DEVELOPING CUSTOM INTRUSION DETECTION FILTERS USING DATA MINING

Chris Clifton
Gary Gengo

The MITRE Corporation
Bedford, Massachusetts

ABSTRACT

One aspect of constructing secure networks is identifying unauthorized use of those networks. Intrusion Detection systems look for unusual or suspicious activity, such as patterns of network traffic that are likely indicators of unauthorized activity. However, normal operation often produces traffic that matches likely "attack signatures", resulting in false alarms. We are using data mining techniques to identify sequences of alarms that likely result from normal behavior, enabling construction of filters to eliminate those alarms. This can be done at low cost for specific environments, enabling the construction of customized intrusion detection filters. We present our approach, and preliminary results identifying common sequences in alarms from a particular environment.

INTRODUCTION

Differentiating between authorized and unauthorized use is a difficult problem. Signatures of an intrusion may also match authorized use, resulting in false alarms. This poses difficulties, particularly when applying commercial intrusion detection systems in a military network. Military networks often face unique constraints – operation over wireless media, unique message traffic, different perceived threats, limited bandwidth, mobile and dynamic environment, robustness in the face of direct attacks on infrastructure – that lead to "normal" operation that is different from civilian networks. This results in an unacceptably high false-alarm rate from Intrusion Detection systems: normal behavior matches potential intrusion signatures in the systems.

MITRE, and the U.S. Army's Communications Electronics Command are using data mining to address this problem. We are approaching this from the perspective that we must build on, not supplant, an existing intrusion detection infrastructure. Our goal is to identify patterns of false alarms coming from intrusion detection systems. We are using generalized frequent episodes, a data mining technique, to analyze intrusion detection system output. This identifies common, recurring sequences of alarms for a given site. These can be manually analyzed to determine if they result from normal operations at that site. This will enable development of site-specific filters to reduce the flow of information from intrusion detection systems. Since the alarm sequences are known to be common, the reward (in terms of reduction in false alarms) is high.

INTRUSION DETECTION BACKGROUND

Many conventional intrusion detection systems are based on *attack signatures*: patterns of network traffic that match known or likely intrusions. This works well for "kiddy scripts", commonly available programs that exploit known security holes. However, identifying more advanced attacks requires more general signatures. These identify patterns of activity likely to be associated with an attack, rather than matches to known attacks. Such "speculative" signatures may also be triggered by normal operations, resulting in false alarms. Intrusion detection systems must make a sensitivity tradeoff between identifying all intrusions, and *not* identifying normal behavior as suspicious.

Commercial tools optimize this tradeoff for typical environments. For a "standard" environment, this results in an acceptable level of false alarms and missed intrusions. However, no environment is really standard. Military environments are often far from typical commercial systems, with unique environments and constraints such as those mentioned above. Although commercial intrusion detection systems allow some user flexibility in adjusting the tradeoff between detecting intrusions and false alarms, in many environments the false alarm rate remains unacceptably high.

Our approach is to develop *custom filters* that reduce the false alarm stream based on known "normal behavior" in a particular environment. We use commercial intrusion detection systems, but filter out produced alarms that fit a pattern caused by normal operation *at that site*. The difficulty with this approach is building these filters, and determining what is normal operation at a site. While much less costly than building a complete intrusion detection system, it still requires considerable human effort. Our approach to reducing this effort is to use data mining technology to discover alarms caused by normal operation.

Data mining has been used elsewhere for intrusion detection. The KDD-Cup competition [KDD99], for example, challenged entrants to learn to classify connections into "okay" and "intrusion", based on a sample of connections where the intrusions were identified. While a laudable goal, we feel this approach has two drawbacks:

1. The production of good "training data" (connections where **all** intrusions have been identified) is not a practical task in most environments. The expertise to do this is limited – and missing even a few intrusions (or marking normal connections as intrusions) can prevent data mining technology from learning a good classifier.
2. This approach ignores all of the good work that has gone into developing meaningful attack signatures, attempting to supplant this with machine learning.

The profile-based data mining approach of [MGJ99] deals with issue 1, but still ignores existing signature-based work. Our approach is to use data mining to augment existing systems (see Figure 1). Data mining will help to identify where those systems fail *in a particular environment* – concentrating the limited local resources to where they will do the most good. We will now describe the particular mining technique used, and describe how this fits into an operation environment.

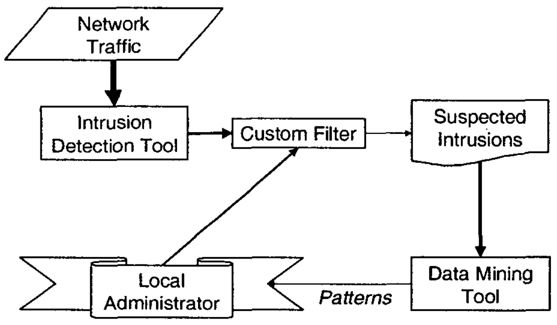


Figure 1: Developing custom filters with data mining.

SEQUENTIAL ASSOCIATION MINING

We develop filters based on sequences of alarms. The idea is that a sequence of operations that are normal in a particular environment may contain operations that look like a potential intrusion. However, the complete sequence is unlikely to be duplicated in an intrusion, so alarms that are part of the complete sequence can be ignored. The problem is in identifying such normal sequences.

We use *frequent episodes* [MTV97] to identify frequently occurring sequences of alarms. An episode is a sequence of alarms that occurs within a specified time window. A *frequent* episode is one that recurs in many time windows. The difficulty is that there may be interleaving operations that are unrelated to the episode – identifying frequent episodes in the presence of such noise is difficult (see Figure 2). This is where data mining technology comes in – it gives us the ability to find the most frequent episodes automatically and efficiently.

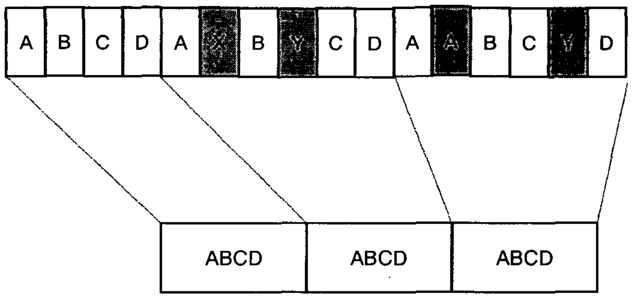


Figure 2: Sample frequent episodes.

Why is it important to find frequent episodes? Our goal is to identify sequences of alarms caused by normal operation. Frequent episodes are sequences of alarms that occur often – this gives us two things:

1. A common sequence of alarms is probably not the result of actual intrusion attempts – attackers will probably not repeatedly try the same method. However, normal operation is repeated. Therefore a frequently occurring sequence of alarms is a good candidate for having been caused by normal operation.
2. We expect a person to determine if a sequence results from normal operation. Analyzing *frequent* sequences is guaranteed to result in the largest reductions in the false alarm stream if the sequences can be filtered. Thus we are applying human effort where it does the most good.

We analyzed over one million intrusion detection alarms gathered from seven machines on a network over a two-week period. We loaded the logs into a relational database. The basic schema is the following.

Log(Event, FromIP, ToIP, time)

The standard frequent episode algorithm assumes a set of possible event types. However, we need additional flexibility. For example, normal operation may result in a single machine connecting to several machines in turn, with each connection causing a particular type of event. However, we would not want to filter this sequence unless all the connections came from the same machine. Recognizing such complex patterns requires additional flexibility, to get this we use *Query Flocks* data mining technology [Tsur98].

PRELIMINARY RESULTS

We analyzed these logs for 2-, 3-, 4- and 5-event sequences that occurred within a one-minute time window with the same FromIP and ToIP.

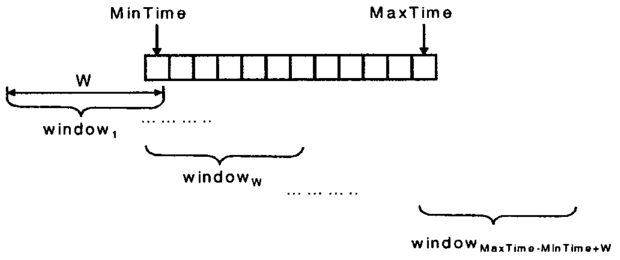
Here are some results on common 2-event sequences occurring in a one-minute time window. These sequences appeared the most number of times on the most number of machines. The results differed depending on how we counted the number of occurrences.

Straightforward Count

Command1	Command2	Count
SYN Flood	SYN Flood	5304224
ident	ident	1071661
print	print	563780
FTP User	FTP User	272894

This method of counting tends to favor sequences of same event as can be seen above. This is because when you are counting 2-sequences, you have to count every possible combination of event sequences. If an event repeats a lot in a small time frame (such as SYN Flood and ident), the count increases dramatically.

To reduce this skewing effect, we only count at most one occurrence per time window. Suppose the earliest event occurs at time MinTime, the latest event is at time MaxTime, and window is W. Then there are MaxTime-MinTime + W time windows in all. The earliest window is the window [MinTime-W, MinTime]. The latest window is [MaxTime, MaxTime + W].



When we count sequences this way, the results are very different.

Number of Occurrences(at most one per window)

Command1	Command2	Count
ftp	FTP User	305785
ftp	FTP Put File	101041
FTP User	FTP Put File	98232
FTP User	FTP Get File	58061
ftp	FTP Get File	55480

It's not surprising that these are common 2-event sequences, given that FTP is a fairly common event.

One advantage of counting sequences in this way is that the results can be easily normalized. For example, we can compute that the 2-event sequence (ftp,FTP User) appears in 14% of the time windows. This is just the Count divided by number of windows "covered" by the logs.

This method of counting is also useful in determining how significant a sequence is. For instance, we can compute that sequence (ftp,FTP User) occurs in 52% of the time windows that contain the "ftp" event. Since this sequence appears in only 14% of the time overall, we having established some correlation be the "ftp" event and the "FTP User" event.

CONCLUSIONS

Intrusion detection is a challenging problem. However, using data mining technology can help to address this problem. We have developed a technique that uses mining for sequential associations to identify common false alarms. This will enable construction of custom, site-specific filters that will improve the selectivity of intrusion detection systems. The ability to inexpensively develop custom filters for specific environments (such as the U.S.

Army's tactical environment) is particularly critical to support the unusual risks and constraints of military systems.

The work presented here is preliminary and ongoing. Although we have identified some interesting patterns, further work is needed to develop a data mining system that can be quickly and easily applied in a new environment, and produce intrusion alarm filters specific to that environment.

ACKNOWLEDGEMENTS

We would like to thank Marie Collins and Frank Geck of the MITRE Corporation and Pete Van Syckle of the Space & Terrestrial Communications Directorate – Special Projecs Office, US Army CECOM for their assistance with this work and in the development of this paper.

REFERENCES

[MGJ99] Ravi Mukkamala, Jason Gagnon, and Sushil Jajodia, "Integrating data mining techniques with intrusion detection," in *Proc. XIII Annual IFIP WG 11.3 Working Conf. On Database Security*, Seattle, WA, July 1999

[Tsur98] Dick Tsur, Jeffrey D. Ullman, Serge Abiteboul, Chris Clifton, Rajeev Motwani, Svetlozar Nestorov and Arnon Rosenthal, "Query Flocks: A Generalization of Association Rule Mining", in *Proceedings of the 1998 ACM SIGMOD Conference on Management of Data*, Seattle, WA, June 2-4, 1998.

[MTV97] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo, "Discovery of Frequent Episodes in Event Sequences", *Data Mining and Knowledge Discovery* 1(3): 259-289 (1997)

[KDD99] Charles Elkan, "Results of the KDD'99 Classifier Learning Contest", http://www-cse.ucsd.edu/users/elkan/clresults.html