Qualitative Probability for Intrusion Detection

Robert P. Goldman and John Maraist

Abstract

We study the performance of a qualitative probabilistic scheme for Intrusion Detection System (IDS) fusion, implemented in the Scyllarus and MIFD systems. Scyllarus and MIFD assess the likelihood of various cyber attack events based on reports from IDSes, using the System $Z+$ qualitative probability scheme of Goldszmidt and Pearl. Scyllarus has shown good performance — accurately identifying attacks while substantially reducing false positives — in tests and active use. However, it is difficult to draw *general* conclusions about an intrusion detection approach based on field testing. Furthermore, the field testing treated Scyllarus as a whole, rather than specifically exploring its qualitative reasoning. In the experiments we describe here, we construct simulated sets of attacks and sensors, and explore the recall and precision of MIFD as we vary configuration parameters. Our experiments are particularly aimed at evaluating the accuracy of the qualitative scheme, and exploring how that accuracy will degrade as the simplifying assumptions underlying the qualitative scheme fit the actual situation less and less well. Our results show that the qualitative reasoning scheme can be expected to degrade gracefully in both recall and precision.

1 Introduction

In previous work, we have developed a technique for intrusion detection system (IDS) fusion (Goldman and Harp, 2009) in cyber defense, whose likelihood computations are based on the System $Z+$ qualitative probability scheme (Goldszmidt and Pearl, 1996). Our original system has been extensively tested in real networks, using both real and synthetic data. Unfortunately, it is difficult to draw many general conclusions based on such evaluations, whose results are very specific to the test network, traffic, and attacks. This is a general problem of evaluating IDSes, which we discuss further below.

As a complement to these in-the-field evaluations, in this paper we present experimental analysis of the underlying reasoning machinery in our system, using simulated sensors and events. Our experimental analysis, divorced from specific networks, sensors, etc., allows us to draw general conclusions about the applicability of our approach, and identify when it will behave well and when poorly. Additionally, our original work was in the context of an open system that fused a set of "take them or leave them" IDSes. We are moving to incorporate our fusion system in an integrated system that will perform computer network defense. In this new framework, we have the opportunity to make decisions about what sorts of sensors to deploy, possibly developing new sensors in the process, and where to deploy these sensors. The results here will help inform such decisions.

The key issue that we explore here is the suitability of qualitative probabilistic techniques, and System $Z+$ in particular. System $Z+$ asks us to treat uncertain phenomena as if they fall into a small set of *qualitatively distinct* levels of likelihood. What we do in our experiments is to explore how robust our techniques are as this abstraction fits the world less and less well. For example, we consider how our qualitative approach degrades as it is used to model probability distributions in which the probabilities corresponding to the qualitative strata get closer and closer together.

While our experiments are aimed specifically at IDS fusion, they should be of interest to other researchers and developers who wish to pursue qualitative methods. At the level of abstraction treated in this paper, our results will be of direct interest to those pursuing qualitative probabilistic approaches to information and sensor fusion. The fusion problems we address are very close in structure to diagnosis problems, so our results should also be of interest to that community.

The experimental results reported here are very encouraging. They show that under very weak assumptions, the sensor fusion approach that we propose will provide good detection rates with a low rate of false positives. They also show that the behavior of the system degrades gracefully when these assumptions are violated. They confirm and explicate previous results in less controlled test environments.

In the next section of the paper, we introduce the problem of IDS fusion, listing many difficult challenges. Then we describe our own approach to this problem, as implemented in the Scyllarus and MIFD systems. We describe our experimental designs, present the results and conclude with some proposals for future work.

2 Intrusion Detection Fusion

Intrusion Detection Systems (IDSes) are systems that sense intrusions in computer networks and hosts. IDS fusion is the problem of fusing reports from multiple IDSes scattered around a computer network we wish to defend, into a coherent overall picture of network status. In the computer security community, this process is often called "IDS correlation," but we prefer "fusion" as it more accurately describes the process, and doesn't collide with statistical terminology. Indeed, in some cases the term "IDS correlation" is not used to describe an automated process, but rather a human-executed process of investigating an IDS-generated alert (e.g., (Drew, 2014)).

The function of a fusion system is to take reports from multiple IDSes and fuse them into a coherent picture of the state of the defended network. To do so, it must answer two questions:

1. How do the reports issued by the IDSes refer to events? Do multiple reports from the same IDS refer to the same event? Do multiple reports from different IDSes refer to the same event?
2. Of the hypothesized events, which do we accept as actual?

In this paper we will focus on the second of these two questions, assuming a solution to the first. For discussions of the first question, see (Goldman and Harp, 2009).

Existing IDSes are not designed to work together, as part of a suite of sensors. Instead, each program generates a separate, and often voluminous, stream of reports, and fusing them into a coherent view of the current situation is left as an exercise for the user. Ideally, network administrators would have a suite of different IDSes active, because different IDSes have different strengths and weaknesses. IDSes are typically divided into network-based (NIDS) and host-based (HIDS). They are also divided into signature-based and anomaly-based.

NIDS systems are easy to deploy, because they need to be installed in only one or two locations to watch all relevant traffic. Unfortunately, they also have substantial problems with false positives and false negatives, because they don't "know" the meaning of the traffic they see. Often they must draw conclusions based only on information in packet headers (although there are some systems that attempt to reconstruct higher level protocols from packet data), and attempts to do extensive reasoning about traffic will either slow down the network, or must ignore some packets altogether. Conversely, a HIDS may take into account much more information about the state of a defended system, but installing HIDS pervasively involves deploying and configuring a lot of software. Typically, HIDS are installed on only the most critical assets.

Signature-based systems attempt to match patterns of known bad behavior. Such IDSes are plagued by false negatives, both because of novel attacks for which there are no patterns to match, and because knowledgeable attackers take countermeasures against signature identification (e.g. encrypting malware or permuting its structure). By contrast, anomaly detectors are prone to false positives. False positives may occur because anomalies are simply anomalies, and not malice. They may also occur because the detector believes there is an anomaly when none exists. This can happen because computer usage patterns are notoriously difficult to learn: they are often non-stationary, they vary substantially from location-to-location, etc.

The most substantial challenge in managing IDSes is the information overload that they can impose. Much of this overload comes from the high false positive rate. This rate can be because of inaccurate sensors, as outlines above. However, it can also come from base rate problems, where even accurate sensors will give a high number of false positives (Axelsson, 1999). IDS owners regularly either ignore or partially disable them, unable to absorb the massive stream of reports. "Users attending an 'ABCs of IDS' event at London's City University yesterday said more the 80 per cent of the alerts they received were false, with one citing 60 alerts he had received about non-existent problems that morning at 0300." (Leydon, 2001) Recently, Target is reported to have ignored warnings about the data breach that resulted in theft of millions of credit cards (Schwartz, 2014): "They are bombarded with alerts. They get so many that they just don't respond to everything," said one expert (Finkle and Heavey, 2014). To get a sense of the gravity of this problem, see Figure 1, which shows how our earlier system was able to winnow the flow of reports in a small corporate network (Goldman and Harp, 2009).

Data reduction, a comprehensive overview, and overcoming false positives and false negatives are the goals of IDS fusion. Such systems aim to overcome the limitations of existing IDSes by assembling a suite of sensors. This can be a very efficient way to overcome the problem of false positives, as long as we can find sensors that fail relatively independently.

Related work SecurityFocus has developed the Attack Registry and Intelligence Service (ARIS) (ARIS, 2003). The ARIS extractor collects IDS reports from four different IDSes, formats them in XML, and presents them in an incident console. However, it makes no attempts to fuse the reports or weigh the evidence for and against them. MetaSTAT is a fusion system that is built on a set of STAT-based IDSes (Vigna et al., 2001). STAT is a signature-based IDS that detects events by matching against extended finite-state event models. MetaSTAT uses finite-state models of across-sensor events to consume at a higher level the events generated by lower-level sensors. MetaSTAT does not attempt to judge the plausibility of different events. EMERALD/eBayes (Valdes and Skinner, 2001) fusion is the most similar to Scyllarus and MIFD. The eBayes sensors are Bayes net-based, and the correlation approach allows "upstream" sensors to adjust the priors on "downstream" sensors. eBayes's fusion is limited to clustering together alerts that meet a similarity criterion; they do not have models of high-level events as in our system (see below). To the best of our knowledge, eBayes does not address the difficult issues of acquiring probability parameters for IDS fusion. Prelude Correlator (Vandoorselaere, 2008) is part of the open source Prelude IDS information system, and allows users to analyze reports sent to Prelude from compatible IDSs. Users provide rules written in Lua, a scripting language inspired by Scheme and Icon. Its function is closest to the Scyllarus clustering subsystem, but knowledge resides in stateful rules instead of an ontology of attacks. A commercial product, Arcsight Enterprise Security Manager (ArcSight, 2008), also ties correlated IDS reports to an installation's security goals and vulnerability information.

3 The Scyllarus and MIFD systems

Our original system, entitled Scyllarus, was an open system which performed IDS fusion on a diverse set of third party IDSes (Goldman et al., 2001,?). The successor system, MIFD (Model-based Intrusion Fusion and Detection), is a new version of the Scyllarus system, enhanced as part of the STRATUS system, which integrates IDS fusion into a comprehensive suite for cyber defense (Thayer et al., 2013). For simplicity, we will refer to the IDS fusion system as "MIFD," but except where we specify otherwise, our account will apply to Scyllarus, as well.

The process of IDS fusion involves a number of different subtasks. The preliminary tasks are collection and translation. **Collection:** IDS outputs (reports) must be collected and presented to MIFD. **Translation:** IDS reports must be translated into a common data structure, the *sensor report*, for processing. This involves a certain amount of data rectification.

The first task performed by MIFD itself is **Clustering:** From the sensor reports, MIFD generates *event hypotheses* to represent the underlying events of interest that may have caused the sensor reports. A sensor report that provides evidence for an event hypothesis is a *supporter*. A very simple Bayes net generated by MIFD would look like Figure 2(a), a single event with a number of sensor reports. In general, though, sensor reports may be ambiguous, and support multiple different event hypotheses. A more complex example is shown in 2(b), where sensor reports are ambiguous between two hypotheses. Note that these sensors don't give measurements that would distinguish between the two hypotheses: they are simply on and off. In our experience IDS reports typically detect indirect features of intrusions, and are often spoofed by benign events that we may detect. E.g., a local software update server may look like an attacker performing reconnaissance. For this reason, MIFD's belief networks contain benign event hypotheses as well as attack event hypotheses. Note also that some events have components; in such cases one event hypothesis may be a supporter of another. We will not discuss such complex events in this paper. The result of the clustering process is a directed graph of event hypothesis and sensor report nodes that is a Bayesian belief network.

The final inference step is **Assessment:** MIFD assigns a degree of belief to each of the event hypotheses. MIFD does this by performing Bayesian updating on the belief network constructed by the clustering process. As we discuss below, MIFD does this using the System $Z+$ qualitative probability scheme rather than standard probability theory.

Scyllarus and MIFD are not batch processes and are intended to be run for a long time. Scyllarus has run for months at a time; MIFD is currently being used only in small-scale simulations, so does not have all the features needed for long-term operation. To en-

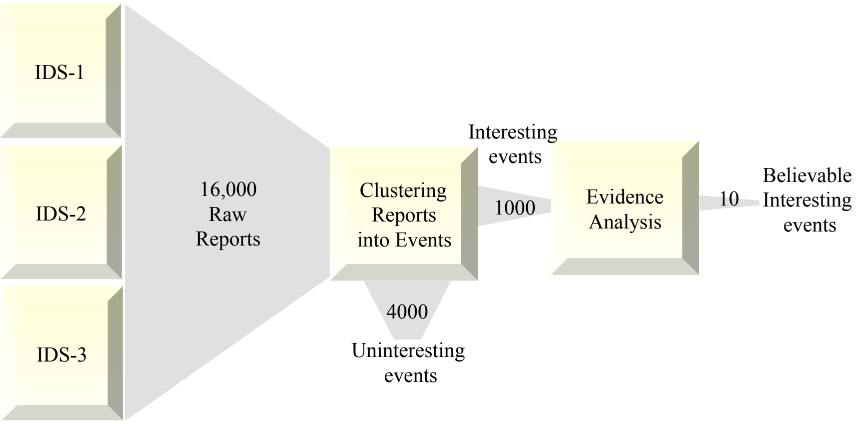


Figure 1: Scyllarus workload reduction (Goldman and Harp, 2009).

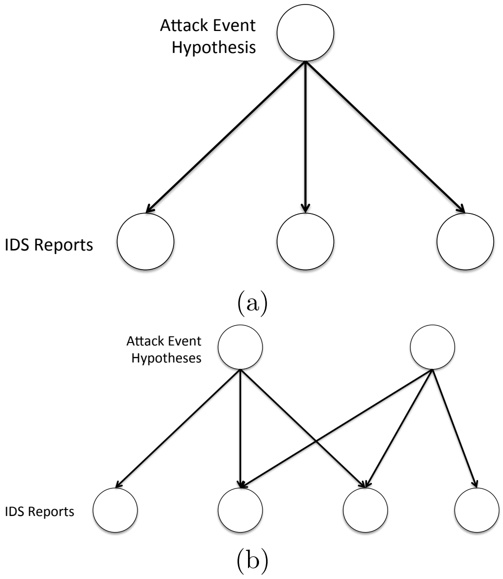


Figure 2: Bayes networks generated by MIFD.

able long term function, a number of steps are necessary. First, set of steps above are performed cyclically. MIFD reads reports, clusters them, and periodically interrupts the reading-clustering process to perform assessment. Second, in general the Bayesian belief network is not fully connected. When performing assessment, MIFD operates on connected components individually. Third, in order to free up resources, Scyllarus periodically removes old event hypotheses and sensor reports from working memory and writes them out to disk (this feature has not yet been implemented in MIFD).

MIFD follows Pearl's exhortation that the important thing about probability theory is the patterns of inference it enables (explaining away, combining forward causal inference and evidential reasoning, etc.), rather than precise numerical calculations (Pearl, 1988). He recommends we use these patterns even when precise numbers are not available, and that is what MIFD aims to do. As outlined above, MIFD treats IDS fusion as an abductive problem, formalized using Bayes nets. Unfortunately, we do not have the necessary parameters available to us – probabilities of attacks, false positives, false negatives, etc. Indeed, these probabilities are unlikely *ever* to be available to us: the populations of attacks change, one network has little in common with another, distributions are non-stationary, etc. No current ML technique can overcome these limitations: for a good discussion of the challenges IDS poses to ML, see (Sommer and Paxson, 2010).

Our approach, based on qualitative probabilities, shares the basic structure of normal probability theory but abstracts the actual probabilities used. Our approach is based on System $Z+$, developed by Moisés Goldszmidt and Judea Pearl (Goldszmidt and Pearl, 1996). In System $Z+$, events are given a natural number rank, κ, that corresponds to their degree of surprise (e.g., a rank of one is more surprising than zero). The semantics of this scheme comes from a set of probability distributions in which the probabilities are polynomials in some infinitesimal ϵ. In this scheme, the κ rank corresponds to the exponent of the leading term of the polynomial. The scheme is similar to the "big-O" notation used for evaluating computational complexity in computer science. The effect of this semantics is to give System $Z+$ a qualitative flavor by providing a "ladder" of events of qualitatively different orders of likelihood.

As far as computation is concerned, we may apply the normal operation of probability theory: conditionalization, Bayes' law, etc. However, the arithmetic operations we use must change. Rather than multiplying probabilities, we add degrees of surprise; rather than adding probabilities, we use min. Goldszmidt and Pearl (Goldszmidt and Pearl, 1996) provide the following substitutions:

$P(\omega) = \sum_{\phi \in \omega} P(\phi)$	$\kappa(\omega) = \min_{\phi \in \omega} \kappa(\phi)$
$P(\omega) + P(\neg\omega) = 1$	$\kappa(\omega) = 0 \vee \kappa(\neg\omega) = 0$
$P(\omega \mid \phi) = P(\omega \wedge \phi)/P(\phi)$	$\kappa(\omega \mid \phi) = \kappa(\omega \wedge \phi) - \kappa(\phi)$

When operating in Bayes networks, with conditionally independent ω and ϕ we additionally have $\kappa(\omega \wedge \phi) = \kappa(\omega) + \kappa(\phi)$.

There are a number of efficient algorithms for finding the posterior distributions of Bayesian networks, conditional on observations of some of the random variables. These algorithms may readily be adapted to provide posterior κ rankings instead of probabilities. MIFD uses a translation from Bayes networks into an ATMS (Forbus and deKleer, 1993); see (Charniak and Goldman, 1988; Poole, 1993; Provan, 1989) and (Goldman and Harp, 2009) for this encoding. This is not an especially efficient implementation but in our past experience, runtime was dominated by I/O and difficult cases were handled by special-purpose optimizations; finding the optimal inference method in terms of runtime has been less important than finding an algorithm easy to modify for System $Z+$.

In MIFD, the assessment component will rank all hypotheses h as either *likely*, *plausible*, or *unlikely*. h is *likely* if $\kappa(h) < \kappa(\neg h)$, *plausible* if $\kappa(h) = \kappa(\neg h)$ and *unlikely* otherwise.

The MIFD system requires comparatively few κ parameters given the above design. For sensors, we need κ(false-positive), and for events we need κ(event). In practice, we assign global defaults for these, based on how generally accurate the input IDSes are: for example, we set these κ's so that it takes $n = 2$ or3 sensors for us to judge an event as *likely*. We also apply the following consistency constraints:

$$\begin{aligned} &\kappa(\text{false-positive}), \kappa(\text{benign}) \\ &< \kappa(\text{attack}) \\ &< n \cdot \kappa(\text{false-positive}) \end{aligned} \tag{1}$$

That is:

- A single sensor false positive is less surprising than an attack event.
- A benign event is also less surprising than an attack event.
- An attack event is less surprising than false positives from n of the sensors which detect that event.

With a few exceptions, this paper assumes $n = 3$ sensors, so we have $0 < \kappa(\text{benign-event}) < \kappa(\text{attack-event}) < 3\kappa(\text{false-positive})$. In actual deployments, we start with this, and then nudge the false positive rankings up in the rare cases where we have a particularly good sensor. If we have a particularly bad sensor, we typically drop it. We are much less likely to adjust attack event probabilities, but there are exceptions such as reconnaissance (very common), and some complex events.

4 Experimental Design

Our experiments all aim to probe the strengths and limitations of the System $Z+$ qualitative abstraction. According to this abstraction, we treat events with different κs as being qualitatively different. Of course, events in the real world are only quantitatively different, so the question is, as we address different probability distributions *as if* they are qualitatively different, how does our performance degrade? In general, we will vary the probability distributions so that probabilities corresponding to different κs are initially very far apart, and then bring them together gradually. What we are hoping to see is a graceful degradation in recall and precision as the qualitative simplification fits the underlying probabilities less and less well.

Of course, we are not interested in arbitrary probability distributions, but only those that reflect features of the IDS fusion problem: Our experiments are based on *settings* in which there are *attack event prototypes* and *benign event prototypes*. There are also *sensors*, which simulate IDSes, each of which can sense some number of attack events and possibly additionally (inadvertently) some number of benign events. Since there is no opportunity to perform fusion if events are not sensed, in our simulated configurations each attack event will be sensed by some number of sensors, greater than 1. We vary the sensor-to-attack ratio in our experiments.

To set up an experiment, then, we specify a set of parameters (a configuration) according to which we randomly generate a number of *settings*, as outlined above. For each setting we carry out some number of *runs*. In each run we sample a number of attack events and benign events that occur, based on the corresponding prototypes. These are Bernoulli random variables; we discuss how their probabilities are set below. In the experiments we report here, we generated 1,000 settings, and 100 runs per setting.

After sampling events, we sample from the sensors according to their false positive and false negative probabilities. For each of the i events which the sensor might detect, we consider the probability *fn* of a false negative, so the probability of the sensor firing is $1 - fn^i$. If none of the sensed events occurred, the sensor will fire according to its false positive probability.

Having generated the sensor reports for the run, we then use them as input to MIFD, and assess the resulting Bayes networks. We extract the set of attack event hypotheses that have been labeled as *likely* by MIFD. We compare this set with the ground truth and compute recall and precision, which we report in our results section.

We perform sampling separately for precision and recall, because the probability of attacks is very low. For precision, false positives are of critical importance, so we simply sample based on the event probabilities. However, for recall, we care only about trials where attacks actually occur. Accordingly, we perform rejection sampling, rejecting any event samples in which no attacks occur.

4.1 MIFD execution environments

Generation of each experimental run of MIFD is governed by a number of experimental parameters. Each of these parameters may be either a constant, or a random variable. Over any particular experiment, we will vary one or two parameters, and set others to values typical for an IDS.

- *num-events* — The total number of events to be detected by each sensor, a measure of the sensor's specificity. Except where we vary this parameter for a particular experiment, we use a random variable which returns 1 80% of the time, and 2 otherwise.
- *sensor-to-attack-ratio* — The number of sensors which should detect each attack event. We default to a constant value of 3 for this parameter.
- *sensor-overlap* — The number of attacks to be detected by each sensor, specified as a probability checked when deciding whether to associate an additional attack with a sensor. We default to a constant value of 0.2 for this parameter.
- κ(false-positive)— The qualitative probabilities of false-positives and false-negatives for the sensors. We choose these to be 1 in configurations where it takes 3 reports to rank a hypothesis likely, or 2 where it takes only 2.
- p(false-negative) — MIFD does not reason about false negatives at the moment; we take the probability of false negatives to be whatever corresponds to $\kappa = 2$ in the configuration.
- κ(attack) and κ(benign) — The qualitative probabilities of attack and benign events. We default to constant values of 2 and 1 respectively for these parameters.
- *sensor-to-benign-ratio* — The number of sensors which should detect each benign event. We default to a constant value of 3 for this parameter.
- *kappa-translations* — Translations (into a constant or into a random variable) of qualitative probability values to real values in $[0, 1]$. By default we take $\kappa(0) = 0.5$, $\kappa(1) = 0.01$, $\kappa(2) = 0.001$.
- *num-attacks* — Number of attack prototypes; the number of attacks that may take place . Defaults to 4.

Creating an experimental setting entails creating attack event, benign event and sensor prototypes, and assigning event detection among sensors. The number of attack prototypes is set by the *num-attacks* parameter; each prototype is associated with an occurrence probability given by (or sampled for each prototype from) κ(attack). The number of sensors is not fixed, but instead depends on several configuration parameters. For each attack, we keep track of the number of sensors which we must assign to detect it; this value is initially set from *sensor-to-attack-ratio*. We create new sensors as long as any attack requires assignment to an additional sensor. A new sensor is initially assigned a total number of events which it will detect from *num-events*, and some set of attack events. Each sensor is assigned at least one attack; it is assigned additional attacks up to its total number of events as consecutive samples of a Bernoulli random variable with success probability *sensor-overlap* return 1. After all sensor report prototypes are created and assigned attacks, we assign benign events to the sensors as needed to satisfy their total number of events. We create the necessary number of benign event prototypes to satisfy both *sensor-to-benign-ratio* and each sensor's total event count, and distribute them randomly among the sensors according to their total event counts.

4.2 Metrics

We evaluate MIFD's performance in terms of *precision* and *recall*, by comparing the set of events that MIFD considers *likely*, with ground truth from the simulator. Recall is the percentage of actual attack events which are labeled as likely. Precision is the percentage of attack events labeled as likely which actually occurred. Because of the high rates of sensing and the low rate of events, precision and recall must be in the high nineties, or the sensing system is likely to be unusable.

4.3 Objectives

Our experiments aim to probe the limits of performance of our information fusion approach. Our first two sets of test check to see how MIFD's performance degrades as events which it treats as qualitatively different get closer and closer in likelihood. Our next test shows how MIFD's performance degrades as the sensors it incorporates get less and less discriminating. Finally, we examine how MIFD's sensor fusion can help with base rate issues: here we do some experiments, and also provide some analytic information.

Varying probabilities. Our first experiment aims to see how performance of MIFD degrades as we progressively violate the assumption that the different levels of the stratified likelihood ranking are qualitatively different. Specifically we consider a sequence of settings, in which we assign exact Bernoulli probabilities to correspond to the κs, in all of which $P(\kappa = 0) = 0.5$, but where the other $P(\kappa = i)$ vary. We performed this test with *sensor-to-attack-ratio* both 2 and 3; in the former case we took κ(false-positive)=2, κ(benign)=2, κ(attack)=3 to preserve the coherence inequalities (Eqn. 1). Over the course of the sequence of settings, the probabilities corresponding to the κs get closer and closer.

Non point-value distributions. Our second experiment is a variation on the first, in which we use a beta distribution to define a second order probability distribution corresponding to each κ ranking. Recall that the beta distribution is the Bayesian subjective prior for Bernoulli distributions in which increasing $\nu = \alpha + \beta$ corresponds to increasing "virtual sample size" or increasing confidence. In our experiments, we fix the mean values μ of the distributions to which we translate $\kappa(0)$, $\kappa(1)$ and $\kappa(2)$ respectively at 0.5, 0.01 and 0.001, and vary the central tendency by decreasing the number of virtual samples, to represent decreasing certainty in our parameter assignment.

Crowding sensors. We next explore how the performance of our techniques degrade as the sensors increasingly overlap in their "field of view." We implement this experiment using the *sensor-overlap* and *num-events* parameters. In all settings for this experiment we take the *sensor-overlap* to be 0.9. This value is significantly higher than in earlier runs, but we do not expect performance in Setting 1 of this experiment to differ greatly from Setting 1 of the earlier experiments because of their low values of *num-events*: 80% of the time there will be only a single event associated with a sensor, in which cases the *sensor-overlap* is not consulted at all. Later settings in this experiment sample higher values from *num-events*. In these later settings more attacks will be associated with sensors, increasing the possibility that MIFD cannot distinguish the true cause of sensor reports.

Base rate. Our fourth experiment addresses base rate issues: As the rate of true attacks goes down, how does our performance degrade? Base rate problems, where even a high accuracy sensor can perform unacceptably trying to detect very unlikely events, plague intrusion detection (Axelsson, 1999). We examine these issues by reducing the probability of generating an attack event as part of the ground truth for MIFD run. We do not actually change the κ(attack) value used by MIFD for its assessment step. We are instead introducing a discrepancy between the model and the actual attack event probability.

5 Results

Varying probabilities. Figure 3 shows MIFD's performance as the stratified likelihood rankings becoming less distinct. Although the theoretical basis of the qualitative probability levels is in fact infinitesimal, MIFD performs reasonably for $P(\kappa = 1)$ and $P(\kappa = 2)$ taken as high as 0.05 and 0.005 respectively. Moreover this level of performance is preserved under sensor-to-attack ratios of both 2 and 3. We had expected MIFD to be more brittle under sensor-to-attack ratio 2, but this was only partly true; although precision does decline more quickly with ratio 2 than with ratio 3, recall actually declines *less* quickly,

Qualitative abstractions of non point-value distributions. Figure 4 shows MIFD's performance when we interpret the qualitative likelihood levels as various beta distributions. The results are consistent with MIFD's performance on a point-value translation of the qualitative likelihoods to the values taken as these distriubtions' means.

Crowding sensors. Figure 5 shows how MIFD's accuracy declines when sensors correspond to more than one event, although both recall and precision are well above 90% through sensors responding to two distinct events. Unsuprisingly, precision declines sharply as sensors respond to three or more events. However this decline is not particularly serious in practice: actual IDS sensors rarely detect more than a single event.

Base rate. Figure 6 shows the challenge of base rates, and how MIFD aims to address it through sensor fusion. The figure shows that the $p(\text{false-alarm}|\text{alert})$ is extremely high, even with quite accurate sensors (low false alarm probability). The figure also shows that requiring corroboration, as MIFD does, from two

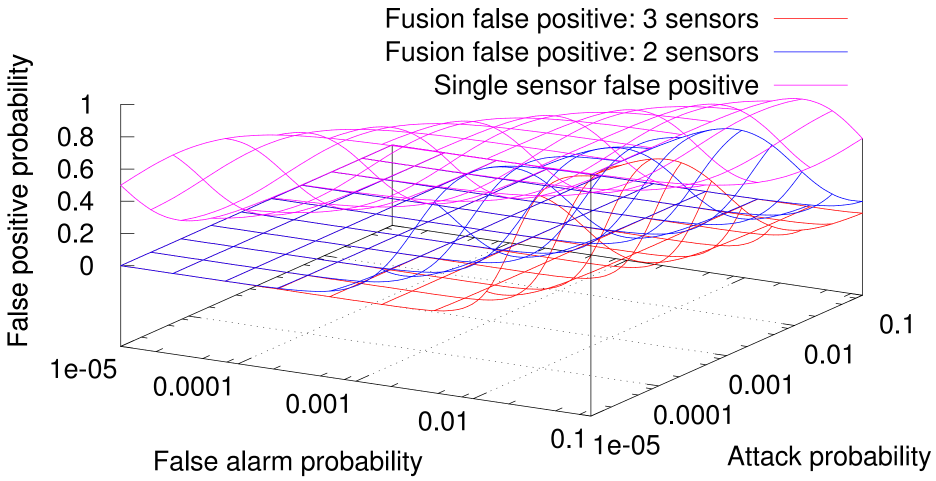


Figure 6: Base rate problems and sensor fusion.

or three sensors, can reduce the false alarm rate substantially. Note that this requires the sensors to fail independently. If multiple sensors' false positive rates are correlated, then corroboration can fail to achieve the objective of lowering the false positive rates. Note that it was specifically to handle such correlated failures that we introduced *benign events* into Scyllarus and MIFD.

Figure 7 shows how MIFD's false positive rate rises with actual event rarity. Later settings in this series have few actual attacks; MIFD does find them, but mistakenly hypothesizes additional attacks.

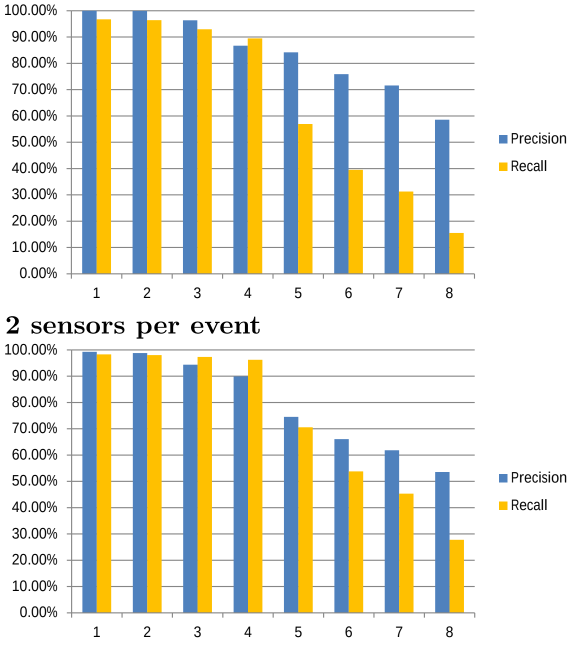
6 Conclusions

The results presented in this paper help to clarify why it is that Scyllarus and MIFD have been successful in practical deployments. They show that the qualitative Bayesian scheme they use is not very sensitive to the actual probabilities of events, and that its performance degrades gracefully. The results also hold out hope that IDS fusion can help tame the high false positive rates that plague the field of intrusion detection. In future work, we hope to extend our evaluation to consider MIFD's behavior when handling events for which topology is critical. In real networks, the spread of malware involves spread through network links, both actual communications links and superimposed networks like those induced by email address lists, etc. Now that we are working in an integrated framework, we can also use the results at hand to inform the design and fielding of intrusion detectors with a specific eye to their incorporation in a fusion system. Finally, we hope that our results will give encouragement to prospective users of qualitative schemes based on probabilistic reasoning.

Acknowledgments

This paper was supported by DARPA and the U.S. Air Force under contract number FA8650-11-C-7191. Approved for Public Release, Distribution Unlimited. *[The previous is the clearance expression we will put in upon clearance]* The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

3 sensors per event

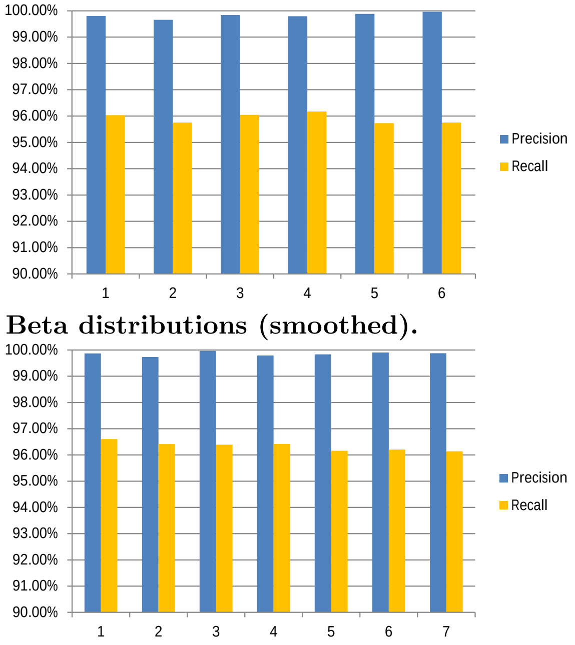

2 sensors per event

Settings

	$P(\kappa = 1)$	$P(\kappa = 2)$
1	0.005	0.0005
2	0.01	0.001
3	0.05	0.005
4	0.1	0.01
5	0.25	0.125
6	0.33333	0.22222
7	0.375	0.28125
8	0.46875	0.439453125

Figure 3: Results and settings of the **varying probabilities** experiment for sensor-to-attack ratios of both 2 and 3.

Beta distributions.


Beta distributions (smoothed).

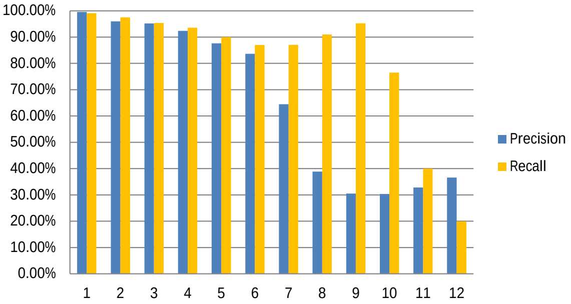
Settings

	Samples
1	10
2	5
3	4

	Samples
4	3
5	2
6	1

Figure 4: Results and settings of the **non point-value distribution** experiment. The mean values μ of the distributions translating $\kappa(0)$, $\kappa(1)$ and $\kappa(2)$ are respectively 0.5, 0.01 and 0.001; the sample sizes are as given above.

Crowding sensors.

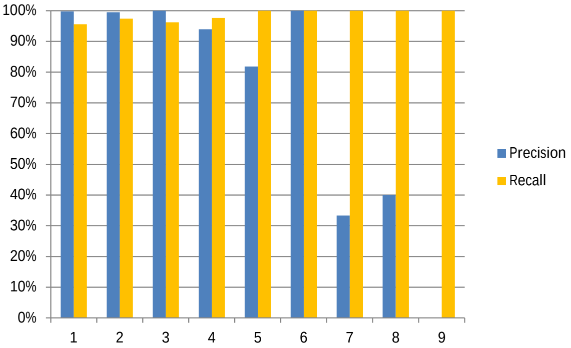


Settings

	Distribution of *num-events*		Distribution *num-events*
1	1 – 80%, 2 – 20%	7	3 – 80%, 4 – 20%
2	1 – 30%, 2 – 70%	8	3 – 30%, 4 – 70%
3	2 – 100%	9	4 – 100%
4	2 – 80%, 3 – 20%	10	4 – 80%, 5 – 20%
5	2 – 30%, 3 – 70%	11	4 – 30%, 5 – 70%
6	3 – 100%	12	5 – 100%

Figure 5: Results and settings of the **crowding sensors** experiment.

Base rate.



Settings

	Factor		Factor
1	1	6	0.005
2	0.5	7	0.001
3	0.1	8	0.0005
4	0.05	9	0.0001
5	0.01		

Figure 7: Results and settings of the **base rate** experiment. The key shows the factor by which we multiply modeled attack event probability to get the actual probability.

References

ArcSight (2008). Arcsight enterprise security manager. http://www.arcsight.com/product_info_esm.htm.

ARIS (2003). Attack registry & intelligence service. http://aris.securityfocus.com/AboutAris.asp. ARIS analyzer Data Sheet.

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*, CCS '99, pages 1–7, New York, NY, USA. ACM.

Charniak, E. and Goldman, R. P. (1988). A logic for semantic interpretation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 87–94.

Drew, S. (2014). What is the role of security event correlation in intrusion detection? Web page. SANS Intrusion Detection FAQ; date indicates when retrieved.

Finkle, J. and Heavey, S. (2014). Target says it declined to act on early alert of cyber breach. Reuters web site. http://www.reuters.com/article/2014/03/13/us-target-breach-idUSBREA2C14F20140313.

Forbus, K. D. and deKleer, J. (1993). *Building Problem Solvers*. MIT Press, Cambridge, Massachusetts.

Goldman, R. P. and Harp, S. A. (2009). Model-based intrusion assessment in common lisp. In *Proc. Int'l Lisp Conference*.

Goldman, R. P., Heimerdinger, W., Harp, S. A., Geib, C. W., Thomas, V., and Carter, R. L. (2001). Information modeling for intrusion report aggregation. In *DARPA Information Survivability Conference and Exposition(DISCEX-2001)*, pages 329–342. DARPA and the IEEE Computer Society.

Goldszmidt, M. and Pearl, J. (1996). Qualitative probabilities for default reasoning, belief revision and causal modeling. *Artificial Intelligence*, 84(1–2):57–112.

Lee, W., Mè, L., and Wespi, A., editors (2001). *Recent Advances in Intrusion Detection (RAID 2001)*, number 2212 in LNCS. Springer-Verlag.

Leydon, J. (2001). IDS users swamped with false alerts. *The Register*. http://www.theregister.co.uk/2001/12/15/ids_users_swamped_with_false/.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., Los Altos, CA.

Poole, D. (1993). Probabilistic horn abduction and Bayesian networks. *Artificial Intelligence*, 64:81–129.

Provan, G. (1989). An Analysis of ATMS-based Techniques for Computing Dempster-Shafer Belief Functions. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1115–1120. Morgan Kaufmann Publishers, Inc.

Schwartz, M. (2014). Target ignored data breach alarms. InformationWeek web site, https://www.informationweek.com/security/attacks-and-breaches/target-ignored-data-breach-alarms/d/d-id/1127712.

Sommer, R. and Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *Proceedings of the IEEE Symposium on Security and Privacy*.

Thayer, J., Burstein, M., Goldman, R. P., Kuter, U., Robertson, P., and Laddaga, R. (2013). Comparing strategic and tactical responses to cyber threats. In *SASO Workshop on Adaptive Host and Network Security AHANS*.

Valdes, A. and Skinner, K. (2001). Probabilistic alert correlation. In (Lee et al., 2001).

Vandoorselaere, Y. (2008). Prelude correlator. https://trac.prelude-ids.org/wiki/PreludeCorrelator. Prelude Correlator online documentation.

Vigna, G., Kemmerer, R. A., and Blix, P. (2001). Designing a Web of Highly-Configurable Intrusion Detection Sensors. In (Lee et al., 2001), pages 69–84.