

Data Mining Approaches for Intrusion Detection in Email System Internet-Based

Victor-Valeriu Patriciu, Liviu Rusu, Iustin Priescu
Military Technical Academy
 {vip, liviur, iustin}@mta.ro

Abstract

As the Internet grows at a phenomenal rate email systems has become a widely used electronic form of communication. Everyday, a large number of people exchange messages in this fast and inexpensive way. With the excitement on electronic commerce growing, the usage of email will increase more exponential.

In this paper we present our research in developing general method for intrusion detection in email system Internet-based. The main ideas are to use data mining techniques to discover consistent and useful patterns of email system that can recognize anomalies and known intrusions.

Key words: security, data mining, intrusion detection, email system

1 Introduction

As the Internet grows at a phenomenal rate email systems has become a widely used electronic form of communication. Everyday, a large number of people exchange messages in this fast and inexpensive way. With the excitement on electronic commerce growing, the usage of email will increase more exponential.

As network-based computer systems play increasingly vital roles in modern society, they have become the target of our enemies and criminals. Therefore, we need to find the best ways possible to protect our systems. Intrusion prevention techniques, such as user authentication (e.g. using passwords or biometrics) are not sufficient because as systems become ever more complex, there are always system design flaws and programming errors that can lead to security holes [2,4].

One useful method of classification for intrusion detection systems is according to general strategy for detection. There are two categories under this classification [2]:

- *misuse detection* - finds intrusions by looking for activity corresponding to known techniques for intrusion. This generally involves the monitoring of network traffic in search of direct matches to known patterns of attack (called signatures). This is essentially a rule-based approach. A disadvantage of

this approach is that it can only detect intrusions that follow predefined patterns;

- *anomaly detection* - the system defines the expected behavior of the network (or profile) in advance. Any significant deviations from this expected behavior are then reported as possible attacks. Such deviations are not necessarily actual attacks. The primary advantage of anomaly-based detection is the ability to detect novel attacks for which signatures have not been defined.

Another useful method of classification for intrusion detection systems is according to data source. There are two general categories under this classification:

- *host-based intrusion detection* - the data source is collected from an individual host on the network. Host-based detection systems directly monitor the host data files and operating system processes that will potentially be targets of attack. They can, therefore, determine exactly which host resources are the targets of a particular attack;

- *network-based intrusion detection* - the data source is traffic across the network. This involves placing a set of traffic sensors within the network. The sensors typically perform local analysis and detection and report suspicious events to a central location. Since such monitors perform only the intrusion detection function, they are usually much easier to harden against attack and to hide from the attackers.

Today's Internet security systems are specialized to apply a large range of techniques, usually knowledge-based (data mining), to an individual misuse detection problem, such as intrusion, virus or spam detection. Moreover, these systems are designed for one particular network environment, such as medium-sized network enclaves, and only tap into an individual cross-section of network activity such as email system activity [1].

Table 1 enumerates a range of Internet-based applications for enhancing security. These applications cover a set of detection, security and marketing programs that exists within the government, commercial and private sectors. Each of these applications are within the security capabilities techniques by applying data mining algorithms over appropriate audit data sources.

No.	Application:	Description and Variations:	Examples:	Audit Sources:
1.	Malicious email detections	Viruses Worm Spam		Email
2.	Intrusion Detection	Network-based detection Host-based detection Application-based detection	Standard IDS Less standard IDS Future IDS	TCP/IP System logs Application logs
3.	Fraud Detection	Unauthorized outgoing email Unauthenticated email Unauthenticated transactions	Console usurped Child attacks teacher Deceptive source Purchase / credit fraud	Email HTTP Transaction services
4.	User community discover	Closely connected user-base	Email circles	Email
5.	Pattern discovery	Account-based patterns Community-based patterns	Suspect activities Clandestine activities	Email, cookie_email, HTTP, TCP/IP, FTP, telnet
6.	Policy violation detection	ISP or enclave security policies	User espionage Outgoing Spam	All Email sources

Table 1 Internet-based applications for enhancing security

The central theme of our approach is to apply data mining techniques for intrusion detection in email system Internet-based. Data mining generally refers to the process of (automatically) extracting models from large stores of data [2]. The recent rapid development in data mining has made available a wide variety of algorithms, drawn from the fields of statistics, pattern recognition, machine learning, and database. Several types of algorithms [2] are particularly relevant to our research:

Classification: maps a data item into one of several pre-defined categories. These algorithms normally out-put “classifiers”, for example, in the form of decision trees or rules. An ideal application in intrusion detection will be to gather sufficient “normal” and “abnormal” audit data for a user or a program, then apply a classification algorithm to learn a classifier that will determine (future) audit data as belonging to the normal class or the abnormal class;

Link analysis: determines relations between fields in the database. Finding out the correlations in audit data will provide insight for selecting the right set of system features for intrusion detection;

Sequence analysis: models sequential patterns. These algorithms can help us understand what (time-based) sequence of audit events are frequently encountered together. These frequent event patterns are important elements of the behavior profile of a user or program.

Data mining refers to a process of non-trivial extraction of implicit, previously unknown, and potentially useful information from data. Examples of intrusion detection systems that use data mining include JAM (Java Agents for Meta-learning – W. Lee et al. 2000), MADAM ID (Mining Audit Data for Automated Models for Intrusion Detection – W. Lee et al. 2000) [2,4]. For example JAM [2],

developed at Columbia University, uses data mining techniques to discover pattern of intrusion. It then applies a meta-learning classifier to learn the signature of attacks.

The association rules algorithm determines relationships between fields in the audit trail records, and the frequent episodes algorithm models sequential patterns of audit events. Features are then extracted from both algorithms and used to compute models of intrusion behavior. The classifiers build the signature of attacks. So essentially, data mining in JAM builds a misuse detection model.

JAM generates classifiers using a rule learning program on training data of system usage. After training, resulting classification rules is used to recognize anomalies and detect known intrusions. The JAM system has been tested with data from *Sendmail*-based attacks.

2 Experiments with RIPPER on Sendmail Data

The procedure of generating the *sendmail* traces were detailed in [5]. Briefly, each file of the trace data has two columns of integers, the first is the process ids and the second is the system call “numbers”. These numbers are indices into a lookup table of system call names. For example, the number “5” represents system call *open*. The set of traces include:

Normal traces: a trace of the *sendmail* daemon and a concatenation of several invocations of the *send-mail* program;

Abnormal traces: 3 traces of the *sscp* (*sunsendmailcp*) attacks, 2 traces of the *syslog-remote* attacks, 2 traces of the *syslog-local* attacks, 2 traces of the *de-code* attacks, 1 trace of the *sm5x* attack and 1 trace of the *sm565a* attack.

These are the traces of (various kinds of) abnormal runs of the *sendmail* program [4].

System Call Sequences (length 7)	Class Labels
4 2 66 66 4 138 66	“normal”
...	...
5 5 5 4 59 105 104	“abnormal”
...	...

Table 2. Pre-labeled System Call Sequences of Length 7

In order for a machine learning program to learn the classification models of the “normal” and “abnormal” system call sequences, we need to supply it with a set of training data containing pre-labeled “normal” and “abnormal” sequences. We use a sliding window to scan the normal traces and create a list of unique sequences of system calls. We call this list the “normal” list. Next, we scan each of the intrusion traces. For each sequence of system calls, we first look it up in the normal list. If an exact match can be found then the sequence is labeled as “normal”. Otherwise it is labeled as “abnormal” (note that the data gathering process described in [5] ensured that the normal traces include nearly all possible “normal” short sequences of system calls, as new runs of failed to generate new sequences). Needless to say all sequences in the normal traces are labeled as “normal”. See Table 2 for an example of the labeled sequences. It should be noted that an intrusion trace contains many normal sequences in addition to the abnormal sequences since the illegal activities only occur in some places within a trace.

We applied RIPPER [3,4], a rule learning program, to our training data. The following learning tasks were formulated to induce the rule sets for normal and abnormal system call sequences:

- Each record has positional attributes p_1, p_2, \dots, p_n one for each of the system calls in a sequence of length n ; plus a class label, “normal” or “abnormal”;
- The training data is composed of normal sequences taken from 80% of the normal traces, plus the abnormal sequences from 2 traces of the attacks, 1 trace of the *syslog-local* attack, and 1 trace of the *syslog-remote* attack;
- The testing data includes both normal and abnormal traces not used in the training data.

RIPPER outputs a set of if-then rules for the “minority” classes, and a default “true” rule for the remaining class.

The following exemplar RIPPER rules were generated from the system call data:

- normal: $p_2=104, p_7=112$; meaning: if p_2 is 104 (*vtimes*) and p_7 is 112 (*vtrace*), then the sequence is “normal”;
- normal: $p_6=19, p_7=102$; meaning: if p_6 is 19 (*lseek*) and p_7 is 102 (*sigvek*), then the sequence is “normal”;
- ...;

- abnormal – true – meaning: if none of the above, the sequence is abnormal.

These RIPPER rules can be used to predict whether a sequence is “abnormal” or “normal”. But what the intrusion detection system needs to know is whether the trace being analyzed is an intrusion or not. We use the following post-processing scheme to detect whether a given trace is an intrusion based on the RIPPER predictions of its constituent sequences [4]:

1. Use a sliding window of length $(2l+1)$, 7, 9, 11, 13, etc., and a sliding (shift) step of l , to scan the predictions made by the RIPPER rules on system call sequences.
2. For each of the $(\text{length } 2l+1)$ regions of RIPPER predictions generated in Step 1, if more than l predictions are “abnormal” then the current region of predictions is an “abnormal” region. (Note that l is an input parameter)
3. If the percentage of abnormal regions is above a threshold value, say 2% then the trace is an intrusion.

Traces	% abn. [5] Frost	% abn. in experiments			
		A	B	C	D
sscp-1	5.2	41.9	32.2	40.0	33.1
sscp-2	5.2	40.4	30.4	37.6	33.3
sscp-3	5.2	40.4	30.4	37.6	33.3
syslog-r-1	5.1	30.8	21.2	30.3	21.9
syslog-r-2	1.7	27.1	15.6	26.8	16.5
syslog-l-1	4.0	16.7	11.1	17.01	13.0
syslog-l-2	5.3	19.9	15.9	19.8	15.9
decode-1	0.3	4.7	2.1	3.1	2.1
decode-2	0.3	4.4	2.0	2.5	2.2
sm565a	0.6	11.7	8.0	1.1	1.0
sm5x	2.7	17.7	6.5	5.0	3.0
sendmail	0	1.0	0.1	0.2	0.3

Table 3 Comparing Detection of Anomalies

RIPPER only outputs rules for the “minority” class. For example, in our experiments, if the training data has fewer abnormal sequences than the normal ones, the output RIPPER rules can be used to identify abnormal sequences, and the default (everything else) prediction is normal. We conjectured that a set of specific rules for normal sequences can be used as the “identity” of a program, and thus can be used to detect any known and unknown intrusions (anomaly intrusion detection). Whereas having only the rules for abnormal sequences only gives us the capability to identify known intrusions (misuse intrusion detection).

We compare the results of the following experiments that have different distributions of abnormal versus normal sequences in the training data:

Experiment A: 46% normal and 54% abnormal, sequence length is 11;

Experiment B: 46% normal and 54% abnormal, sequence length is 7;

Experiment C: 46% abnormal and 54% normal, sequence length is 7;

Experiment D: 46% abnormal and 54% normal, sequence length is 7;

Table 3 shows the results of using the classifiers from these experiments to analyze the traces. We report here the percentage of abnormal regions (as measured by our post-processing scheme) of each trace, and compare our results with Forrest et al., as reported in [5].

From Table 3, we can see that in general, intrusion traces generate much larger percentages of abnormal regions than the normal traces. We call these measured percentages the “scores” of the traces. In order to establish a threshold score for identifying intrusion traces, it is desirable that there is a sufficiently large gap between the scores of the normal sendmail traces and the low-end scores of the intrusion traces. Comparing experiments that used the same sequence length, we observe that such a gap in A 3.4, is larger than the gap in C, 0.9 and 1.19 in B is larger than 0.7 in D.

The RIPPER rules from experiments A and B describe the patterns of the normal sequences. Here the results show that these rules can be used to identify the intrusion traces, including those not seen in the training data, namely, the *decode* traces, the *sm565a* and *sm5x* traces. This confirms our conjecture that rules for normal patterns can be used for anomaly detection.

The RIPPER rules from experiments C and D specify the patterns of abnormal sequences in the intrusion traces included in the training data. The results indicate that these rules are very capable of detecting the intrusion traces of the “known” types (those seen in the training data), namely, the *sscp-3* trace, the *syslog-remote-2* trace and the *syslog-local-2* trace. But comparing with the rules from A and B, the rules in C and D perform poorly on intrusion traces of “unknown” types. This confirms our

conjecture that rules for abnormal patterns are good for misuse intrusion detection, but may not be as effective in detecting future (“unknown”) intrusions.

3 Conclusions

In this paper we present our research in developing general method for intrusion detection in email system Internet-based. The main ideas are to use data mining techniques to discover consistent and useful patterns of email system that can recognize anomalies and known intrusions. This framework consists of classification, association rules, and frequencies episodes programs that can be used to (automatically) construct detection models.

The experiments on *sendmail* system call data demonstrated the effectiveness of classification models in detecting anomalies. The accuracy of the detection models depends on sufficient training data and the right feature set. We suggested that the association rules and frequent episodes algorithms can be used to compute the consistent patterns from audit data.

4 References

- [1] S.J. Stolfo, S. Hershkop, K. Wang, O. Nimeskern, and C.W. Hu, *Behavior Profiling of Email*, First NSF/NIJ, ISI, 2003;
- [2] W. Lee, S.J. Stolfo, K.W. Mok, *Algorithms for Mining System Audit Data*, in Proc. KDD, 1999;
- [3] W.W. Cohen, *Fast Effective Rule Induction*, in 12th Conference on Machine Learning, CA, 1995;
- [4] W. Lee, S. Stolfo, *Data Mining Approaches for Intrusion Detection*, in 7th Usenix Security, 1998;
- [5] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, *A sense of self for unix processes*, in Proc. of the 1996 IEEE SSP, p. 120–128, CA, 1996;
- [6] V.V. Patriciu, I. Priescu, *Using Data Mining Techniques for increasing Security in E-mail System Internet-based*, in 11th Conference CAIM, Oradea, 2003.