the systems under test only looked for a limited subset of the attacks or considered only a subset of the protocols or connections represented in the data set. This probably means that the data presented to these systems was extracted or filtered in some way so that the number of cases considered was much smaller for these systems than it was for others. This should be reflected in the presentation of false alarm data, especially when the false alarm data is characterized as false alarms per unit time, but there is no mention of these factors in the available papers or presentations.

[5] Junkawitsch et al. [1998] and Kendall [1999] are referenced in the DISCEX paper [Lippmann et al. 2000] as descriptions of what are said to be typical commercial and government systems. A casual inspection of the references indicates that both of these systems are substantially more sophisticated than the posited baseline.

systems are very good, nothing useful is contributed by setting up a weak straw man and then demolishing him. Including one or more commercial systems in the evaluation, and perhaps using them to debug and validate the test data before distributing them to the research community for evaluating their systems could have provided an objective target for comparison as well.

4. THE EVALUATION DATA

For reasons having to do with privacy and the sensitivity of actual intrusion data, the experimenters chose to synthesize both the background and the attack data used during the evaluation. There are problems with both components, which are discussed separately below. 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 corpus 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.

As noted by Lincoln Lab in the DISCEX article, such corpora have been used in other areas, including the wordspotting community of which the Lincoln Lab group is a part. Corpora such as these require careful construction and validation if their content and structure are not to bias the systems that are developed using them for test data. Stephen L. Moshier, an early researcher in the wordspotting field offers a word of caution in a personal communication concerning such corpora:

> "I don't know how well the results from a test corpus actually predicted operational performance, but anyway the statistics were used mainly to compare different techniques. For that the best one could do was a nonparametric test using the total numbers of events, after applying the same stimuli to the different analysis techniques. I observed that these results were generally invalid, because changing to a different test corpus changed the experimental outcome. The results stabilized only after greatly increasing the size of the experimental database."

The corpus generated by Lincoln is unique in the Intrusion Detection arena and, as such, is the only substantial body of data that can be used for repeatable comparisons of IDS systems. At the same time, it may suffer from problems such as those noted above and may not provide results that reflect field performance. It appears to be used by researchers who were not part of the DARPA evaluation who should be aware of both its strengths and limitations.

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 data generation facility generates the two components simultaneously and the network traffic that is captured for the evaluation consists of a mix of generated background data, embedded attacks, and responses from systems that are part of the test framework. If we view background data and responses as noise and attack data and responses as signal, the IDS problem can be characterized as one of detecting a signal in the presence of noise. As we show in Section 6.3, the evaluation produces two measures: one that is primarily a function of the noise, the other primarily a function of the signal (albeit embedded in the 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.

4.1 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 mail messages match those observed. The DISCEX paper [Lippmann et al. 2000, Sect. 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. As far as we can tell, the measures of similarity involve content for SMTP sessions and frequency of use for various UNIX utilities, but are not concerned with the content of the utility interactions. This article indicates that much of the material used in the simulation as payloads for file and Web transfers is obtained from or similar to that seen in public domain sources.

As far as can be determined from the record, neither analytical nor experimental validation of the background data's false alarm characteristics were undertaken. No detailed characterization of the data appears in the record and, more important, 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 the operating point that results from plotting the percentage of intrusions detected against the percentage of false alarms emitted for a given decision criterion. 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 [1993] reported on anomalous packets some years ago. Observations by Paxson [1999] 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 such as storms of FIN and RST packets, fragmented packets with the don't fragment

Table I. Data Rates for Selected Training Days

		File Size (Kbytes)		Ratio	Data Rate
Week	Day	Comp'd	Uncomp'd	(%)	Kbits/sec.
3	Tues.	39,192	87,007	45	8.8
4	Tues.	388,944	508,535	76	51.4
5	Tues.	118,314	336,265	35	34.0
5	Wed.	44,121	98,066	45	9.9

flag set, legitimate tiny fragments, and data that differs from the original in retransmission. In particular, poor implementations of various network protocols are common in the PC world and 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 and might be considered as things with which an IDS ought to be concerned. Spontaneous packet floods should definitely be treated as suspicious until the source is identified. As far as we can tell from the public record, the inclusion of such packets was not considered in the generation of background traffic. The limited inclusion of fragmented packets as suggested by Ptacek [Ptacek and Newsham 1998] in the 1999 evaluation [Das 2000] does not address this issue.

None of the sources that we have examined contains 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.[6] We have performed a superficial examination of several days of the TCPdump training data. The results are presented in Table I. Each of the days represents a 22 hour period of training data. The rate is calculated from the uncompressed file size and is slightly high due to the inclusion of the TCPdump header information.

We chose two days with relatively small compressed file sizes and two of the larger ones (based on the data given on the Lincoln Lab download pages. Most of the days are represented by compressed file sizes of under 100 Mb. We suspect that the average data rates that these files represent are on the order of a few tens of kilobits per second, a rate that seems to be surprisingly low for an installation with "hundreds" of workstations. In contrast, MRTG data from Portland State University shows that the total traffic into and out of the single building housing part of the Computer Science and Electrical Engineering departments averages about 500 kilobits per second for about 100 workstations and traffic for the Portland State Engineering School with about 1000 workstations and servers is on

[6] This factor may not be relevant for an off-line evaluation such as the one reported here, but we would expect the evaluators to take the data timing into account in performing their evaluations.

the order of 5 megabits per second in each direction[7] or a total of about 10 megabits per second. Paxson [1999] indicates sustained data rates in excess of 30 megabits per second on the FDDI link monitored by the Bro IDS. Since one would expect false alarm rates to be proportional to the background traffic rate for a given mix, the false alarm rates reported by Lincoln Lab may need to be adjusted.

4.2 Attack Data

Similar arguments can be made about the synthetic attack data. Most of the attacks used were implemented via adaptations of 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 the background noise. Kendall [1999, Sect. 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 [1999, Table 12.1] gives a tabulation of the attack data. 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[8] is highly unlikely to reflect performance in the field, but neither the dissertation nor any of the presentations discusses the issue. While the Lincoln Lab group downplayed the importance of the aggregate evaluation rate (and thus the significance of the attack mix) in discussions at the Phoenix PI meeting, it was clear that the DARPA sponsor wanted a single metric for comparing systems and approaches.

4.3 Eyrie AFB

The simulated data is said to represent the traffic to and from a typical Air Force Base, referred to as Eyrie AFB (eyrie.af.mil 172.16....). The dissertation [Kendall 1999, Figure 3–11] and the information available from the Lincoln Lab Web site seem to differ on the details of the configuration. According to the network diagram associated with the week 1 test data on the Web, the base complement seems to consist of four real machines: two SPARC Ultras running Solaris 2.5.1 (Locke and Pascal), a 486 machine running Linux (Marx), a SPARC running SunOS 4.1.4 (Zeno), and a P2 (Hobbes) which serves to generate all the base's background traffic. The dissertation indicates three fixed targets, running Solaris, SunOS, and Linux, an internal sniffer (running Solaris?), and an internal traffic generator plus an additional Linux target that can take on a variety of IP addresses. The host list for Eyrie for week 1 lists the fixed targets (Redhat

[7] Current statistics are available from http://network.cat.pdx.edu.

[8] Figure 6 of the DISCEX paper [Lippmann et al. 2000] compares the three best systems in terms of an aggregate detection rate measured over all 120 attacks in the test data.

4.0, Kernel 2.0.27 on Marx) plus a variety of other hosts running a mix of Solaris 2.5.1, SunOS 4.1.4, Linux Redhat 5.0 (Kernel 2.0.32), Windows 95, Windows 3.1, and Macintosh operating systems. The latter apparently are virtualized by Hobbes which runs Linux Redhat 5.0 (Kernel 2.0.32). The host list for week 3 (the last week for which such a list is available) lists additional hosts linux1–linux10, which are probably implemented on the additional Linux target mentioned in the thesis, but this is not clear since the network diagram included in the week 3 documentation does not show this host. The DISCEX article [Lippmann et al. 2000, Sect. 3] is less specific, but apparently, the test environment evolved during the first three weeks of training data generation and the diagrams at the Web site are correct and apply until superceded by a subsequent description.

The dissertation contains a list of the attacks [Kendall 1999, Appendix A] from the test phase of the evaluation. The vast majority of the attacks (45) target Pascal, 28 target Marx, 12 target Zeno, 10 target one of the virtual Linux machines, 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.[9] The skewed nature of the attack distribution may represent a bias that affects the results of 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 were 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. We do not accuse any participants of doing this, we only note that it is possible and very easy to do.

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 in Kendall [1999, Sect. 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.

[9] Probes can be part of normal internet traffic, indicating that a legitimate corresponding host may be attempting to determine whether a service needed for communication is available. If a probe gets a response and no subsequent communication is attempted, one may be able to infer hostile intent. In the absence of a reply, no such inference is possible.

4.4 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. Data validation is a well-known problem in the testing of software and the evaluation is effectively an exercise in software reliability estimation. The record produced by the Lincoln Lab evaluators does not show that the test environment does not systematically bias the evaluation results.

4.5 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 6.1, 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 a corpus 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 detectors might expect.

For systems using *a priori* 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 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 show 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 the same 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.

5. THE TAXONOMY OF ATTACKS

Kendall's thesis uses a taxonomy of attacks that was originally developed by Weber [1998]. 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 Lincoln Lab 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. Under the guise of creating stealthy attacks, patterns that appear naturally in background traffic can be inserted and labeled attacks.

5.1 Alternative Taxonomies

Developing alternative taxonomies for intrusions is beyond the scope of this article but useful taxonomies might result from the following approaches.

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 kind of taxonomy, 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 same reason. 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. Probes are lower still. 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.

A strength of this taxonomic approach is that it leads to an understanding of what one must do to detect attacks, for example, on httpd. 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. This view could lead to easily specialized, lightweight detectors applied close to vulnerable components. It is a small step from detection viewed this way to intervention and, not surprisingly, to concepts such as wrappers.

Another obvious approach is to classify attacks based on whether a completed protocol handshake is necessary to carry out the attack. This separates attacks into the class that admits a spoofed source address and those that require the attacker to reveal an immediate location. Basing a taxonomy on the severity of attacks or their potential for inflicting real damage (as seen from the viewpoint of those deploying the attacked system) might better suit the needs of the sponsors of the evaluation.

Many other taxonomies are possible. A recent paper by Axelsson [2000a] presents an overview of IDS approaches with several taxonomic classifications that provide insight into the detection problem. 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.

5.2 When Is It an Attack?

Because it is attacker-centric in its viewpoint, intent to attack is implicit in Weber's taxonomy. This leads to the simplistic scoring mechanisms discussed below. Unfortunately, intent is not easy to discern from the detector's viewpoint. We note that many of the attacks described by Kendall take advantage of bugs in the software of the attacked system while others can be viewed as pathological use cases in the normal spectrum of usage. For example, probes and probe responses are a normal feature of the attacked systems. They are provided to enable peer systems to negotiate methods for carrying out a common mission. From a detector point of view, probing may be normal or may be a precursor to some other activity. In the latter case, it may not be possible to confirm the intent of the prober (and thus recognize an attack) until some more overtly malicious act takes place that could be linked to the probe. Of course, it is possible to define probing above an arbitrary threshold as an attack with probing below that threshold viewed as benign. If such a definition is adopted, it is not a false negative from the standpoint of the detector to fail to declare an attack for probe activity below the threshold, even if the prober might consider the probing an attack. For the purposes of performing an evaluation under such conditions, the evaluator and evaluated need to agree on their definitions. There is no indication that this has been done.

Similarly, a packet that causes a buffer overflow is not necessarily an attack, although a packet whose content is crafted in such a way as to cause execution of specific code probably is. Buffer overflows generated by attack tools are typically designed to give the attacker access to the attacked system. As such, they will have a structure that is recognizable as machine code for the attacked system. If the attacker's objective is to deny others the use of the attacked service, the buffer contents are not constrained except in terms of size; examination of messages that cause service failures may or may not provide evidence of an intent to attack. If we assume that all messages that can provoke the same overflow are similar attacks, our taxonomy should not separate them into widely distinct classes as is the case with Weber's taxonomy.

6. THE EVALUATION

The results of the evaluation and the way in which they have been presented by Lincoln Lab suggest 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 is well known in other fields [Whiting-O’Keefe et al. 1984] 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. 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 certain styles of detection that may not be used in all IDS systems.

6.1 TCPdump Data and the Unit of Analysis Problem

The largest dataset 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. No mention of this possibility has been made in the thesis or available presentations and it is unclear from the public record as to whether there was any examination of the throughput of the sniffer to determine if it was adequate for the peak traffic rates seen or of the data to check for errors or omissions. Given the low average rates seen, dropped packets due to *TCPdump* are probably unlikely unless very high peak rates are present in some of the data. The thesis indicates that attacks were “verified” by hand and that this process was very labor intensive [Kendall 1999, Sect. 13.2.2], but it is unclear what the verification process was.

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 identifies the attack. Examination of a sample of the *TCPdump* data indicates that it contains additional traffic, for example, messages originating with the Ethernet hubs, that are not on the list. Presumably, this is an indication that such sessions will not appear as part of an attack and omitting a whole class of data from the session list may convey unintended information about the evaluation data to the participants.

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. If we want to talk about percentages, we need to determine an appropriate denominator for our numbers. 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.

Consider the following possibilities:

(1) A very simple system looks solely for matches of patterns that may indicate an attack within the individual packets, but only those associated with specific protocols. No state is maintained between packets. In this case, the appropriate unit of analysis is the packet.

(2) A more complex system models intrusions by treating packets associated with a given run of a given protocol as input symbols to a finite state recognizer. If the recognizer enters an accepting state before the end of the protocol run, an intrusion is signaled. In this case, the protocol run (session) is the appropriate unit. If not all protocols can contain intrusions, the system may make negative decisions on individual packets reasoning that protocol Z cannot take part in an intrusion since there are no initial states associated with the start of a session of it. In this case, the denominator is the sum of the number of sessions of potential intrusion-detecting protocols plus the number of packets from other protocols.[10]

[10] We leave the consequences of this strange hybrid to the reader. Consider the case in which we configure a sensor similar to *TCPdump* to filter out all protocols for which no intrusion recognizer exists. This will make the denominator for false alarms much smaller. Is this legitimate? The documentation provided with the 1998 sample data contains the following:

> "Sessions in list files will not include all TCP/IP connections that occur in a simulation, but only those connections that must be scored. Some services and sessions that are not involved in attacks will be ignored."

It is not clear whether the unlisted connection sessions are included in the denominator used to compute the percentage of false alarms.

While the identification of sessions in the training data is understandable, carrying the concept over into the test data would seem to provide the experimenters with information that would not be available in a real environment. There is no evidence that this was used to advantage by the researchers. The use of sessions as the unit of analysis presents other potential problems. For scoring purposes, attacks are, of necessity, associated with a single session under this model. The scoring rules used appear to preclude delayed labeling of earlier sessions when a multisession attack is recognized by its final session. Associating lower probability alarms with precursor session activities that may or may not be part of an attack may raise the false alarm rate unnecessarily. The requirement to score one session before examining data from subsequent sessions appears to constrain detection strategies. The possibility of multisession attacks could also result in a slightly optimistic false alarm rate since the denominator would be too large if a false alarm is reported that is based on multisession data.

6.2 Audit and Dump Data

BSM audit data from a Solaris host (Pascal) inside the AFB was also provided to investigators. We have not looked at this data in detail, but are concerned that there is no discussion of the adequacy of the audit configuration used[11] or of the ability of BSM used alone to provide adequate data for intrusion detection purposes. It is our impression that the "out of the box" audit capabilities of Solaris leave something to be desired and that substantial improvements or additions are necessary to ensure an adequate audit trail of which data generated by BSM is only one component. The CERT security improvement modules "Preparing to Detect Signs of Intrusion" and "Detecting Signs of Intrusion" [CERT Coordination Center 2000] and their associated practices and implementations outline a number of procedures for collecting and using audit data. Similar information is available from the SANS Institute [Pomeranz 1999]. BSM data is mentioned in passing in the CERT implementations and not at all in the SANS document. This is suspicious to say the least and casts doubt on the choice of BSM data as the primary host-based intrusion data source. The fact that some systems performed quite well using this data mitigates some of this criticism, but the question of whether they could have performed better with other or additional sources of host based data remains.

In addition, file system dump data (either as raw data or as signatures) was provided to permit integrity checkers to look for intrusion residue. We have not examined this data.

Note that the unit of analysis problem also exists for audit and dump data. Depending on exactly what is audited, it may be difficult or easy to identify the same sessions in audit data that were identified in the

[11] The audit configuration is described as part of the sample data available from the Lincoln Lab Web site, but the information is presented as a set of configuration files, without a supporting analysis.

TCPdump data. Audit data has the potential to identify items that do not appear in the externally observed *TCPdump* data. Protocol runs or sessions with internal machines can also show up in the audit data as can console logins and the like. These factors need to be taken into account in determining an appropriate unit of analysis for audit data. For file dump data, the appropriate unit of analysis is probably defined by the examined entities, that is, files and directories. The commercial version of Tripwire, for example, includes this number [Tripwire, Inc. 2000, Appendix B, p. 112] in its detailed reports.

6.3 Scoring and the ROC

The Lincoln Lab team decided to use a technique variously known as the *Receiver Operating Curve*, *Relative Operating Characteristic*, or *ROC* 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 [Lippmann et al. 2000, Sect. 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 [Swets 1988; Swets and Pickett 1982]. Its adoption by the Lincoln Lab group is not surprising given that their background is in speech recognition (wordspotting in particular) and not in computer security or intrusion detection. Much of the discussion that follows is due to Egan [1975]. 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 recognizing a wanted (or useful) 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. Shortly after its inception, 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 in the following quote from Egan [1975, 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, which deals 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 best called ROC analysis.

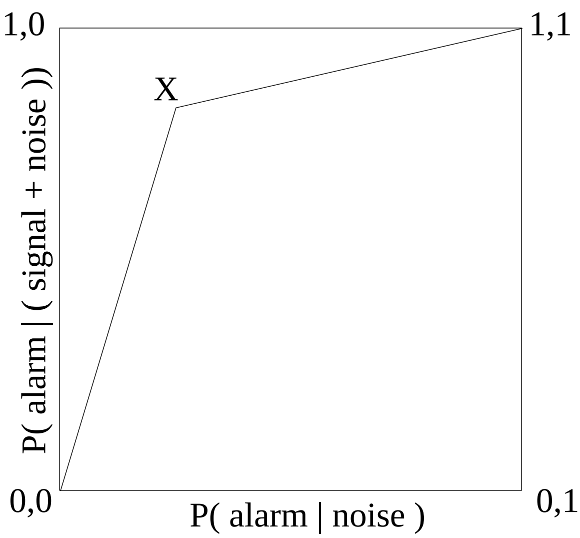

Fig. 1. A typical ROC curve resulting from a single evaluation.

A typical ROC curve is a plot on two axes as seen in Figure 1.[12] 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 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.

Note the crucial aspects of the process: First, the observer receives an input, and second, the user makes a decision concerning that input. Thus, the observer controls the unit of analysis problem by defining the unit of analysis as the quantity of input on which a decision is made. In order to provide appropriate denominators for the percentages used in ROC analysis, both positive and negative decisions must be recorded to provide event counts from which denominators can be computed.

The classical detection systems for which the ROC was developed use detectors that consider both the noise and signal plus noise distributions and make a decision that classifies the observation into one or the other based on the distributions and prior experience. As far as we are able to tell, none of the IDSs under evaluation use a likelihood ratio estimator that considers both the signal and noise distributions as their decision criteria and little is known about the *in vitro* distributions of intrusions and

[12] Early presentations of the 1998 evaluation data by Lincoln used this form of ROC. More recent presentations, including the DISCEX paper [Lippmann et al. 2000], have used a formulation that presents the false alarm data in terms of false alarms per unit time. As discussed in Section 6.4, this formulation confounds detector characteristics and data rate.

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). A recent article by Axelsson [2000b] discusses a classical detection theory approach as applied to intrusion detection. The issue of tuning systems that use *a priori* distributions implicitly by learning or training procedures has been discussed above.

If the ROC is an appropriate mechanism for presenting the results of an IDS evaluation in which binary decisions are made, the curve will consist of a single operating point that expresses the percentage of true positives (the proportion of the actual attacks detected) plotted against the false positive percentage (the proportion of the units of analysis for which the system signaled an attack when none was present) 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 and may not be valid since it is based on a notion that does not really apply in the case of most IDSs. If one moves from the measured point towards the (0,0) origin, one passes along the line that would be obtained by changing the decision criteria towards the decision that everything is noise. In this case, we detect no signals but neither do we emit any false positives. Similarly, moving from the datapoint towards the (1,1) point represents the performance of the system if we changed the decision criteria towards one in which everything is considered signal. The shape of the curve is a function of both system detection rates and the actual mix of signal and noise. If no signal is present, the curve will follow the lower horizontal axis; if no noise is present, it will follow the left-hand vertical axis. In the environment in which most IDS systems operate, the signal percentage is very small requiring very low false positive rates for useful detection as discussed in Section 6.5.

During the DARPA PI meeting in August, Lincoln Lab acknowledged that the single-point ROC curves were not particularly useful, but they blamed the researchers because most of them assigned session scores that were either 0 (meaning no intrusion) or 1 (meaning certainty of an intrusion associated with the session). Had the scores been given as values that indicated the degree of confidence or belief that the system had detected an intrusion, it would have been possible to apply a sliding threshold to the scores and produce a ROC based (more or less) on a continuous distribution. For many IDSs, this would not be at all appropriate. For example, a system that performs a pattern match either finds a match or it does not. It makes no sense to say that the system has matched the string with a 75% probability. If one has a priori knowledge that 75% of the time a particular string is matched it is associated with an intrusion and the other 25% of the time it is benign, then a likelihood estimator recognizing an intrusion at the 75% confidence level might be appropriate. Unfortunately, neither the artificial data used in the evaluation nor the real data on which it was based has been characterized in ways that would facilitate this approach.[13]

During the evening session to discuss the 1999 evaluation which was about to begin, investigators were requested to report values other than 0 or 1 in their 1999 results. In the discussion of a system that uses an abstract state machine as a recognizer, it was suggested that, if the machine had 3 intermediate states between its entry and accepting states for a particular class of attack that it would be appropriate to report an attack at the 0.25 confidence level on the first state transition, and to report the same attack at the 0.50, 0.75, and finally 1.0 levels upon transition to the second, third, or accepting states respectively. This is equivalent to saying that a recognizer for "abcd" recognizes "acbd" as being "abcd" with 50% confidence and is clearly wrong.[14]

6.4 Errors per Unit Time

The DISCEX paper uses a nonstandard variation of the ROC presentation [Lippmann et al. 2000] that labels the X axis with false alarms per day rather than percent false alarms. A search of the traditional ROC literature [Swets 1988; Egan 1975; Swets and Pickett 1982] shows no mention of this formulation. It does appear, without comment or justification in the word-spotting literature [James and Young 1994; Jeanrenaud et al. 1994; Lippmann et al. 1994; Junkawitsch et al. 1998; Dharanipragada and Roukos 1998], where it is usually, but not always, referred to as a *ROC curve*. We speculated that false alarms per unit time might be a surrogate measure for percent false alarms under the assumption that spoken word rates are approximately constant (at least compared to Internet traffic rates) across many speakers or passages and we were concerned that there might be assumptions associated with the word-spotting usage that would not hold for intrusion detection. Our speculations as to the origin of the variation were wrong.

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

[13] Remarks made by Lincoln Lab during the August 1999 DARPA PI meeting indicate that such usage is intended as discussed in the following paragraph. Following this approach for an operational deployment is dubious since data necessary to determine the confidence levels is unlikely to be available.

[14] This is somewhat of a simplification, of course. The attack may not be sensitive to the order in which some steps are performed and the recognizer can take this into account. If the sensing process is known to be imperfect so that the probability of failing to sense a step that in fact was present can be estimated, there might be some justification for assigning an appropriate probability of detection to an incomplete recognition. As far as we can tell, perfect sensing is assumed for the Lincoln Lab data in that there seem to be no attacks in which key steps have been elided from the datasets simulating, for example, an overloaded network sensor.

(A. Martin personal communications). We were able to locate Stephen L. Moshier, one of the founders of Verbex and an author of some of the reports mentioned by A. Martin. S. L. Moshier (personal communication) reported 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 become overloaded at a higher rate. So that is a model of the user, not a model of the incoming voice signals."

What has apparently occurred here is a pragmatic, but *ad hoc* corruption of a basically powerful analytic technique. It becomes institutionalized within the community in which it originated, then exported without consideration of its underlying basis and without validation in the new environment.

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. Given adequate characterizations of the signal and noise distributions, it is even possible to discuss optimal detectors.

The *pseudo-ROC*, as we choose to call the 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 article is misleading for a number of reasons as explained by Huff [1954],[15] notably because of its failure to present the relevant information. Using the dataset as provided for the evaluation, but reassigning values to the timestamps attached to the data items, the false alarm rate per unit time can be manipulated to any degree desired. At the very least, the pseudo-ROCs presented by Lincoln Lab [Lippmann et al. 2000] should be labeled with the data rate on which the false alarm axis is based.[16] 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 hundred-fold increase in the false alarm rate when reported per unit time.

The desire to present data in such a way as to reflect the utility of the system under certain assumptions of user capabilities is understandable. Presenting the data this way is legitimate as long as the underlying

[15] Despite its somewhat inflammatory title, this book is a classic and should be read by everyone who wishes to avoid pitfalls in presenting statistical information in a meaningful fashion.

[16] The figures are labeled with the number of total sessions. One has to go to the text to determine that the average number of sessions per day is 1/10 of this value. Even so, there is nothing in the article that indicates the average session size or that the bit rate is in the tens of kilobits per second rate. In other words, the reader cannot easily estimate the expected false alarm rate in another context such as his or her own installation.

assumptions are made explicit. Even if this is done, the pseudo-ROC does not appear to be quite the right approach, especially as false alarm behavior improves. If the consideration is the number of alerts per unit time that the system operator can handle, a better appropriate X-axis might be total alerts, that is, true detections plus false alarms under the assumption that either case requires approximately the same operator effort to handle. If operator effort differs for the two cases, or the effort differs depending on the intrusion detected, some sort of weighted sum might be more useful as might a conversion to operator hours or fractions of a full-time operator required. In any case, the operator workload figures are a function of input data rate as well as the detector characteristics and the signal and noise distributions and the presentation has no predictive utility unless the rate and distribution assumptions are made obvious.

6.5 The Base Rate Fallacy, False Alarm Rates, and Operator Workloads

A recent paper by Axelsson [1999] explains why effective intrusion detection may require false alarm rates vastly lower than the 0.1% designated by DARPA. The paper gives the motivation for the problem with a simple example from the field of epidemiology. The example shows that false alarms will dominate true detections in any system in which the rate at which true events actually occur is less than the false alarm rate. This holds even in the face of a perfect ability to detect true events.

As an example, suppose that we have an intrusion detection system with a 100% true detection rate and a 0.1% false alarm rate. Suppose that we have a data set containing 1,000,000 units of analysis so that the system will make 1,000,000 decisions and that 1 in 50,000 units contains a true intrusion. Since the detection for true intrusions is perfect, we will raise alarms for all of the approximately 20 intrusions contained in the data set. We will also raise about 1000 false alarms while examining this data for a total of about 1020 alarms.

Now suppose that a million units of analysis represents a day's usage and that we expect on the order of 1 to 5 true detections per day (an intrusion rate of between one in a million and one in two hundred thousand units of analysis) and we wish to keep the total operator workload below the threshold of 100 events per day set by Lincoln Lab. This means that the false alarm rate is 0.0095% or less. If the data rate increases while the intrusion rate per unit time remains constant, the false alarm rate must decrease in proportion to maintain a constant operator workload. Increases in the intrusion rate also require an improvement in false alarm rate to keep the workload constant, but these become significant only if the number of detected true intrusions starts to use an appreciable fraction of the operator capacity.

Axelsson's work has been criticized as having an unrealistic view of the unit of analysis problem. It is claimed, for example, that selecting packets based on the protocol (and perhaps operation within the protocol) greatly reduces the number of cases analyzed so that higher false alarm rates are tolerable. While we don't entirely reject this criticism,[17] we think that it emphasizes the importance of coming to grips with the unit of analysis problem. After all, the sponsor of the IDS research (DARPA) being evaluated has set a goal of 0.1% for false alarm rates. It is up to them to specify 0.1% of what and ensure that the evaluation uses the appropriate unit.

6.6 Other Possible Objectives

We feel that an important objective of any evaluation of research systems is to provide information that gives insight into the potential of the systems being evaluated. This means developing an understanding of the system so that it can be determined whether deficiencies found during the evaluation are superficial or fundamental. For example, a number of both commercial and research systems use some form of pattern or string matching for at least part of their detection mechanism. The extent to which the pattern set covers the attack space is another factor that needs to be controlled in evaluating such systems. Researchers are more likely to concentrate on detection algorithms than on populating their systems with a complete pattern base. For such systems, it may be more important to determine that new patterns could be easily added (and the system made effective against a previously missed attack) than to measure the effectiveness of the incomplete pattern set.

If detector improvement is a goal of the evaluation, a set of post evaluation activities may be in order. These would begin with a generalization of the attacks used in the evaluation to develop an understanding of possible variants and alternative manifestations. This could be performed by the evaluators (Lincoln) as part of their search for new attacks and variations or by others. Given these "meta attacks," investigators would be encouraged to examine the fundamental assumptions underlying their detectors and systems to determine if there are inherent problems in applying their algorithms to the general cases.

7. THE 1999 EVALUATION AND ITS RESULTS

The 1999 evaluation was concluding as this article was first written and its results were not available. Preliminary results were presented at a DARPA PI meeting in December of 1999, but we have not had time to examine the presentations made there until recently. As far as we can tell, no additional details of the 1999 experimental setup or its results have been produced for public use. Superficially, the setup is similar to that provided for 1998 except that additional hosts and host types (including Windows NT) have been added. Sniffing data is now available both inside and outside the base. In addition, inside attackers are present and a directory of sensitive data is present as a specific target of attack. New attacks were present in the

[17] We suspect that selecting packets to analyze in this manner can be done with very close to 100% accuracy. If the result of the screening is a very small proportion of packets subject to error-prone further analysis, using the packet as the unit of analysis may result in an acceptably low false alarm rate.

training mix which consisted of three weeks of data with attacks present only in the second week. This was intended to provide "clean" data for training anomaly systems. The evaluation data emphasized new attacks (ones not present in the training data) and "stealthy" versions of old attacks intended to slip beneath the detection thresholds of many systems.

None of these changes are particularly substantive. One new factor is the elimination of session identification. Participants are required to identify the time of each attack. If the time given is within any session involved in the attack, credit is given. In addition, participants are asked to provide additional information identifying[18] the recognized attacks. This portion of the evaluation is optional, but "highly recommended."

The method used for scoring and for constructing ROC curves is simply inappropriate given the detection process used by many of the IDSs being evaluated. Each detected intrusion is supposed to be accompanied by a real number that indicates the confidence with which an intrusion was detected at the reported time involving the reported target. If a detection process suitable for the construction of a ROC is being used, the confidence level should reflect the system's historical ability to distinguish between signal and signal plus noise events in *a priori* known distributions. In this case, the distributions are not known with any precision and are not used in the decision process. Even if the distributions are known, there is probably not enough prior history to justify a good decision procedure.

The 1998 experiment did not describe the security policy in effect at Eyrie AFB. In the absence of any policy, the determination of whether a potentially hostile activity is an attack or intrusion is subjective. A permissive policy may explicitly or implicitly authorize activities that a more restrictive policy would forbid. For 1999 there is a statement of the "security policy" used at Eyrie AFB. The policy is quite loose: no services are blocked, root access from the outside is permitted for users that have root access, and so on. The policy may well tacitly permit some actions that might be considered intrusions. For example, while it is forbidden to install a network sniffer, one might argue that running "xsnoop" to capture an X users' session was legal if it was not done as preparation for an illegal activity (which preparation is forbidden). While simulating a poorly run installation may provide more possibilities for intrusion detection, it raises questions about whether actions that might be considered intrusive at a more tightly managed installation ought to be considered as intrusions here. The system security policy is one factor that can be used in deciding whether a sensed action should be considered as an attack.

Scans and probes are a special case because they are not always associated with activities that are or ought to be called intrusions. The apparent absence of these in the background data creates an unrealistic bias towards

[18] The additional information includes start date and time, duration, lists of sources, and targets with the ports involved. Attacks are to be identified according to the Weber [1998] taxonomy used in Kendall's [1999] Thesis. As noted above, this taxonomy is attacker-centric and may not be at all appropriate from the detection side.

setting detection sensitivities that would produce very large false alarm rates in the field. Most installations implicitly or explicitly set thresholds for such activity to reduce operator workload even though it means missing some probe or scan activities. Others simply report aggregate counts of such probes and scans on demand, but do not issue alarms for individual instances. During the 1998 evaluation, for example, a few relatively short scan sequences were among the attacks. Many IDS users would deliberately configure their systems to ignore these sequences (even though they are easily and reliably detected) because of their frequency and failure to correlate with later more serious activities. Assigning low probabilities to them does not really help if the number detected is sufficiently large. Setting an arbitrary threshold for this sort of activity in the evaluation process as opposed to encouraging the reporting of a confidence level determined in an *ad hoc* manner, would remove the subjectivity and allow the evaluation to concentrate instead on capabilities. If stealthy scans and probes are to be detected, capabilities to do this in the context of normal, nonintrusive scan and probe activity are needed.

While there are no publications as yet that describe the 1999 evaluations and results, slides from two presentations are available. The most comprehensive set of slides is from the December PI meeting and is available at <http://schafercorp-ballston.com/id_sia>[19]maintained by a DARPA support contractor. The other presentation was given by the Lincoln Lab group at the DISCEX conference in lieu of a presentation based on the article [Lippmann et al. 2000] which appears in the proceedings. It is available at the Lincoln Lab site referenced in Section 3.

The first presentation makes extensive use of the ROC variation in which the lower axis represents false alarms per day rather than false alarm percentages[20]. This leaves the issue of an appropriate unit of analysis for the study unresolved and mixes detector and data rate properties in the presentation of the results. The DISCEX presentation does not use any ROC forms.

The 1999 presentations represent a substantial improvement over their 1998 counterparts in trying to explain the results of the evaluation. As a result of suggestions made by the author and others at the Phoenix PI meeting in August, 1999, the evaluation proper was followed by an analysis of missed detections and false alarms in which the investigators were asked to explain why their systems made errors. The results of this exercise are encouraging, especially with respect to missed detections. Many of the missed detections were caused by deliberate omissions on the part of the investigators who had been told to concentrate efforts on algorithms and techniques as opposed to providing complete coverage in terms of signatures and protocols analyzed. Missed detections caused by these omissions

[19] This site is password protected. Contact John Frank (JFrank@Schafercorp-ballston.com) for access to this material.

[20] This form was used in the DISCEX paper [Lippmann et al. 2000] and has been discussed extensively in Section 6.3.

could be corrected and presumably would be corrected if the systems were deployed or brought to market. Probes missed due to thresholds being set too high are presumably fixable, although perhaps at the expense of higher false alarm rates since not all probes represent intrusive activity. It would be interesting to rescore the evaluation counting all easily fixable missed detections as correct. While this might lead to a somewhat optimistic view, it would approximate the performance that might be expected from a commercialized version of a research system. The problem of signatures for "new" attacks still remains, however. The situation with respect to false alarms is less encouraging. The false alarms are attributed to "Normal Variability" in the case of anomaly detectors and to the "Overlap in Traffic Patterns for Normal and Attack Traffic" for both systems. This would seem to call for an investigation of the fundamental principles underlying intrusion detection to determine if these limitations are artifacts of the approaches taken or represent a more serious and possibly intractable problem.

8. CONCLUSIONS

The Lincoln Lab evaluation program is a major, and in many ways impressive, undertaking, but its effects remain unclear. Several IDS researchers attribute much of the progress that they made to the program and especially its evaluation corpus. Others have complained that participation in the program had a serious adverse impact on their research efforts, estimating that as much as one third to one half of their total effort was spent on evaluation-related tasks that provided them with little or no benefit. It is not clear that results from the evaluation translate into field performance for the systems in question. Reducing the performance of these systems to a single number or to a small group of numbers or graphs as was done in 1998 does not appear to be 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. The 1999 approach of investigating missed detections and false alarms to try to understand their causes is probably much more useful in improving IDS systems, but it is not clear that gedanken experiments of the kind suggested by Alessandri[2000] would not have a similar effect for substantially less effort. Unsurprisingly, both the 1999 and 1998 evaluations demonstrated that the research systems, like their commercial counterparts, are very poor at detecting new attacks. As an anonymous reviewer pointed out, these results are unquestionable, but are almost independent of the background traffic, testbed, and scoring procedures. We add that because of this, they probably could have been obtained with much less effort than was required by the evaluation process.

The IDS field appears to be making relatively little progress at the present time. None of the systems funded by DARPA has achieved major breakthroughs and no individual system or combination of systems approaches the goals that DARPA set for the program. While DARPA apparently hoped that the evaluation would help its program reach those goals, this has not happened. It is hoped that this critique will lead to a rethinking of the evaluation process and a recreation of it in a form that will help DARPA reach its goals for IDS development.

It is interesting to speculate as to whether the criticisms made in in this article would have had a significant effect on the outcome of the evaluation. Given the emphasis placed on false alarm behavior in the 1998 evaluation, we see the failure to validate the background data as crucial. Until this is done, little weight can be attributed to the false alarm results. The case for changes in the taxonomy is less clear. We feel that a taxonomy based to a greater degree on manifestations would have been more helpful in explaining results and might have helped in communicating the nature of the improvements needed, but this is pure speculation on our part. Since the classes established by the taxonomy were used in scoring the evaluation, changes in the grouping of attacks might have made a significant difference in the rankings given evaluated systems. Since the publicly available reports from Lincoln do not show results for individual systems, we cannot rescore using an alternative grouping to confirm this speculation. We think that the unit of analysis question may be more important. DARPA set quantitative goals for the systems it sponsored. Providing measurements that relate to these goals in a meaningful way requires that appropriate units of analysis be determined. The 1998 approach provided an approximation for a suitable unit of analysis, but the 1999 evaluation provided no unit of analysis and reported false alarm results in a form that confounded system and data characteristics. As noted above, the detailed analysis of missed detections (new and old) and false alarms is largely independent of the evaluation methodology and thus is unaffected by the criticisms. However, if obtaining these results was a primary goal of the Lincoln effort, then performing the evaluation either as it was done or with the changes suggested is clearly an inefficient approach.

9. RECOMMENDATIONS

This section contains some recommendations for activities related to evaluation that might be of benefit to the IDS community as a whole. It covers the development of more appropriate measures of performance, better traffic characterization and validation, extension of the experiment to commercial systems, and establishment of a canonical attack repository to support future research. Most important is adequate funding for efforts of this kind. If an effort of this kind is worth doing, it is worth doing well. Many of the deficiencies noted in the article might not have occurred if the funds to pursue troublesome issues such as data validation and pilot studies had been available.

Better documentation, in the form of detailed technical reports describing the specifics of, and the rationale behind, the experimental approach might have obviated the need for this article. It appears that such reports are now in preparation[21] and we eagerly await their appearance. For future experiments, the timely creation of a public record should serve to remove some criticisms and better focus others.

9.1 Measures

The operating points or curves obtained by plotting true positives against false positives is a relatively poor basis for characterizing research IDS systems since it provides no insight into the reasons for IDS performance (good or bad). We note that, even if the measure is useful, there is no denominator for the false alarm component. Real effort is needed to make sure that an appropriate common denominator is found for both the true and false positive terms. As noted earlier, this requires solving the unit of analysis problem which may result in different denominators for different systems. Requiring each system to report both detections and nondetections might be a fruitful approach

Work is necessary to develop more helpful measures of performance. An example of a constructive measure would be a metric that allows a useful description of the differences between signatures that are recognized correctly as an attack and those that provoke a false alarm. Metrics of this type could contribute to the refinement of system capabilities.

9.2 Traffic Characterization

There is a need for calibrated and validated artificial test data sets or test data generators. As long as the false alarm rate is used as a measure of system effectiveness, it must be possible to make sure that the false alarm rate for synthetic data has a well-understood relationship to the false alarm rate of “natural” data for systems under test. At this point, we do not have an adequate understanding of the factors that contribute to false alarms to provide such a calibration. Recent work by Maxion and Tan [2000] provides some guidance on the criteria for successful synthesis, but a great deal of additional observational work on real data sources is needed.

9.3 Evaluation of Commercial Products

Since the objective of the IDS research program is to produce IDS systems that are “better” than commercial products, a commercial baseline is needed, based on the same techniques and criteria used to evaluate research systems. This should be done by an independent party, and should be an ongoing activity since the systems to be evaluated are in a constant state of change and the evaluations are likely to be obsolete within a few months of their performance.

[21] Two theses [Das 2000; Korba 2000] appeared in June of 2000 while this article was under review. Additional articles and reports are in preparation or submitted for publication, but are not publicly available (and are not cited) although some drafts appear on the Lincoln’s password-protected Web site.

9.4 Attack Examples for the Research Community

The research community has complained that it expends substantial effort generating attacks to test IDS systems. Developing an "attack on demand" facility that would track the latest attacks and make them available would be useful. As with the previous suggestion, this needs to be an ongoing effort to ensure that new attack capabilities are brought to the attention of both the research and commercial communities in a timely fashion. Such a system would support an ongoing effort to understand the nature of vulnerabilities and the attacks that exploit them and could foster improvements in infrastructure as well as IDSs.

ACKNOWLEDGMENTS

I want to thank many members of the intrusion detection community for helpful comments on earlier versions of this article. Some of them wish to remain anonymous and I respect those wishes. 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 whence it was appropriated by Lincoln Lab. Stefan Axelsson of Chalmers University provided useful comments on an earlier draft of the article and valuable insights into intrusion detection as a signal detection problem. Julia Allen and Tom Longstaff of CERT have been most supportive and without their efforts, the article would not exist. Jim Binkley of Portland State provided data on typical network traffic levels at the PSU School of Engineering. Richard Lippmann and the other members of the evaluation effort at Lincoln Laboratory have taken some of the comments made in earlier versions of this article to heart, resulting in substantial improvements in the 1999 analysis and brighter prospects for future efforts of this kind.

REFERENCES

ALESSANDRI, D. 2000. Using rule-based activity descriptions to evaluate intrusion-detection systems. In *RAID2000*, H. Debar, L. Me, and S. F. Wu, Eds. Springer-Verlag, New York, NY, 183–196.

ALLEN, J., CHRISTIE, A., FITHEN, W., MCHUGH, J., PICKEL, J., AND STONER, E. 2000. State of the practice of intrusion detection technologies. CMU/SEI-99-TR-028,CMU/SEI. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

AXELSSON, S. 1999. The base-rate fallacy and its implications for the difficulty of intrusion detection. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*. 1–7.

AXELSSON, S. 2000. Intrusion-detection systems: A taxonomy and survey. 99-15 (March).

AXELSSON, S. 2000. A preliminary attempt to apply detection and estimation theory to intrusion detection. 00--4 (March).

BELLOVIN, S. M. 1993. Packets found on an internet. *SIGCOMM Comput. Commun. Rev. 23*, 3 (July), 26–31.

CERT COORDINATION CENTER. 2000. Cert security improvement modules. http://www.cert.org/security-improvement

DAS, K. 2000. Attack development for intrusion detection. Master's Thesis. Massachusetts Institute of Technology, Cambridge, MA.

DEBAR, H., DACIER, M., WESPI, A., AND LAMPART, S. 1998. An experimentation workbench for intrusion detection systems. Res. Rep. RZ 2998 (#93044) (Sept.). Research Division, IBM, New York, NY.

DHARANIPRAGADA, S. AND ROUKOS, S. 1998. A fast vocabulary independent algorithm for spotting words in speech. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (May). 233–236.

DURST, R., CHAMPION, T., WITTEN, B., MILLER, E., AND SPAGNUOLO, L. 1999. Testing and evaluating computer intrusion detection systems. *Commun. ACM 42*, 7, 53–61.

EGAN, J. P. 1975. *Signal Detection Theory and ROC Analysis*. Academic Press, Inc., Orlando, FL.

EINBINDER, H. 1964. *The Myth of the Britannica*. Grove Press, New York, NY.

GRAF, I., LIPPMANN, R., CUNNINGHAM, R., FRIED, D., KENDALL, K., WEBSTER, S., AND ZISSMAN, M. 1998. Results of DARPA 1998 offline intrusion detection evaluation. http://ideval.ll.mit.edu/results-html-dir/

HEBERLEIN, L. T., DIAS, G. V., LEVITT, K. N., MUKHERJEE, B., WOOD, J., AND WOLBER, D. 1990. A network security monitor. In *Proceedings of the IEEE Symposium on Research in Security and Privacy* (Oakland, CA). IEEE Computer Society Press, Los Alamitos, CA, 296–30304. http://olympus.cs.ucdavis.edu/papers.html.

HUFF, D. 1954. *How to Lie with Statistics*. W. W. Norton & Co., Inc., New York, NY.

JAMES, D. A. AND YOUNG, S. J. 1994. A fast lattice-based approach to vocabulary independent wordspotting. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 2*. 337–380.

JEANRENAUD, P., SIU, M., ROHLICEK, J. R., METEER, M., AND GISH, H. 1994. Spotting events in continuous speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 2*. 381–384.

JUNKAWITSCH, J. AND HÖGE, H. 1998. Keyword verification considering the correlation of succeeding feature vectors. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing* (May). 221–224.

KENDALL, K. 1999. A database of computer attacks for the evaluation of intrusion detection systems. Master's Thesis. Massachusetts Institute of Technology, Cambridge, MA.

KLATT, D. H. 1977. Review of the ARPA speech understanding project. *J. Acoust. Soc. Amer. 62*, 1345–1366.

KORBA, J. 2000. Windows NT attacks for the evaluation of intrusion detection systems. Master's Thesis. Massachusetts Institute of Technology, Cambridge, MA.

LIPPMANN, R., HAINES, J. W., FRIED, D. J., KORBA, J., AND DAS, K. 2000. The 1999 DARPA off-line intrusion detection evaluation. In *RAID2000*, H. Debar, L. Me, and S. F. Wu, Eds. Springer-Verlag, New York, NY, 162–182.

LIPPMANN, R. P., CHANG, E. I., AND JANKOWSKI, C. R. 1994. Wordspotter training using figure-of-merit back propagation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 2*. 385–388.

LIPPMANN, R. P., CUNNINGHAM, R. K., FRIED, D. J., GARFINKEL, S. L., GORTON, A. S., GRAF, I., KENDALL, K. R., MCCLUNG, D. J., WEBER, D. J., WEBSTER, S. E., WYSCHOGROD, D., AND ZISSMAN, M. A. 1988. MIT Lincoln Laboratory offline component of DARPA 1998 intrusion detection evaluation. http://ideval.ll.mit.edu/intro-html-dir/.

LIPPMANN, R. P., FRIED, D., GRAF, I., HAINES, J., KENDALL, K., MCCLUNG, D., WEBBER, D., WEBSTER, S., WYSCHOGRAD, D., CUNNINGHAN, R., AND ZISSMAN, M. 2000. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In *Proceedings of the on DARPA Information Survivability Conference and Exposition* (DISCEX '00, Hilton Head, South Carolina, Jan. 25-27). IEEE Computer Society Press, Los Alamitos, CA, 12–26.

MAXION, R. A. AND TAN, K. M. C. 2000. Benchmarking anomaly-based detection systems. In *Proceedings of International Conference on Dependable Systems and Networks* (June). 623–630.

PAXSON, V. 1999. Bro: A system for detecting network intruders in real–time. *Comput. Netw. J. 23-24* (Dec.), 2435–2463.

POMERANZ, H. 1999. Solaris security: Step by step. http:www.sans.org

PTACEK, T. H. AND NEWSHAM, T. N. 1998. Insertion, evasion, and denial of service: Eluding network intrusion detection. http://www.secinf.net/info/ids/idspaper/idspaper.html

PUKETZA, N., CHUNG, M., OLSSON, R. A., AND MUKHERJEE, B. 1997. A software platform for testing intrusion detection systems. *IEEE Software 14*, 5 (Sept.), 43–51. http://seclab.cs.ucdavis.edu/papers.html

PUKETZA, N. J., ZHANG, K., CHUNG, M., MUKHERJEE, B., AND OLSSON, R. A. 1996. A methodology for testing intrusion detection systems. *IEEE Trans. Softw. Eng. 22*, 10, 719–729. http://seclab.cs.ucdavis.edu/papers.html

SIFT. 1984. Peer review of a formal verification/design proof methodology. RTI/2094/13-01F.

SIFT. 1985. Peer review of a formal verification/design proof methodology. Conference Publication CP-2377.

SWETS, J. A. 1988. Measuring the accuracy of diagnostic systems. *Science 24*, 48, 1285–1293.

SWETS, J. A. AND PICKETT, R. M. 1982. *Evaluation of Diagnostic Systems*. Academic Press, Inc., New York, NY.

TRIPWIRE, INC. 2000. Tripwire 2.2.1 for UNIX, Users Manual. WWW.TripwireSecurity.com.

WEBER, D. 1998. A taxonomy of computer intrusions. Master's Thesis. Massachusetts Institute of Technology, Cambridge, MA.

WHITING-O'KEEFE, Q. E., HENKE, C., AND SIMBORG, D. W. 1984. Choosing the correct unit of analysis in medical care experiments. *Med. Care 22*, 12 (Dec.), 1101–1114.

Received: April 2000; revised: November 2000; accepted: November 2000