On Widening the Scope of Attack Recognition Languages

Jon Doyle, Howard Shrobe, and Peter Szolovits

Massachusetts Institute of Technology
Cambridge, MA 02139

May 30, 2000
Revised July 13, 2000

Abstract: The Intrusion Detection (ID) community has developed numerous proposals for languages with which to describe signatures of attacks on computers and networks. By and large, these languages provide means for describing sequences of specific events indicative of attacks through their presence or absence in the history of some computational system. This note argues from examples that meeting the needs of information warfare requires significantly extending the expressive capabilities of attack description languages.

This work is supported by DARPA under contract F30602-99-1-0509.

1 Introduction

Intrusion Detection (ID) research efforts have explored two main methods of recognizing intrusions or attacks on computer systems and networks, namely signature recognition and anomaly detection. In signature detection, one examines the events occurring in the system under guard to determine if this history matches "signature" patterns that characterize types of attacks. In anomaly detection, one examines the statistics of the events and compares these statistics to see if they depart from tables of "normal" statistics.

While much progress has been made, experience has shown that neither of these methods offers much protection against serious adversaries. Signature recognition performs decently against unembellished repetitions of known attacks, but fails badly on novel attacks or attack variations. Such limitations make secrecy about the library of signatures a principal defensive maneuver, but a weak one. The size of the space of possible variations on known attacks means that an adversary determined to evade detection stands a good chance of succeeding if he simply invents new variations he has never heard of before. Anomaly detection, in turn, has performed poorly to date through variability of normal human activities and the consequential weakness of knowledge about the relevant reference classes. One does much better at predicting the behavior of a typist, compared to someone who only knows about keystrokes, if one knows first that the typist is typing English, and better still if one knows the typist is typing travel itineraries or newspaper sports score reports.

We maintain that some limitations experienced with signature and statistical methods stem not from the methods themselves but from reliance on inadequately expressive languages for describing significant patterns. A richer language, particularly one based on multilevel abstractions and mechanisms for expressing uncertainty in characterizing events, permits one to express more of the central and essential regularities and worrying abnormalities needed to analyze behavior properly. Such a language can increase the difficulty of evading signature or anomaly detection by restructuring the spaces of events in ways that lessen the likelihood that evasive attempts succeed.

This note presents a number of examples illustrating the utility of constructs embodied in the "trend template" language of Haimowitz and Kohane [1, 2]. These constructs combine the features of signature and anomaly detection methods to provide a richer description language of known utility for medical monitoring tasks. We expect that information warfare monitoring tasks will benefit from a still richer language combining and extending the descriptive capabilities of a trend template language with the strengths of extant attack recognition languages, and are currently developing a proposal for possible features of a common attack recognition language (CARL).

> EXAMPLE 1: An academic research laboratory maintains an ensemble of computers running a Visual Surveillance and Monitoring application. On

> January 12, 2001 several of the machines experience unusual traffic from outside the lab. Intrusion Detection systems report that several password scans were observed. Fortunately, after about three days of varying levels of such activity, things seem to return to normal; for another three weeks no unusual activity is noticed. However, at that time, one of the machines which is crucial to the application begins to experience unusually high load averages and the application components which run on this machine begin to receive less than the expected quality of service. The load average, degradation of service, the consumption of disk space and the amount of traffic to and from unknown outside machines continue to increase to annoying levels, but then level off.

What happened here? Hackers gained access to the application server by correctly guessing a password. Using this they had set up a public FTP site containing among other things pirated software and erotic imagery. The load on the server increased as word spread about this new transshipment site, and leveled off as demand saturated the machine.

One can describe the pattern of activity here in terms of activities during several temporal regions. First there was a period of attacks (particularly password scans). Then there was a "quiescent" period. Then there was a period of increasing degradation of service. Finally, there was a leveling off of the degradation but at the existing high level.

One can resolve this summary into finer details that give more insight and perhaps improve the likelihood of recognition. To do this, we describe the trends of average resource load levels and the average volume of traffic from external sites. During the initial attack and quiescence periods, the load levels stay roughly constant while the external site activity goes up and down, because the attacks themselves do not involve much effort. During the exploitation and saturation phase, the load average climbs to saturation well before the external site activity levels out, because a few initial misusers suffice to swamp the host while word of the site continues to spread to further misusers.

> EXAMPLE 2: The US State Department runs an application to protect a US embassy in Africa during a period of international tension. The Department's system administrators observe a variety of information attacks being aimed at the application server. At least some of these attacks are of a type known to be occasionally effective in gaining root access to machines like the server. These attacks are then followed by a period of no anomalous behavior other than a periodic low-volume communication with an unknown outside host.

It is quite possible that an intruder has gained root access to the server. It is also possible that the intent of the intrusion is malicious and political. It is less likely, but still possible, that the periodic communication with the unknown outside host is an attempt to contact an outside control source for a “go signal” that will initiate serious spoofing of the application.

One can again describe the pattern of activity here in terms of activities during several temporal regions, coupled with environmental information. First there was a period of attacks seemingly aimed at obtaining root access occurring during a period of heightened international tension related to application being run. Then there was a “false peace” period of no attacks (or merely normal attacks) coupled with periodic low-volume foreign communications.

2 Limitations of current attack recognition languages

These examples illustrate some of the limitations that current attack recognition languages appear to suffer. We mainly will pick on STATL [3], as it is one of the most clearly defined languages designed for use in attack recognition, but expect other languages may share some of its limitations. In fact STATL is an extensible language and, with the proper extensions, might not have the limitations we attribute to it here. If it or other extant attack recognition languages do not have these limitations, the purpose of this note will be served by stimulating first the description of how to effect the desired attack recognition with one of these languages, and second the determination of whether such languages provide the best means for doing so. For simplicity of exposition, we will proceed as though STATL does have the limitations it seems to have, and that other attack languages share these limitations.

Recognizing the attacks described in the examples involves characterizations of patterns that refer to abstract times and durations and to relations between abstract temporal intervals. Moreover, the examples highlight the substantial uncertainties involved in exactly when the component events occur. The initial period of increased attack levels in Example 1, for instance, represents a rise of attack volume above a fluctuating background level of “normal” attacks. One might come to some fairly definite identifications of this event in a forensic analysis well after the events have played out, but attempts to recognize the attack in progress will likely suffer significant doubt about just where the rise in attacks starts and ends.

STATL’s strength lies in describing sequential and conditional events, but it appears to provide only for concrete times and durations, and does not provide any easy or obvious way of expressing or relating abstract intervals or for expressing or grading uncertainties about when events start and end. Instead, it seems to tie down all events with specific times, durations, and unambiguous changes in system states.

The attack patterns illustrated in Examples 1 and 2 also refer to changes in statistical trends over the intervals in question, such as increasing, decreasing, or constant values. It is not clear that these trends find easy expression in STATL, though perhaps the language-extension mechanism provides the means to include derivatives and other mathematical operations on signals. Similarly, STATL lacks a way of talking about periodic signals, unless perhaps by talking about signals with a concrete period, or by means of a further extension.

Finally, the attack patterns also refer to causal relationships between events, or more generally, to non-statistical relationships between events such as intentionality. While reporting languages like CISL [4] include some vocabulary for reporting causation and intent, recognition methods based on inferring large-scale plans from many piecemeal intentional actions require inclusion of such relations in an attack language. Such recognition methods require somewhat richer vocabularies for intentional concepts than the skeletal concepts provided in CISL (see [5]).

3 Trend templates

Our answer to these limitations is to enrich the language in which one can characterize patterns. The starting point of such an enrichment follows Haimowitz and Kohane [1, 2], who developed a language of "trend templates" we will call TTL for expressing temporal patterns like those involved in the examples, along with methods for recognizing instances of trend templates in the stream. The key elements of TTL are as follows:

1. Landmark times. These may be concrete times (i.e., fully-specified points on the calendar), but often are abstract times that play some special role in the event being characterized.

2. Temporal intervals. These may be of specific or abstract durations constrained by relations to other intervals and to landmark times.

3. Temporal relations. These include the Allen [6] interval relations and others.

4. State constraints. These specify characteristics of objects during temporal intervals, such as constant values, increasing or decreasing values, shapes of curves, etc.

5. Regression functions. These model criteria for matching templates against data, and so describe means for deciding when events occur when uncertainty exists about starting and ending times.

4 Further critique and comparison

We now examine some examples of trend descriptions, mostly motivated by hypothetical command and control scenarios, that exercise or exceed the expressive powers of STATL and the original TT language.

It seems likely that the TT language can characterize events that cannot be similarly characterized in languages that provide for descriptions only in terms of specific times and interval durations. Specifically, it appears that one cannot use only concrete times and durations to express descriptions like the following:

A. An X event occurs during a Y event.

B. An X event follows a Y event, and the switch-over time occurred some time between 5 and 7 PM.

C. An X event overlaps and follows a Y event, with the overlap lasting at least 5 minutes.

Correlation of simultaneous event histories also highlights potential difficulties. STATL appears to focus on decomposition of histories into concatenated intervals during which states are constant. There may be a way to describe the multiple overlapping time-varying state constraints expressible in the TT language in these attack languages, but even if that is possible, it is not likely to be convenient given the sequential focus of the languages. Descriptions like the following provide targets for expression here:

D. The resource load activity stayed constant while the external site activity rose and fell, and then the resource load activity rose swiftly to saturation levels while the external site activity rose more gradually and saturated later.

E. The traffic volume through X has been increasing while the traffic volume through Y has been steady.

TTL does not necessarily cover all the concepts desirable in a robust attack language. In particular, it has no facility for expressing probabilistic information. One can easily think of CC2 monitors needing to refer to such information. Consider, for example, the following requests a commander might make of a threat-detection system.

F. Warn me if the probability of a class X attack in sector Y goes over 25%.

G. Warn me if the rate of increase of the probability of a class X attack in sector Y goes up more than 25% on an hour by hour basis.

H. Discount any threat correlator which reports attack probabilities that vary too much and too quickly over several five-minute periods.

Similarly, TTL made no explicit provision for expressing negative information (information about absence of events), nor for expressing information about the value of events. Examples here include:

I. No attacks are hitting target X or targets of class Y.

J. The attacks are increasingly on more important targets.

The original TT language also did not include a very rich language for describing waveforms or periodicity. Examples here include:

K. The external requests are oscillating at 1 Hz, with oscillation between larger and smaller volumes of requests taking the shape of a square wave (or sawtooth wave, etc.).

L. The frequency of attacks for which success possibly compromises the secrecy of our plan database is increasing.

M. The frequency of congestion (oscillation) has been increasing for the past day.

Neither STATL nor TTL language provide any way of keying recognition methods to the systemic properties of recognition subsystems. Examples of such expressions include:

O. The attacks on command resources are increasing.

P. The success rate of attacks has been decreasing.

Q. It is becoming harder to detect attacks; we are detecting fewer, even though traffic is up without any change in our own behavior.

R. Sensor X is operating at selectivity Y and sensitivity Z on its ROC curve.

S. The effectiveness of our defenses is decreasing; the fraction of attack attempts that cause compromises is increasing.

More generally, as noted earlier, attack recognition languages require means to key methods to intentions and other psychological properties of adversaries. Examples here include:

T. The intent of attack X is Y.

U. The attack hits some machine in every enclave, but appears to prefer NT hosts when they exist.

5 Conclusion

The examples and discussion above suggest that while current recognition languages provide many important capabilities, no single extant language provides all the capabilities one might desire in an effective attack recognition language.

We believe that the principal value of a core attack recognition language inheres in the set of descriptive and analytical concepts it provides right from the start.

Such was the advantage of Fortran over early assembler languages for mathematical programming, and such should be the aim of attack recognition languages. Thus even when the attack language provides an extension mechanism, the core language should exhibit richness sufficient to express many of the concepts identified in the preceding.

References

1. I. J. Haimowitz and I. S. Kohane, 1993. "Automated trend detection with alternate temporal hypotheses". In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pages 146–151, Chambery, France, 1993.

2. I. S. Kohane and I. J. Haimowitz, 1993. "Encoding patterns of growth to automate detection and diagnosis of abnormal growth patterns". Pediatric Research, 33:119A, 1993.

3. Steve T. Eckmann, Giovanni Vigna, and Richard A. Kemmerer, 2000. "STATL: An Attack Language for State-based Intrusion Detection". Dept. of Computer Science, University of California, Santa Barbara.

4. Rich Feiertag, Cliff Kahn, Phil Porras, Dan Schnackenberg, Stuart Staniford-Chen, Brian Tung (editor), 1998-2000. "A Common Intrusion Specification Language (CISL)", www.gidos.org.

5. Jon Doyle, 1999. "Some representational limitations of the Common Intrusion Specification Language". Laboratory for Computer Science, Massachusetts Institute of Technology, October 26, 1999. http://www.medg.lcs.mit.edu/projects/maita /documents/cc2/cisl/revised.txt

6. James F. Allen, 1983. "Maintaining Knowledge About Temporal Intervals", Communications of the ACM, 26:11, Nov. 1983, 832-843.