An Enhanced Data Mining Based Intrusion Detection System (IDS) using Selective Feedback

Ajayi Adebowale
Idowu S.A
Babcock University, Ilishan-Remo, Ogun State, Nigeria

Abstract— **Intrusion detection systems aim to identify attacks with a high detection rate and a low false alarm rate. Data mining helps in identifying implicit and sometimes long patterns in network traffic data and consequently stating valid bounds for network traffic. Classification-based data mining models for intrusion detection are often ineffective in dealing with dynamic changes in intrusion patterns and characteristics, making it imperative for them to become adaptive to the flow of traffic going through the network. There must be a continuous learning on the part of the IDS so it can train itself to identify false negatives that were overlooked before in a certain period of time.**

This study explored the use of selective feedback to improve the efficiency of C4.5 (a data mining based research IDS) by using some algorithms based on machine learning paradigms namely, "smart learners", "incremental learners" and "meta learners". Using C4.5 as the IDS in the framework for Intrusion detection and the NSL-KDD dataset to represent real streaming network traffic, several experiments were performed and an evaluation of the classifier's performance was done using the confusion matrix and classification errors as the evaluation metric.

***Key Words*— Data mining, Intrusion detection systems, Smart learners, Meta learners, C4.5, NSL-KDD.**

I. INTRODUCTION

As network-based computer systems play increasingly vital roles in modern society, they have become an ever increasing target of intrusions. In addition to intrusion prevention techniques, such as user authentication and authorization, encryption, and defensive programming, intrusion detection is often used as another wall to protect computer systems [8]. Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network. Intrusions are caused by attackers accessing the systems from the Internet, authorized users of the systems who attempt to gain additional privileges for which they are not authorized, and authorized users who misuse the privileges given them. Intrusion Detection Systems (IDSs) are software or hardware products that automate this monitoring and analysis process.

The two main intrusion detection techniques are misuse detection and anomaly detection. Misuse detection systems, for example, IDIOT [7] and STAT [6], use patterns of well known attacks or weak spots of the system to match and identify known intrusions [8]. For example, a signature rule for the "guessing password attack" can be "there are more than four failed login attempts within two minutes." Misuse detection techniques in general are not effective against novel attacks that have no matched rules or patterns yet. Anomaly detection (sub)systems, for example, the anomaly detector of IDES [11], flag observed activities that deviate significantly from the established normal usage profiles as anomalies, that is, possible intrusions. For example, the normal profile of a user may contain the averaged frequencies of some system commands used in his or her login sessions. If for a session that is being monitored, the frequencies are significantly lower or higher, then an anomaly alarm will be raised. Anomaly detection techniques can be effective against unknown or novel attacks since no a priori knowledge about specific intrusions is required. However, anomaly detection systems tend to generate more false alarms than misuse detection systems because an anomaly can just be a new normal behavior. Some IDSs, for example, IDES and NIDES [2], use both anomaly and misuse detection techniques. While accuracy is the essential requirement of an IDS, its extensibility and adaptability are also critical in today's network computing environment. There are multiple "penetration points" for intrusions to take place in a network system. For example, at the network level carefully crafted "malicious" IP packets can crash a victim host; at the host level, vulnerabilities in system software can be exploited to yield an illegal root shell. Since activities at different penetration points are normally recorded in different audit data sources, an IDS often needs to be extended to incorporate additional modules that specialize in certain components (e.g., hosts, subnets, etc.) of the network systems. The large traffic volume in security related mailing lists and Web sites suggests that new system security holes and intrusion methods are continuously being discovered. Therefore IDSs need to be adaptive in such a way that frequent and timely updates are possible.

II. DATA MINING AND INTRUSION DETECTON SYSTEMS

Currently building an effective IDS is an enormous knowledge engineering task. System builders rely on their intuition and experience to select the statistical measures for anomaly detection [11]. Experts first analyze and categorize attack scenarios and system vulnerabilities, and hand-code the corresponding rules and patterns for misuse detection. Because of the manual and ad hoc nature of the development process, current IDSs have limited extensibility and adaptability. Many IDSs only handle one particular audit data source, and their updates are expensive and slow [1]. These limitations have led to an increasing interest in intrusion detection techniques based upon data mining.

Data Mining is the automated process of going through large amounts of data with the intention to discover useful information about the data that is not obvious. Useful information may include special relations between the data, specific models that the data repeat itself, specific patterns, and ways of classifying it or discovering specific values that fall out of the "normal" pattern or model. Given its success in commercial applications, data mining holds great promise for the development of tools for gaining fundamental insights into the network traffic data, thereby allowing system administrators and network engineers to automatically detect emerging cyber attacks [9] [4]. Various data mining techniques have been proposed to enhance IDS. This study however explored the use of selective feedback to improve the efficiency of C4.5 (a data mining based research IDS) by using some algorithms based on machine learning paradigms namely, "smart learners", "incremental learners" and "meta learners".

While accuracy is the essential requirement of an IDS, its extensibility and adaptability are also critical in today's network computing environment. Classification-based data mining models for intrusion detection are often ineffective in dealing with dynamic changes in intrusion patterns and characteristics, making it imperative for them to become adaptive to the flow of network traffic [10]. This study explored the use of algorithms based on machine learning paradigms to provide a feedback to C4.5 so it can train itself to identify false negatives that was overlooked before in a certain period of time.

III. C4.5

Classifier systems play a major role in machine learning and knowledge-based systems, and Ross Quilan's work on **ID3** and **C4.5** [13] is widely acknowledged to have made some of the most significant contributions to their development.

C4.5 starts with large sets of cases belonging to known classes. The cases, described by any mixture of nominal and numeric properties, are scrutinized for pattterns that allow the classes to be reliably discriminated. These patterns are then expressed as models, in the form of decision trees or sets of if-then rules that can be used to classify new cases, with emphasis on making the models understandable as well as accurate. The system has been applied successfully to tasks involving tens of thousands of cases described by hundreds of properties. It also deals with typical problems such as missing data and over hitting.

Ths study uses a java implementation of C4.5 (J48 in Waikato Environment for knowledge Analysis) as the research IDS.

IV. SMART LEARNERS

With an increase in the number of attacks on networks, there has been a need for understanding those attacks individually in their own capacity [3]. This serves as motivation for expert learners wherein we try to build classifiers, expert in their own way to identify different attacks. Expert Learners are like dedicated services providing signatures for only a particular type or class of attacks [5]. Such experts are very useful in generating rules which might be missed in dealing with hierarchies of attack classes. In this study two classes of smart learners were considered. One of them was building classifiers for individual attacks at a lower level of hierarchy e.g. satan, smurf and the other was trying to build classifiers at a higher level of hierarchy wherein we club together all records that belong to a super class of attack e.g. DOS, and R2L.

V. METALEARNERS

Meta learning is the study of principled methods that exploit meta-knowledge to obtain efficient models and solutions by adapting machine learning and data mining processes. While the variety of machine learning and data mining techniques now available can, in principle, provide good model solutions, a methodology is still needed to guide the search for the most appropriate model in an efficient way [12]. Meta learning provides one such methodology that allows systems to become more effective through experience. Meta-learners in conjunction with expert learners can provide feedback about known attacks as well as unknown attacks by analyzing the flow of traffic and confusion matrices for the smart learners.

VI. EXPERIMENTAL METHODOLOGY

A. The Dataset

The dataset used in this research is the NSL-KDD dataset. NSL-KDD is a data set suggested to solve some of the inherent problems of the KDD cup'99 data set. It is basically a processed version of the KDD cup'99 dataset. This dataset enables researchers to train their algorithms on the full dataset (because of its smaller amount of records) instead of using a portion of the full dataset as in the case of the KDD cup'99 data set. More information about this dataset can be found in [14].

B. Data mining Tool

Our experiments were done using Weka 3.6.7. Weka (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection.

VII. SUMMARY OF RESULTS

Preliminary tests involve finding out how C4.5 performs on the kddcup dataset without any tuning to either the training dataset or the way C4.5 classifies the test records. The training dataset contained the full training dataset (125973 records) and the test dataset contained the full test dataset (22544 records) labeled as in the original form. The results were as below:

Basic Classifier Results

Records	Errors
125973	20 (0.0156%)
22544	2069(9.17889%)

Results of expert classifier (super classes) on NSL-KDD dataset

	Total records	DOS	U2R	R2L	Probe	Normal
Training	125973	45927	52	995	11656	97278
Test	22544	5741	39	5993	2377	60593
On Training						
Tree Size		18135	17924	18126	17025	1365
Errors		4598	4587	4583	4599	40
Correct		121375	121386	121390	121374	493981
Detection Rate		100.00 %	100.00 %	100.00 %	100.00 %	99.9%
On Test						
Tree Size		17924	17924	401	17025	1365
Errors		902	8227	1128	8653	20289
Correct		21642	14317	21416	13891	2255

In the first part of the experiment a bit vector for the five super classes dos, u2r, r21, probe and normal was created based upon their performance as super classes. Then an analysis was performed on the same to come up with a technique for the feedback of select records to the main classifier.

As a second part to this experiment, we decided to add 10% of each of the records belonging to "R2L", "normal" and "unknown" that were classified as normal to the training dataset of the expert classifiers of "R2L" and "Normal" classes and see their performance. The added 760 records were chosen at random from the three selected group of records. The results were averaged over 10 runs and are summarized in the table below.

Table 4.7
Results for enhanced r21, normal, unknown experts using meta-classifier feedback

Total records	Total	R2L	Normal	Unknown
Training	125973	995	97278	18729
Test	22544	5993	60593	18729
On Training				
Size tree		326	1589	1330
Errors		198	296	48276
Correct		125775	125677	77697
On Test				
Errors		92	1328	420
Correct		22152	21216	22124
		98.26%	94.11%	98.14%

Table 4.8
Results for unknown expert using meta-classifier feedback

Total records	Total	Normal	Unknown
Training	125973	97278	18729
Test	22544	60593	18729
On Test			
Errors	10287	10156	282
Correct	300742	50437	18447
Detection	97%	83%	98%

As can be seen, the classifier does significantly well as compared to the basic classifier which has a low detection rate. Also when tested against only the 18729 of the unknown records, it correctly classifies most of them as "unknown" leaving only 282 records as misclassified in the average case.

VIII. CONCLUSION

The central theme of our approach to intrusion detection is to apply data mining programs to the extensively gathered audit data to compute models that accurately capture the actual behavior (i.e., patterns) of intrusions and normal activities. This approach significantly reduces the need to manually analyze and encode intrusion patterns, as well as the guesswork in selecting statistical measures for normal usage profiles. Furthermore, data mining programs can be applied to multiple streams of evidence, each from a detection module that specializes in a specific type(s) of intrusion (Expert classifiers) or a specific component of the network system (e.g., a mission critical host) to learn the combined detection model that considers all the available evidence (Metalearning). Therefore by adding a metalearning feedback loop to an IDS framework, IDSs can be extended and adapted easily via automated integration of new modules.

REFERENCES

[1] Allen, J., Christie, A., Fithen, W., Mchugh, J., Pickel, J., & Stoner, E. 2000. State Of The Practice Of Intrusion Detection

Technologies. Cmu/Sei-99-Tr-028,Cmu/Sei. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

[2] Anderson, D., Frivold, T., & Valdes, A. 1995. Next-generation intrusion detection expert system (NIDES): A summary. SRI-CSL-95-07 (May).

[3] Axelsson, S. (2000). "The base-rate fallacy and the difficulty of intrusion detection", ACM Trans. Information and System Security 3 (3), pp. (186-205).

[4] Bloedorn et al. (2003). Data Mining for Network Intrusion Detection: How to Get Started. The MITRE Corporation McLean, VA.

[5] Ganesh,K.A. (2003), " *Enhancing an Intrusion Detection System framework using selective feedback*", Masters thesis, Graduate school of the Ohio University.

[6] Ilgun, K., kemmerer, R. A., & Porras, P. A. 1995. State transition analysis: A rule-based intrusion detection approach. *IEEE Trans. Softw. Eng. 21*, 3 (Mar.), 181–199.

[7] Kumar, S. & Spafford, E. H. 1995. A software architecture to support misuse intrusion detection. In Proceedings of the 18th National Conference on Information Security. 194–204.

[8] Lee, W. & Stolfo, S.J et al. (2000). "A data mining and CIDF based approach for detecting novel and Distributed intrusions", In Proc. of Third International Workshop on Recent Advances in Intrusion Detection (RAID 2000), Toulouse, France

[9] Lee, W. & Stolfo, S.J. (1998). Data mining approaches for intrusion detection, In Proc. of the Seventh USENIX Security Symp., San Antonio, TX.

[10] Lee, W., Stolfo, S. J., & Mok, K. W. 1999. Mining in a data-flow environment: Experience in network intrusion detection. In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-99).

[11] Lunt, T. 1993. Detecting intruders in computer systems. In Proceedings of the 1993 Conference on Auditing and Computer Technology

[12] Mitchell, T. 1997. *Machine Learning*. McGraw-Hill, Inc., New York, NY.

[13] Quinlan,J.R.(1993), "C4.5: Programs for Machine Learning," Morgan Kaufmann,San Mateo,CA

[14] Tavallaee,M. , Bagheri, E. , Lu,W. & Ghorbani, A. 2009. "A Detailed Analysis of the KDD CUP 99 Data Set," Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

Authors

Ajayi Adebowale Received a B.Sc degree in Mathematics (computer science) from University of Agriculture Abeokuta 2007, he is a Cisco Certified Network Associate and holds an M.Sc degree in Computer science from Babcock University. He can be contacted at deboxyl {at} gmail.com

Sunday Idowu Phd is an Associate Prof. in the Department of Computer Science, Babcock University, Ilishan-Remo, Ogun State, Nigeria. He holds a Masters degree in Software Engineering, and Ph.D in computer science from Andrews University, MI, USA and University of Ibadan, Oyo State, Nigeria, respectively. His research areas are Software Engineering, Web Application Development and Security. He has published works in several journals of international repute. He can be contacted at saidowu07 {at}gmail.com.