"Why 6?" Defining the Operational Limits of stide, an Anomaly-Based Intrusion Detector

Kymie M.C. Tan and Roy A. Maxion
kmct@cs.cmu.edu and maxion@cs.cmu.edu

Dependable Systems Laboratory
Computer Science Department
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213 / USA

Abstract

Anomaly-detection techniques have considerable promise for two difficult and critical problems in information security and intrusion detection: detecting novel attacks, and detecting masqueraders. One of the best-known anomaly detectors used in intrusion detection is stide[1]*. Developed at the University of New Mexico, stide aims to detect attacks that exploit processes that run with root privileges. The original work on stide presented empirical results indicating that data sequences of length six and above were required for effective intrusion detection. This observation has given rise to the long-standing question, "Why six?" accompanied by related questions regarding the conditions under which six may or may not be appropriate.*

This paper addresses the "Why Six" issue by presenting an evaluation framework that maps out stide's effective operating space, and identifies the conditions that contribute to detection capability, particularly detection blindness. A theoretical justification explains the effectiveness of sequence lengths of six and above, as well as the consequences of using other values. In addition, results of an investigation are presented, comparing stide's anomaly-detection capabilities with those of a competing detector.

[1] Rather than STIDE or Stide or s-tide, we have chosen "stide" in keeping with the way the detector was referred to in the recent and frequently-cited paper by Warrender et al. [12].

1 Introduction

In a solid body of work inspired by the way the natural immune system distinguishes *self* from *other*, Forrest and her colleagues at the University of New Mexico [1, 2, 3] presented and analyzed the effectiveness of a detection scheme aimed at enhancing the security of computer systems. They saw computer system security as an instance of the more general problem of distinguishing self, e.g., the normal behavior of system programs, from other. A good example is the behavior of a trojanized system program (other) as opposed to the behavior of the same system program, but uncompromised (self).

Out of the efforts in computer immunology, evolved the detector that is now called "stide" (Sequence TIme-Delay Embedding). Stide's predecessor, the original self/other detector, was initially presented as a change-detection algorithm applied to the detection of computer viruses. It has since been applied to the task of detecting intrusions or exploits by way of detecting abnormal behavior in processes that run with root privileges on Unix systems. Stide operates on categorical data in the form of system kernel calls issued by the running process to the kernel of the host system. The reference to "time" in the name of the detector reflects the time-series nature of the categorical data upon which the detector was deployed.

Through the series of papers that have documented the many experiments aimed at studying the effectiveness of stide with respect to the detection of exploits and intrusions in Unix systems, the one curiosity that has been most conspicuous due to its significant impact on the performance of the detector, has been the ques-

tion of the "best" or most appropriate detector-window length or sequence length (used interchangeably) required in any application of the algorithm. For stide, this value is set *a priori*, and used to determine the length of all the sequences obtained from both training and test data. In the literature, we find that a sequence length of six is referred to consistently with stide by independent investigators [9, 5] and in experiments performed by the authors of the detector. For example, although a sequence length of 10 was finally settled on, in the results for the experiments in [4] it was observed that a sequence length of *at least* six appeared to be necessary in order to detect anomalies in all the intrusive data presented to the detector. Such an observation naturally prompts questions regarding the appropriate value of the detector-window parameter for stide, a problem that is not at all foreign to the community [5]; for example:

- Why does a detector-window length of six appear to work, while lengths less than six do not?
- Is a detector-window length of six appropriate for all data from differing environments?
- What is the impact on detection accuracy if an "incorrect" detector-window length is used?
- If not by "ad hoc means" [5], how else can the "best" detector-window length be determined?

That the value of the sequence-length parameter impacts the performance of the detector has not only been noted by the original authors, but also in subsequent, independent work [4, 5, 6, 9]. Marceau [6], noting stide's reliance on a "magic number," suggested a way of obviating the need for *a priori* selection of a fixed window length. She did not, however, address the issues of why or how fixed lengths of certain sizes affect stide's performance (nor was it her goal to do so). Lee & Xiang [5] proposed an information theoretic solution to the problem of choosing the optimal detector-window length for probabilistic anomaly detectors. We address their approach in Section 5.

The question of the appropriate detector-window length may have implications for aspects of detection other than performance. In particular, we were interested to know whether the results obtained by the original investigators represent a serendipitous match between the particular data sets used and the detector-window length. In other words, is six a necessary parameter value for this kind of detector, or simply sufficient for the data at hand? Answering questions of this nature is essential if we wish to avoid deploying detectors of this kind (anomaly detectors in general, not just stide) in environments where they may fail or be grossly ineffective.

2 Problem, approach and hypothesis

A long-standing issue with stide has been one of understanding why the number six is the "magic number" that makes stide work. People tend to dislike and distrust magic numbers, because they don't know how they are determined, and because they are not assured that for their situation the same magic number will effect the desired results. For example, Stillerman et al. [9, pp. 68] said, "We originally used a sliding window of length six, and later experimented with shorter window lengths. Somewhat to our surprise, a window length of two was just as good as a window length of six in detecting attacks." Their paper did not attempt an explanation.

The problem addressed in this paper is that of determining why six is the magic number that makes stide work. In addition, we take on the issue of what happens if that magic number is not set correctly in stide.

Our approach is to establish a framework of sequence types (rare, common and foreign), and within this framework to show how a very specific kind of anomaly, namely a minimal foreign sequence, affects the detection capabilities of stide.

Our hypothesis is this: a detector window of at least six was required to detect anomalies in all intrusive traces in the Hofmeyr et al. [4] dataset because the length of the smallest *minimal* foreign sequence present in one of the intrusive traces was six. An experiment is run to validate the hypothesis.

We begin by describing stide, and by replicating the essential finding of stide's creators – that six is the magic number. Through doing this, we establish the experimental foundation for drawing conclusions on the same basis as did the original authors of stide. We then address the idea, suggested by Lee and Xiang [5], that conditional entropy can be used to determine the best window size for probabilistic classifiers, and its implications for stide. This is followed by a brief exposition of a framework of sequence types, upon which a designed experiment is based, and offered in the next section. Experimental results are presented and discussed, to include the notion of different detectors being sensitive or blind to different phenomena. We then show how minimal foreign sequences can be found in the New Mexico data on which stide was originally run, and we end by answering the questions posed in Section 1.

3 Brief description of stide

Stide acquires a model of normal behavior by segmenting training data into fixed-length sequences [12]. This is done by sliding a detector window of length DW over the training data. Each length DW sequence obtained from the data stream is stored in a "normal

Data set	Program	Normal Data		Intrusion Data		
		No. of traces	No. of lines	Name of attack	No. of traces	No. of lines
1	Synthetic sendmail (UNM)	346	1,799,764	sunsendmailcp	3	1119
				decode	12	3067
				forwardingloops	10	2569
2	Synthetic sendmail (CERT)	294	1,576,086	syslogd	23	6504
				unsuccessful intrusion sm565a	3	275
				unsuccessful intrusion sm5x	8	1537
3	Synthetic ftp	8	180,315	wu.ftpd	5	1363
4	Synthetic lpr	9	2398	lprcp	1001	164,232
5	Live lpr (MIT)	2698	2,915,394	lprcp	1001	165,248
6	Live lpr (UMN)	1231	553,336	lprcp	1001	164,232

Table 1. Data sets used to replicate experiments documented in [4]; from University of New Mexico.

database" of sequences of length DW. A similarity metric is then used to establish the degree of similarity between the test data and the model of normal behavior obtained in the previous step. Sequences of length DW are obtained from the test data using a sliding window, and for each length DW sequence, the similarity metric simply establishes whether that sequence exists or does not exist in the normal database. A length DW sequence from the test data that is found to exist in the normal database (where "existing" requires that a sequence be found in the normal database that is an identical match for the sequence obtained from the test data), is assigned the number 0. Sequences that do not exist in the normal database are assigned the number 1. The decision is binary; either there is an exact match for a sequence from the test data in the normal database (0) or there isn't (1).

The detector's final response to the test data, manifested as the anomaly signal, involves a parameter known as the "locality frame". The locality frame is a value determining the length of a temporally local region over which the number of mismatches are summed up. For example, if the locality frame is set to 20, then at each point of the test data the number of mismatches in the last 20 (overlapping) sequences, including the current sequence, is determined. The number of mismatches that occur within a locality frame is referred to as the locality frame count (LFC). The locality frame count is the final value that is used to determine how anomalous the test data is. The length of the locality frame is a user-defined parameter that is independent of the length of the detector-window used to segment both training and test data.

4 Replicating the stide experiments

Hofmeyr et al. [4], in a study of intrusion detection using sequences of system calls, noted explicitly that "the best sequence length to use would be 6 or slightly larger than 6." Although a sequence length of six was used in earlier work (e.g., [2]), it is in the Hofmeyr et al. paper that the significance of six as a sequence length was made most explicit. The number six was obtained empirically.

To verify that our own investigation was on solid ground, and to ensure that our own hypotheses could be validated using the original datasets employed in the Hofmeyr et al. study, we began by replicating their work. The similarity metric used in Hofmeyr et al. was based on Hamming distances; stide employs a different metric. We used both metrics on the same data set to show that the question of why a length-six sequence works best remains pertinent in both cases.

The data on which the experiment was run was obtained from the University of New Mexico web site [11]. The data were comprised of six separate data sets, each containing normal and intrusion data. The data sets were derived from several different system programs. Table 1 provides summary information about the data.

The results presented in Figure 1 show the response of the detector that employed the Hamming-distance similarity measure. This graph only shows the curves associated with the sunsendmailcp and decode attacks, and mimics the graph of results presented in [4]. (The rest of the data are not shown, because they clutter the graph, and obscure its message, but are otherwise similar in that they indicate anomalies at sequence lengths less than six.) We found in our results, as the New Mexico team did, that there was an absence of an anomaly signal for sequence lengths of less than six for the system calls corresponding to process ID 283

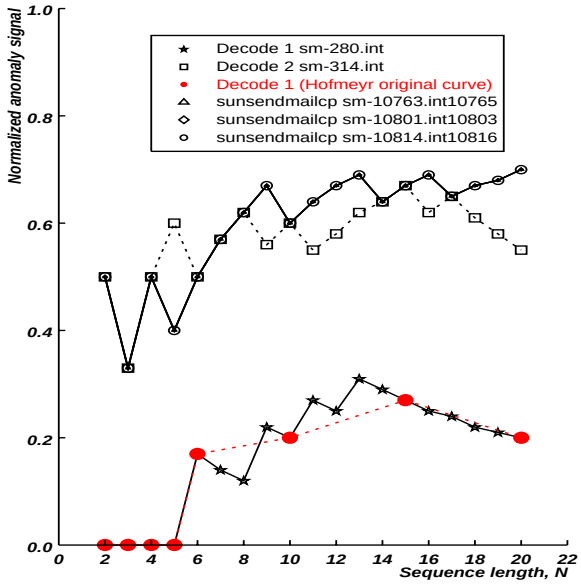

Figure 1. Normalized Hamming-distance similarity measure plotted against sequence length. Note that the first nonzero response to decode 1 occurs at sequence length six.

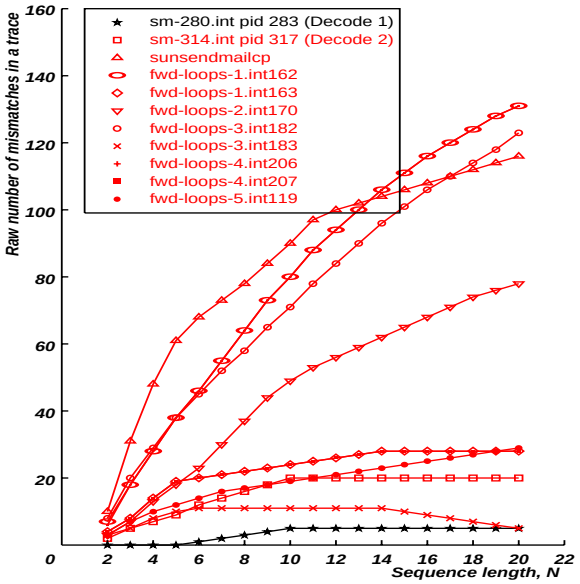

Figure 2. Stide mismatch response vs. sequence length. A length of at least six is required to detect anomalies in *all* intrusive traces; see decode 1 trace at bottom.

in the file labeled sm-280.int (decode 1).[2] We see this phenomenon echoed in the results for stide (Figure 2), where the decode 1 intrusion was not detectable for sequence lengths of less than six, either. These results show that for both metrics, a sequence length of six or greater was required to detect anomalies in all intrusive traces.

The graph in Figure 1 differs slightly from the graph presented in Hofmeyr et al. [4, Figure 2]. The graph in Figure 1 plots the Hamming distance result for *every* sequence length from 2 to 20. The similarity between our graph and Hofmeyr's graph becomes apparent once the starred data points in Figure 1 are removed. Despite slight differences in appearance, the essence of the two graphs is the same: namely that the first nonzero anomaly signal occurs at sequence length six for decode 1. The lowermost curve in Figure 1 is for decode 1, the data set that will later be shown to be the source of the Why-6 question. The large bullets connected by the dotted line depict the curve from [4, Figure 2].

[2] Two of the University of New Mexico files contain the original data for the decode attack. One of these is sm-280.int, which Hofmeyr refers to as decode 1; the other is called decode 2. Each file contains system calls corresponding to several PIDs.

5 Why 6, conditional entropy and stide

Of the work that has addressed the issues of selecting appropriate sequence length, that of Lee and Xiang [5] comes closest to our own. They suggested that the conditional entropy of intrusive sequences could be a key factor contributing to the use of six-symbol sequences in the New Mexico sendmail data. They plotted conditional entropy of the UNM `sendmail` data against sequence length in [5, Figure 1]; their graph shows that as sequence length increases, conditional entropy decreases; in particular, there is a distinct knee in their entropy curve corresponding to a sequence length of seven. They said, "conditional entropy drops to very small values after sequence length reaches 6 or 7." They show that for probabilistically-based classifiers, there is a relationship between the fall off in entropy and the appropriate window size for the classifier.

That conditional entropy could influence the selection of sequence lengths for stide seems like an appealing idea. Although Lee and Xiang [5] showed that conditional entropy may be useful for selecting appropriate sequence lengths for probabilistic classifiers, they did not attempt to extend the concept to stide. In this

section we will show that stide does not respond to changes in conditional entropy. We also offer an example suggesting that conditional entropy might not be a universal sequence-length selection metric, even for probabilistic classifiers.

To show that conditional entropy does not affect stide, we need to establish pairs of training and test data that differ *only* in terms of increasing irregularity (measured as conditional entropy) and nothing else. This means that the alphabet size, alphabet symbols and sample size are all kept constant, while irregularity is calculated to increase at fixed and steady intervals. We used 11 streams of training and test data pairs [7] that comply with these requirements. The data-generation process does not introduce anomalous sequences or symbols into the test-data stream. The reason for this is because introducing obviously-anomalous phenomena into the data stream would confound the results of the experiment; we would not know whether the detector was responding to the fluctuations in data regularity or to the presence of anomalous sequences.

The data pairs are labelled 1 to 11, and each pair differs from the preceding pair in terms of a measured increase in irregularity. The training and test-data pair labelled 1 are therefore the most regular, and the pair labelled 11 are completely random data. For details on the data-generation technique, which was based on state transition matrices, see [7, Section 4.3]. Into each of these 11 datasets is injected a single anomaly consisting of a single symbol not present in the training data. This is the simplest unequivocally anomalous event that stide can be expected to detect. The detector-window length for stide in this experiment was set to 2 to be consistent with the data generator in which the probability of each element depended only on the value of the previous element in the sequence.

Stide's locality frame count (LFC), which serves as a filter or smoother, was not used, because smoothing plays no role in primary detection, i.e., the initial decision as to whether an anomaly (mismatch, in stide) has been detected or not. For stide, a hit occurs when a mismatch is registered. In the case of our test data, this will occur whenever the anomalous character is within the detector window. We inject only singleton anomalies, not clusters or groups of anomalies. A singleton anomaly consists of two consecutive mismatches as the anomalous symbol passes through the detector window of length 2. An anomaly or mismatch anywhere else will be regarded as a false alarm, whether it occurs in some temporally local region or not; consequently, smoothing is not needed.

Figure 3 presents experimental results in terms of hits and false alarms. We can see that, given a situation in which everything was kept constant, including the type of anomalous phenomena introduced into the data streams, stide remained unaffected by the regularity increase from one data stream to the next, and continued to detect the anomalous symbol present in each of the 11 test-data streams. Stide achieved 100% hits and 0% false alarms. These results appear sensible, because stide has no notion of probability, and will only be affected by probability if some probabilistic phenomenon introduces anomalous sequences into the test data.[3] If the data-generation process does not introduce anomalous sequences, the fluctuations in data regularity itself, in isolation, make no impact on the ability of stide to detect the anomalous symbol. If data regularity, measured as conditional entropy, does not affect the stide detector, then it is highly unlikely that this aspect of categorical data would be the determining factor for the appropriate sequence length that must be employed by stide.

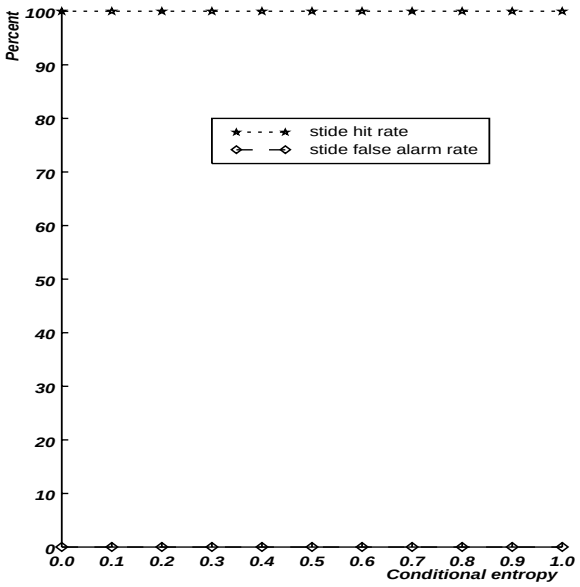

Figure 3. Hit and false-alarm rates for stide, as training/test data increase in irregularity (i.e., conditional entropy increases from 0 to 1). Note that stide's performance is unaffected by changes in conditional entropy.

There remains a certain curiosity about the results of Lee and Xiang [5]. While they showed that entropy falls off as a function of sequence length, and further-

[3] Other detectors *can* be affected by regularity; see [5, 7].

more demonstrated that this is so for the New Mexico data, we conducted a simple experiment which shows that the same kind of fall off can be seen even when the data are completely random (every alphabet symbol has an equiprobable chance of occurring at any point in the data stream). If the data are random, then it is difficult to imagine how any classifier could make sense of them. Hence, this calls into question whether or not conditional entropy can be used to establish window size, even for the probabilistically based classifiers discussed in [5].

For our simple experiment, we created two streams of random data. One stream contained 100,000 elements; the other contained 500,000. Lee and Xiang [5] had used the New Mexico data sets, specifically the sendmail data (numbers in parentheses indicate sample size): bounce-1.int (293), bounce.int (818), sendmail.int, (19526) queue.int (96330), and plus.int (98180). The largest of these contained about 98,000 elements; our sample size of 100,000 is simply rounded up from 98,000. Our other sample size, of 500,000, was arbitrary, simply to have a very large data set to compare against. Our alphabet size was 53, chosen because it was the largest alphabet size found in the UNM data. We used sequences sized 2 to 19, and calculated conditional entropy for each size, using the conditional-entropy formula provided in Lee and Xiang. Plotting conditional entropy against sequence length shows a fall off at sequence length six, where the curve drops to zero. The shape of the curve, with a knee at sequence length six, strongly resembles the curves obtained by Lee and Xiang in their plot of the New Mexico data, yet this curve was produced with completely random data. Hence, conditional entropy may not be a suitable sequence-length selection metric, even for probabilistic detectors. A conclusive answer to this question awaits further investigation.

6 Framework: Sequence types and hypothesis

The strength of the stide detection algorithm lies in its ability to detect foreign sequences – sequences not seen in its database of normal sequences. However, factors such as the relationship between the length of the sliding window and the length of the foreign sequence, as well as the effect of sliding a window over the foreign sequence, can make a significant impact on the detection capabilities of stide.

In order to describe these factors, we must first establish a framework within which such concepts can be expressed. Stide characterizes the normal behavior of a monitored process in terms of a database comprised of sequences of length DW. These sequences are obtained by sliding a window of this length along the trace (or traces) of system calls that have been obtained from the monitored process in the absence of intrusions. Stide only checks to see whether a test sequence is in the database or not. However, the frequency of the subsequences from which stide's normal-data sequences are *composed* is key to understanding why stide can be blind to certain sequences, as discussed further in Section 8.

To inform that discussion, we must first provide some definitions. A *rare* sequence is one that occurs infrequently in the training data. We arbitrarily define as rare those sequences having a frequency of occurrence less than or equal to 0.5% in the normal traces. *Common* sequences are those sequences occurring more frequently than 0.5%. *Foreign* sequences are those that do not occur at all in trace(s) that were used to define normal behavior. Note that a sequence can be foreign by virtue of containing:

- foreign symbols, i.e., symbols that are not contained in the alphabet set of the training data; or
- a foreign order of symbols, i.e., a sequence in which each individual symbol within the sequence is a member of the training-set alphabet, but where the order of the symbols is one that does not exist in the set of sequences obtained from the training data; or
- combinations of both.

In this work we focus specifically on the second condition, in which a foreign sequence is foreign by virtue of the foreign order of its constituent symbols.

The term *minimal foreign sequence* is defined as a foreign sequence of the second type, having the property that all of its proper subsequences already exist in the normal trace(s). Put simply, a minimal foreign sequence is a foreign sequence that contains within it no smaller foreign sequences. An in-depth exposition of the construction and characteristics of minimal foreign sequences can be found in [10].

We hypothesize that a detector-window length of at least six is required to detect all intrusive traces in the Hofmeyr data sets, and that this minimal foreign sequence of length six must have been composed of either rare or common subsequences. This would explain why neither stide nor the Hamming-distance detector detected this anomaly for detector-window lengths of less than six, i.e., because all the rare subsequences that make up the minimal foreign sequence of length six already exist in the training trace that comprised

the normal database, and consequently will not be seen by stide when using a window length less than six.

7 Experimental method

In this section we introduce the experimental regime for demonstrating the effect of minimal foreign sequences on the performance of two anomaly detectors: stide and a Markov-based detector. A detector based on Markov models [7] will be used as a tool for comparison in order to help illustrate that factors affecting stide may or may not affect another detector employing a different approach; what constitutes an anomaly for one detector is not necessarily an anomaly for another. We will refer to the detector based on Markov models as the Markov detector. The Markov detector employs conditional probabilities in its function as an anomaly detector. Briefly, it determines the probability of seeing an event, given the previous N events.

We showed in Section 5 that (ir)regularity in data, as measured by conditional entropy, does not affect the stide detection algorithm, nor can it be used to determine the appropriate detector-window length for stide. It is our hypothesis that the sizes of minimal foreign sequences in a given data set influence the appropriate detector-window size for stide. The following is an outline of the experimental procedure that we used to show this.

- Generate training data;
- Generate background test-data stream;
- Construct and select minimal foreign sequences of lengths 2 to 9, composed of rare subsequences, from the training data;
- Inject the minimal foreign sequences, as anomalies, into the background test-data stream to create ground-truth test data;
- Run both anomaly detectors (stide and Markov) on the same training and ground-truth test data, while varying their detector-window lengths with respect to the length of the injected anomalous sequence; record results.

7.1 Generating the training data

The training data were constructed using a Markov-model transition matrix. The method for generating the training data is documented in [7]. Although numbers were used to represent the elements of the training-data stream, the numbers were treated as categories.

The transition matrix used to generate the training data had a conditional entropy value of 0.1. This means that at each point in the data stream, the next element is highly predictable given the current element, i.e., there is low uncertainty as to what the next element will be. Such a transition matrix was chosen simply because it generated data with the following characteristics:

- A large proportion of the data consists of a repetition of the sequence 1, 2, 3, 4, 5, 6, 7, 8. Ninety-eight percent of a one-million-element data stream, generated with this transition matrix, will consist of a repetition of the sequence 1, 2, 3, 4, 5, 6, 7, 8. This results in a consistent set of obviously common sequences. A test-data stream made up of commonly-occurring sequences is desirable for allowing us to observe the response of a detector to the injected anomaly without being confounded by naturally-occurring rare or foreign sequences.
- Despite the repetition in a large portion of the data resulting in a usable set of common sequences, there is a small amount of unpredictability in the probabilities that populate the matrix which ensures the occurrence of the rare sequences necessary for selecting the constituent rare subsequences in a minimal foreign sequence.

The alphabet size for the training data was 8. We note that alphabet sizes in real-world data are certainly much higher than this; for example, there are about 243 unique kernel calls in BSM audit data. However, our method aims to evaluate the capabilities of a detector in detecting the higher-level concept of an anomaly. Although alphabet size may play a role with respect to certain aspects of the data, such as influencing the size of the set of possible foreign sequences or the size of the set of possible sequences that populate the normal database, a foreign sequence is still a foreign sequence regardless of the alphabet size, and the concept of a rare sequence will also remain immutable regardless of alphabet size. This abstraction allows us to study the response of the detector using synthetic data, as well as to apply the results from the synthetic environment to real-world environments.

The aforementioned matrix was used to generate a training-data stream of 1,000,000 elements. This sample size was an arbitrary choice, selected so that the data set would not be insufficiently small. There were two parameters that were chosen arbitrarily in this experiment: the sample size of 1,000,000 elements, and the length of the minimal foreign sequence (AS), which ranged from 2 to 9.

7.2 Generating background test data

The background data for the test-data stream consisted of the most commonly occurring sequences only, which given the training data described above, contained only repetitions of the sequence 1, 2, 3, 4, 5, 6, 7, 8. This ensured that only common sequences populated the background data. This was a desired property primarily because our aim was to observe the response of a detector to the specific minimal foreign sequence that we were intending to introduce into the background data in the second phase of this procedure. We therefore wanted background data that would not interfere with a detector's response by containing within it any obviously anomalous event that would constitute noise for a particular detector, for example naturally occurring rare or foreign sequences. To maintain consistency with the training data, the background test data was 1,000,000 elements long.

7.3 Characteristics of injected anomalies

We will be injecting an anomaly that consists of a minimal foreign sequence of length AS, composed of rare subsequences, into the data stream. As noted above, a rare sequence is defined to be a sequence that occurs less than or equal to 0.5% of the time in the training data.

The decision to select rare sequences was prompted by the expectation that the Markov detector will have the ability to detect rare sequences. In cases where the length of the detector-window is smaller than the length of the anomaly AS, we encounter the situation where the detector does not "see" all of the minimal foreign sequence at once. Instead, the detector is relegated to producing an anomaly signal based only on the smaller subsequences that form the larger minimal foreign sequence. Under such situations, we would like to observe the effect of the rare subsequences on the performance of both a Markov-based detector and stide. Although we already know that stide does not have the ability to respond to rare sequences, we will nevertheless apply the stide detector to an anomaly with these characteristics, primarily for the sake of charting and comparing the performance space of both detectors in an attempt to quantify how much more the ability to detect rare sequences actually confers upon the detection of foreign sequences under such circumstances.

7.4 Injecting minimal foreign sequences

The minimal foreign sequences and their constituent subsequences must now be chosen carefully so that the injection process itself does not introduce unintended perturbations in the background data. This is particularly significant with respect to sequence boundaries, i.e., where some elements of the injected anomalous sequence and some elements of the background data may combine within a detector's window to produce sequences that affect the anomaly detector in unintended ways. In particular, we want to avoid producing additional, undesired, foreign sequences due to the combination of symbols from the injected sequence and surrounding symbols from the trace.

We have determined that sequences composed by concatenating short, rare sequences from the training trace are likely to be foreign, simply due to the improbability that a substantial number of rare sequences would appear in the training trace in the chosen order. It is a simple matter to generate such sequences, and to verify their foreignness and minimality. These same properties complicate the problem of injecting the anomaly, which remains somewhat of an art. Essentially, the problem is one of ensuring that all of the $2(\mathrm{DW}-1)$ sequences of length DW that can be composed at the boundary of the injection, using contiguous symbols from the anomaly and the background trace, are actually in the database. If this is not the case for some location in the trace, a different anomaly is produced and the process repeated until successful.

The final suite of evaluation data contains one stream of training data and 8 streams of test data, where each test-data stream contains a single minimal foreign sequence whose length is selected from the range 2 to 9. This set of 9 data streams is then used repeatedly, once for each detector-window length of 2 to 15. Note that the length of the detector window dictates the length of the subsequences that compose each minimal foreign sequence. In total, we have 112 test data streams.

7.5 Running the detectors

We ran the stide and Markov based detectors on the suite of data created in the preceding sections. For each minimal foreign sequence being detected, we varied the length of the detector window from 2 to 15. It should be noted that for stide we ignored the locality frame count, focusing on the indication of a match (0) or mismatch (1). We reasoned that although further processing can be performed on the results of the similarity measure for purposes of smoothing away noise or enhancing signal strength, no amount of subsequent processing can compensate for the underlying inability to detect a specific phenomenon.

The locality frame count (LFC) sums up the num-

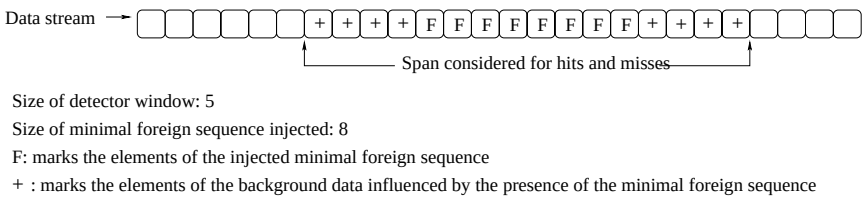

Figure 4. Detector's response to sequences within incident span is used to determine hits and misses.

ber of mismatches experienced within the span of the locality frame. Although the LFC does contribute to the final anomaly signal, it only comes into play *after* a sequence has been determined to be a match or mismatch. If the detection of a foreign sequence is missed, meaning that it does not register as a mismatch, then no amount of applying the LFC or adjusting its length will cause the missed anomaly to be detected.

7.6 Detection capabilities and detection blindness

The outcomes of our experiments are expressed in terms of hits and misses, and in terms of regions of detection blindness and detection equivocality, or weakness. A hit is a correct detection; a miss occurs when the detector failed to detect an injected anomaly. When a detector window slides over an anomaly, e.g., a foreign sequence, at various points of its journey it will view sequences that are composed of a combination of the elements from the foreign sequence and elements from the background data. Under such circumstances, the interaction between the elements of the foreign sequence and the background data will cause sequence types to arise that prompt the anomaly detector to respond in one fashion or another. Regardless of how the detector responds, the response is still influenced by elements of the foreign sequence. Only when the detector window completely clears the entire foreign sequence (i.e., no elements within the detector window belong to the foreign sequence), can we say that the response of the detector is no longer influenced to the foreign sequence. In other words, as long as some part of the foreign sequence is within the span of the sliding detector window, it can be argued that the detector's response is due to the presence of the foreign sequence in the data. As a result, the response of the detector in such a circumstance should also be considered in the process of determining hits or misses. This line of reasoning resulted in the concept of the incident span that we use to determine hits and misses. The incident span (see Figure 4) includes the DW-1 elements of the background data adjacent to the anomalous sequence on one side of the detector window, the AS elements of the anomalous sequence, and the $DW-1$ events of the background data adjacent to the anomalous sequence on the other side of the detector window. The length of this span is therefore $AS+2(DW-1)$ elements. Alternatively, it can be said that $AS+(DW-1)$ sequences of length DW are contained within the incident span.

In a situation in which only a single minimal foreign sequence anomaly is introduced into each test stream, and in which the detector response may range along a continuum from 0 (indicating completely normal) to 1 (indicating maximal abnormality), we describe a detector as:

- blind, in the case where the detector response is 0 for *every* sequence of the incident span;
- equivocal, in the case where the maximum detector response registered in the incident span is greater than 0 and less than 1, indicating that something not unequivocally normal has been seen;
- true, in the case where at least one detector response of 1 was registered in the incident span. (The term "true" connotes veridical, unambiguous detection.)

Binary detectors, such as the sequence-matching portion of stide, are only capable of generating responses of 0 or 1. Other detectors, such as the Markov detector described in this paper, can generate equivocal, or weak, responses. Equivocal responses can be converted to binary responses by applying a threshold that converts below-threshold responses to 0; others to 1. For the purposes of this work, the Markov detector was constrained by the aformentioned method to produce only a binary response, as does stide. In other

words, the threshold for the Markov detector was set to one so that only maximally anomalous (minimal foreign) sequences are registered as hits.[4]

8 Results

Figures 5 and 6 show the results of the experiment described above. They map the detection capability of the stide and Markov detectors with respect to an injected minimal foreign sequence composed of rare sequences.

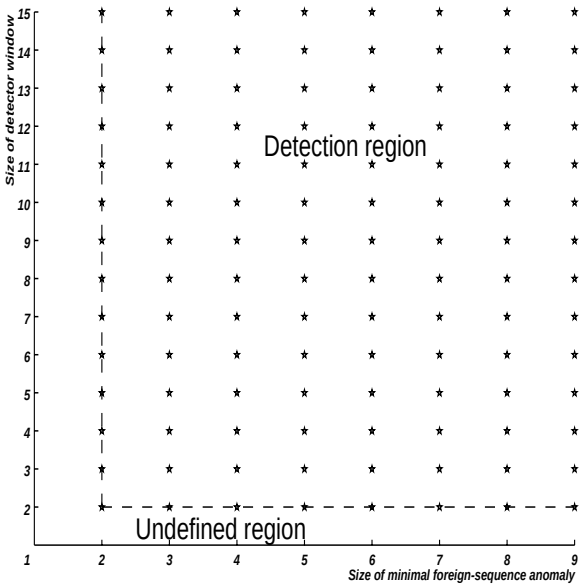

Figure 5. Markov detector efficacy.

The x-axis marks the increasing length of the minimal foreign sequence injected into the test-data stream, and the y-axis charts the length of the detector window required to detect a minimal foreign sequence of a given length. Each star marks the length of the detector window required to detect a minimal foreign sequence whose corresponding length is marked on the x-axis, where the term "detect" specifically means that a maximum anomalous response occurred in the incident span. The areas that are absent of a star indicate that the detector was unable to detect the foreign sequence whose corresponding length is marked on the x-axis, where unable to detect means that the maximum anomalous response recorded along the entire incident span was 0, signifying completely normal.

Since the Markov detector is based on the Markov assumption, i.e., that the next state is dependent only upon the current state, the smallest window length possible is 2. This means that the next expected, single, categorical element is dependent only on the current, single, categorical element. As a result, the y-axis marking the detector-window lengths in Figure 5 begins at 2. Although it is possible to run stide using a detector window of length 1, doing so would produce results that do not include the sequential ordering of events, a property that comes into play with all the detector-window lengths that are larger than 1. This, together with the fact that there is no equivalent on the side of the Markov detector, argued against running stide with a window of 1.

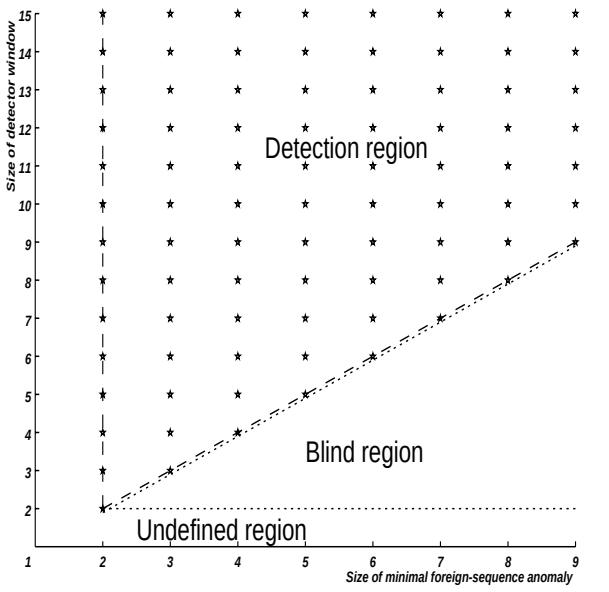

Figure 6. Stide detector efficacy.

The x-axis also begins at 2. This is because the type of anomalous event upon which the detectors are being evaluated requires that a foreign sequence be composed of rare sequences. A foreign sequence of length 1, therefore, would contain a single element that must be both foreign and rare at the same time, and this is not possible. As a consequence, both Figures 5 and 6 show an undefined region corresponding to the detector-window and anomaly length of 1.

[4] Detection thresholds are often used to determine "alarm-worthy" events. The most-anomalous detector response will always register as an alarm, regardless of where the detection threshold is set. An anomalous phenomenon generating such a response will never "disappear" or become a miss when the detection threshold is raised or lowered.

The results show that although the stide and Markov-based detectors both use the concept of a sliding window, and are both expected to be able to detect foreign sequences, their differing similarity metrics significantly affect their detection capabilities. There are three main points to note from the results. First, for stide, the detector-window length parameter must be greater than or equal to the length of the foreign sequence. The minimum length of the detector window required to detect each minimal foreign sequence is the size of the minimal foreign sequence itself. As can be seen from diagonal line in the results, the correlation between detector-window length and anomaly length is strong: $y = x$.

Second, the results show that the similarity metric used by each detector significantly affects detection performance. In stide's case, even though we know that there is a foreign sequence present in the data stream, this foreign sequence is only visible if the length of the detector window is at least as large as the length of the foreign sequence. The similarity measure employed by stide appears to have a weakness in that it is unable to detect minimal foreign sequences composed of rare subsequences under conditions where $DW < AS$. As a result, there are no guarantees that stide will detect faults even if they do manifest as foreign sequences in the data. The Markov detector, on the other hand, appears to have no such weakness. The minimal foreign sequence in the data stream is visible to the Markov detector, even when the length of the detector window is smaller than the length of the minimal foreign sequence. This suggests that there are factors in *this* data stream favoring detectors that employ conditional probabilities. These factors, however, appear to have no effect on the sequence-matching approach employed by stide.

Finally, by charting the performance of stide and the Markov detector with respect to the detection of minimal foreign sequences, we are able to observe the nature of the gain achieved in detection performance between an algorithm that employs conditional probabilities and one that employs the sequence-matching scheme used by stide. This gain in detection ability, due to the use of conditional probabilities, is significant and is illustrated by the blind region marked out in Figure 6.

These results provide evidence that shows a strong relationship between the length of the minimal foreign sequence and the length of the detector window required to detect such a phenomenon. It appears that the appropriate sequence length for stide is influenced by the length and composition of the minimal foreign sequences present in the data.

9 Minimal foreign sequences in data

In Section 4 we saw that something about the decode 1 intrusive trace caused both stide and the Hamming-distance-based detector to completely miss the anomalies present in the decode 1 intrusive trace when detector-window lengths of less than six were employed. In experiments with synthetic data, we found that such behavior is typical of both detectors in the presence of minimal foreign sequences composed of rare or common subsequences. Here we tie these observations together, and propose that the solution to the "why six" problem lies in the presence of a length-six minimal foreign sequence, composed of rare or common subsequences, in the decode 1 intrusive trace. Since no minimal foreign sequences exist in the decode 1 trace with lengths less than six, unlike all the other intrusive traces, no anomalies could be detected in the decode 1 trace when detector-window lengths of less than six were used. This meant that a detector-window length of six was necessary in order to detect anomalies in *all* intrusive traces, including decode 1.

Our task at this point is to identify the minimal foreign sequences that are present in the Hofmeyr data. We wish to elucidate characteristics, such as their constituent subsequences, and their various lengths. This serves several purposes:

- to show that the anomalous phenomenon (minimal foreign sequence) identified in this work actually exist in real-world data;
- to show that regardless of the data, i.e., synthetic or real-world, when the performance of a detector has been established with respect to the anomaly types described in [8], the performance results for a detector are immutable, and will persist reliably across datasets;
- to verify that, in the case of stide, it is the presence of minimal foreign sequences in the data stream that dictates the appropriate detector-window length;
- to solve the "why six" problem.

Table 2 lists the length and number of minimal foreign sequences present in intrusive traces decode 1, decode 2, sunsendmailcp and forwardingloops; these intrusive traces correspond to the synthetic sendmail (UNM) data set 1 of Table 1. Due to space constraints, Table 2 does not include all of the data sets shown in Table 1. Furthermore, results based on those data all show minimal foreign sequences of less than size six, and would hence be redundant. The point we wish

Length	dec.1	dec.2	snsndmailcp	flps-1.162	flps-1.163	flps-2.170	flps-3.182	flps-3.183	flps-4.206	flps-4.207	flps-5.119
1											
2		2	10	7	4	3	8	3	6	4	3
3		1	13	7	2	1	8	1	8	2	1
4			4	1	4	2		2	1	4	
5			2	2			2		2		
6	1	1									
7				2		2	1		2		
8						1					
9			1	1			1		1		
10											
11				1			1		1		
12											
13											
14											
15											
16											
17											
18											
19											
20							1				

Table 2. Number of minimal foreign sequences (MFS) of lengths 1 to 20 for each named intrusive trace in the UNM sendmail data. Empty cells indicate that no MFS of that length could be found in the trace. Note the single length-six MFS in the dec.1 (decode 1) column. The smallest MFS in every other trace is length 2. For stide to detect all intrusive traces in these data, a detector window of length at least six is required.

to emphasize is that decode 1 is the only intrusive trace that does not contain minimal foreign sequences of length less than six (see Table 2). This means that stide required a detector-window length of six in order to detect that single anomaly in decode 1, because there were no minimal foreign sequence anomalies of lengths *less* than six to detect in that intrusive trace. Upon further analysis of the single minimal foreign sequence in decode 1, we find that it is actually a minimal foreign sequence of length six, composed of rare subsequences. Its precise identification is:

```
Filename: sm-280.int283.
Start line number: 79
End line number: 84
Actual Sequence: 2, 95, 6, 6, 95, 5
Translated to system calls:
  fork, connect, close, close, connect, open
```

10 Discussion and conclusions

From the series of experiments described above, we have confirmed our hypothesis that a detector window with length at least six was required to detect anomalies in all intrusive traces used in [4], because the length of the smallest minimal foreign sequence present in one of the intrusive traces was six. We found that the intrusive trace labeled decode 1 contained a single size-six minimal foreign sequence, composed of rare subsequences. The rare subsequences meant that only when the detector-window was large enough to see the entire minimal foreign sequence could that sequence register as an anomaly.

We showed the effect of minimal foreign sequences, composed of rare subsequences, on stide's performance, and how their presence undermines the claim that stide will detect foreign or "unusual" sequences that occur in a stream of data. We have identified conditions under which stide is completely unable to detect the presence of foreign sequences in a data stream. Identifying minimal foreign sequences, and establishing their effect on stide, enabled us to provide a solution to the question of the "best" or most appropriate detector-window length to select in any application of the stide algorithm.

We have also shown that the performance characteristics established for stide on synthetic data remained pertinent across datasets. In this case, even when the detector was deployed on real-world data, we were able to explain its performance behavior using the lessons learnt for that detector on synthetic data.

We now return to the questions posed in Section 1.

Why does a detector-window length of six appear to work, while lengths less than six do not?

Lengths less than six were not large enough to see the single minimal foreign sequence (MFS) of size six in the decode 1 intrusive trace. That sequence was composed of subsequences that did not appear to stide to be anomalous, and so stide was unable to detect them. Note that if the data from decode 1 had been, for example, appended to the data from decode 2, then a window of length 2 would have sufficed for alarming on the intrusive behavior, because decode 2 contains minimal foreign sequences of length 2 (see Table 2). If the MFS in decode 1 had been larger than six, then stide's detector window would have had to be concomitantly larger to detect it, and in this case, stide's magic number would have been different.

Is a detector-window length of six appropriate for all data from differing environments?

A detector-window length of six is not necessarily appropriate for data from differing environments. Minimal foreign sequences, whose lengths have been shown to affect the selection of the appropriate detector-window length for stide, can be of any size. Stillerman et al. [9], for example, found that a window length of 2 was sufficient to detect anomalies in all of their intrusive data. The magic number six, for stide, arose as an artifact of two circumstances: the selection of stide as a detector (blind to rare sequences), and the particular data sets studied in the Hofmeyr et al. [4] work (the presence of a size-six minimal foreign sequence in one data set). If decode 1 hadn't been part of the New Mexico data environment, six would not have been the magic number.

What is the impact on detection accuracy if an incorrect detector-window length is used?

The term "incorrect" can mean two different things. It can mean that the window length is too short to detect a minimum foreign sequence of a particular size. In this case, such a sequence would be misclassified as normal instead of anomalous, resulting in a missed detection and inaccuracy in the detection results. The severity of this would depend on how many misses were incurred and on how serious the missed attacks were.

"Incorrect" can also mean that the window length is too long; that it perhaps far exceeds the length of the longest minimum foreign sequence. The consequence of this would be that more computing power is required to run the detection apparatus, but detection accuracy would not be affected.

If not by "ad hoc means" [5], how else can the "best" detector-window length be determined?

In regard to the work of Hofmeyr et al. [4], the "best" detector-window length is that length which results in at least one anomaly being detected in each intrusive trace. For stide, an anomaly is a foreign sequence. The size of a particular type of foreign sequence, a minimal foreign sequence, directly determines the "best" detector-window length for stide.

For a given set of data, the best detector-window length would be the largest of the set of smallest minimal foreign sequences in each intrusive trace. For example, in Table 2, there are 11 intrusive traces. The smallest minimal foreign sequence is length 2 in all of the intrusive traces except one. That exception was for decode 1, in which the size of the smallest minimal foreign sequence is six. The size-six minimal foreign sequence in decode 1 can be described as the largest of the set of smallest minimal foreign sequences obtained from each intrusive trace.

The best detector-window length is dependent on the size of the relevant foreign sequence. Foreign sequences are found only in test data (not training data – normal data has no foreign sequences), so it may not be possible, in stide's case, to determine the best detector-window length *a priori* based only on normal data.

As a final note, we remind the reader that we are assuming that the minimal foreign sequences we encountered in the real-world data actually are the manifestations of the intrusions of interest. We are currently not aware of any analysis that has established that the anomalies (minimal foreign sequences) present in the intrusive traces are directly attributable to the attacks that were deployed, rather than being due to some other event that occurred while data were being collected. This makes it hard to determine if the alarms raised upon detection of those minimal foreign sequences were hits or false alarms. It is possible that the single minimal foreign sequence of size six in the decode 1 trace was the result of insufficient training data.

These speculations raise a more general issue. To what extent can we establish a link between detectable anomalies and intrusive behaviors? How can we decide, *a priori*, what kind of a sensor stream is appropriate and what detector characteristics are likely to be well matched to the stream. For example, the decode 1 intrusion is characterized, in the UNM data, by exactly one minimal foreign sequence of length six. We have shown that stide, with a window size of less than six, cannot detect this particular incident. Are there intrusive scenarios that would produce minimal

foreign sequences with greater lengths? In a similar vein, given knowledge of the detector and the working definition of normal, i.e., the database, is it possible to either modify an attack so that its trace appears to contain only normal sequences, or so that it contains only minimal foreign sequences of length greater than the size of the detector window? We are beginning to investigate these questions, and preliminary results indicate that escaping detection in these ways is possible for stide-like detectors. We would like to extend these investigations to other anomaly-detection schemes.

In closing, we would like to acknowledge many people's observation that the stide algorithm is very simple. However, it is noteworthy that, despite its simplicity, it's performance characteristics have been little understood, and it was not known how to set its parameters. Interestingly, systems more complex than stide have not exhibited substantial performance benefits over stide, concomitant with their complexity [12]. Hence, it is worthy of study, particularly if what is learned will contribute to the knowledge of the more complex algorithms that are likely to be used in future intrusion-detection systems.

We repeat a remark from Lee and Xiang [5], because it states our sentiments as well as we could state them ourselves. "Although one may argue that our results are simple, obvious and unsurprising, we feel that it is very important to develop a formal framework, even just for stating and validating the obvious, so that the field of intrusion detection can progress more rapidly and rigorously."

11 Acknowledgements

This work was funded by the Defense Advanced Research Projects Agency under contracts F30602-99-2-0537 and F30602-00-2-0528; we are grateful to DARPA for their support. This research would not have been possible without the hard work and scientific generosity of Stephanie Forrest and her research group at the University of New Mexico in making the UNM datasets, detector and documentation readily available. Without those original resources, none of the current results could have been obtained. We thank Wenke Lee, of the Georgia Institute of Technology, for helpful discussions and comments. The input of the research team in the CMU Dependable Systems Laboratory has been of great value. John McHugh (SEI/CERT) was enormously kind and generous in sharing insight, suggestions and help.

References

[1] P. D'haeseleer, S. Forrest, and P. Helman. An immunological approach to change detection: algorithms, analysis and implications. In *IEEE Symposium on Security and Privacy*, 6–8 May 1996, Oakland, California, pages 110–119, IEEE Computer Society Press, Los Alamitos, California. 1996.

[2] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. A sense of self for unix processes. In *IEEE Symposium on Security and Privacy*, 6–8 May 1996, Oakland, California, pages 120–128, IEEE Computer Society Press, Los Alamitos, California. 1996.

[3] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri. Self-nonself discrimination in a computer. In *IEEE Symposium on Security and Privacy*, 16–18 May 1994, Oakland, California, pages 202–212, IEEE Computer Society Press, Los Alamitos, California, 1994.

[4] S. A. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. *Journal of Computer Security*, 6(3):151–180, 1998.

[5] W. Lee and D. Xiang. Information-theoretic measures for anomaly detection. In *IEEE Symposium on Security and Privacy*, 14–16 May 2001, Oakland, California, pages 130–143, IEEE Computer Society Press, Los Alamitos, California, 2001.

[6] C. Marceau. Characterizing the behavior of a program using multiple-length n-grams. In *New Security Paradigms Workshop*, 18-22 September 2000, Ballycotton, County Cork, Ireland, pages 101–110, ACM Press, New York, NY, 2001.

[7] R. A. Maxion and K. M. C. Tan. Benchmarking anomaly-based detection systems. In *International Conference on Dependable Systems and Networks*, 25-28 June 2000, New York, New York, pages 623–630, IEEE Computer Society Press, Los Alamitos, California, 2000.

[8] R. A. Maxion and K. M. C. Tan. Anomaly detection in embedded systems. *IEEE Transactions on Computers, Special Issue on Embedded Fault-Tolerant Systems*, 51(2):108–120, February 2002.

[9] M. Stillerman, C. Marceau, and M. Stillman. Intrusion detection for distributed applications. *Communications of the ACM*, 42(7):62–69, July 1999.

[10] K. M. C. Tan. *Defining the operational limits of anomaly detectors*. PhD thesis, The University of Melbourne, Melbourne, Australia, 2002 (forthcoming).

[11] University of New Mexico. Computer immune systems, data sets and software: Sequence-based intrusion detection. Internet: `http://www.cs.unm.edu/~immsec/systemcalls.htm`, February 2002.

[12] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *IEEE Symposium on Security and Privacy*, 9–12 May 1999, Oakland, California, pages 133–145, IEEE Computer Society Press, Los Alamitos, California, 1999.