International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-2, Issue-5)

Research Article | **May 2013**

Enhance IDS False Alarm Filtering Using KNN Classifier

Nitin Mohan Sharma,
Department of Computer Science,
Lovely Professional University
Punjab, India.

Tapan P. Gondaliya
Department of Computer Science,
Lovely Professional University
Punjab, India.

Abstract. Intrusion detection is one of the important aspects in computer security. Many commercial intrusion detection systems (IDSs) are available and are widely used by organizations. However, most of them suffer from the problem of high false alarm rate, which added heavy workload to security officers who are responsible for handling the alarms. In this paper, we propose a new method to reduce the number of false alarms. We model the normal alarm patterns of IDSs and detect anomaly from incoming alarm streams using k-nearest-neighbor classifier. Preliminary experiments show that our approach successfully reduces up to 93% of false alarms generated by famous IDS.

Keywords: Intrusion Detection Systems (IDS), commercial off-the-shelf (COTS)

1 Introduction

Information security has been one of the major issues concerned by computer professions in recent years. While human daily life is more and more dependent on computers, the number of cyber crimes, as well as the impact caused by the cyber crimes, is growing in incredible rates. According to the statistics collected by the CERT Coordination Center (CERT/CC) in the United States, the number of reported incidents related to computer security in a year raised from 1,334 to 137,529 over the last decade [1]. This is the result of the rapid growth of information technology applications and it shows the importance to protect our information assets from attacks and damages.

In response to the security threats, many technologies have been developed to guard valuable information assets against unauthorized disclosures, illegal modifications and unpredicted service interruptions. Besides common protection mechanisms (e.g. cryptography, firewalls and authentication) that prevent attacks from happening, intrusion detection systems (IDSs) are invented to uncover attacks and to alert administrators for countermeasures. Intrusion detection is one of the essential elements in computer security because prompt reactions to intrusive activities can greatly reduce harm to the systems and loss due to the attacks.

1.1 Intrusion Detection

Intrusion detection is the process of monitoring computer systems or networks and searching for signs of attacks [2]. An intrusion detection system consists of sensor(s) that listen to the activities within a computer system or network and generates alarms to administrators if it finds that suspicious activities (may or may not be an intrusion) occur. Over the years, computer scientists have developed numerous techniques to detect intrusions. Techniques of detecting intrusions fall into two categories, signature-based detection and anomaly-based detection. The former depends on some known attack patterns (signatures). Signature-based IDSs examine audit data (including, but not limited to, network traffic, system call logs, resources utilization, and users' activity logs) and compares it with the signatures. If there is a match, an alarm is generated to warn security officers for further investigation. Most commercial IDSs depend on signature based detection.

Anomaly-based detection attempts to find suspicious activities occur in the target systems. It is based on the assumption that attack scenarios are rare and in some ways attacking activities possess different characteristics from normal behaviors. To find out abnormal events, normal profiles of protected systems are modeled first and are learned by the IDSs. Incoming audit data are then classified to see if it conforms to the normal model. If not, there may be attacks and alarms are raised. Some research works have suggested combination uses of the signature-based and anomaly-based techniques [4]. The hybrid systems take advantages from both detection schemes to achieve faster speed and ability to uncover new attacks.

1.2 Problems of Intrusion Detection Systems

Although IDSs have been used for years and have demonstrated their values to organizations' security, most of them suffer from the problem of high false alarm rate and of having difficulties in fine-tuning. Practitioners often complaint that commercial off-the-shelf (COTS) IDSs trigger tons of alarms, but most alarms are actually false. The number of undesirable false alarms generated by commercial IDSs in a site can be as high as thousands per day! Identifying real alarms from huge volume of alarms is a frustrating task for security staff. Even worse, when security officers receive huge amount of false alarms everyday and treat them as a norm, they may oversee the importance of incoming alerts when real attacks occur [3]. Therefore, reducing false alarms is a critical issue in enhancing efficiency and usability of IDSs. IDSs can be fine-tuned to suppress false alarm generation. However, it is not that easy because improper configuration may degrade security. A signature-based IDS depends on a set of rules to separate intrusive behaviors from

streams of audit data. For example, a telnet connection to a UNIX machine with *root* privileges may be dangerous, so IDSs will trigger alarms when they see such kind of connection. It is obvious that the tighter the rule set, the stronger the security can be achieved. However, a tight rule set always induces more alarms, while many of them are actually not intrusive. Relaxing some rules can reduce the number of false alarms, but this action is risky, causing the IDSs unable to detect certain noteworthy incidents. The tuning problem is actually to search for a balance of reducing false alarm rate and maintaining system security.

1.3 An Overview of Our Approach

This paper reports our research that tries to reduce the number of false alarms without sacrificing security. The objectives are to reduce false alarm rate and to maintain the level of security achieved. Our approach is to let the false alarms being issued as they are and then detect any abnormal patterns from them using data mining techniques. We believe that when an attack is taking place, the alarms generated by the IDSs will have different patterns from that in an attack-free environment. Detecting these abnormal patterns can find out suspicious incidents from tones of false alarms. Those alarms which are classified as normal can be ignored. In this sense the security officers' headache, high false alarm rate and hard configuration of IDSs, can be released.

It is worth noting that our work is to reduce the number of false alarms, hence the usability of IDSs is improved. We do maintain the security level offered by intrusion detection algorithms, but we are not going to enhance it. More specifically, alerted attacks detected by the IDSs should not be filtered out by the false alarm reduction process; however, attacks that the IDSs missed will remain undetected. Enhancing detection ability of IDSs is beyond the scope of this paper. The remaining parts of this paper will review research works on alarms handling for IDSs; propose an approach to filter false alarms; and describe experiments to evaluate our idea with results.

2 IDS Alarm Handling Using Data Mining

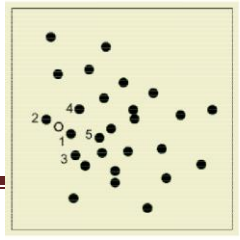
Mining IDS false alarms is not a new research area. Manganaris *et al.* analyzed alarm streams to find association rules [5]. He suggested an alarm handling framework that filters false alarms for IBM's network operations center, which provides real-time intrusion detection services to customers. His approach is to characterize IDS normal behavior in terms of frequent alarm sets with minimum occurrence in bursts of alarms. Then incoming alarms were classified to look for non-frequent alarm sets which are considered to be suspicious.

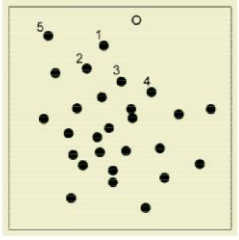
Julisch, also from IBM, proposed to find alarm clusters and generalized forms of false alarms to identify root causes [6]. He observed that 90% of false alarms are correspondent to a small number of root causes. Knowing the root causes, human expertise can adjust the IDSs regularly or remove the root causes to reduce the number of false alarms by 82% as shown in Julisch's experiments. He also mined IDS alarms for episode rules [7], which try to predict the set of alarms follows when a specific set of alarms occurs. He believed that the rules are useful because with the knowledge of such alarm patterns representing legitimate uses of the protected systems, highly similar alarms (which are supposed to be legitimate, too) can be filtered easily in the future. However, in his report, the episode rules offered only 1% of alarm reduction rate and 99% of alarms were left for manual processing. Data mining technologies have shown their capabilities to reduce more than a half of false alarms. Our approach described in this paper further reduces the false alarm rate using KNN classifier.

3 Suggested False Alarm Filtering Using KNN Classifier

In this section, we suggest a filtering method for IDS alarms that significantly reduce false alarm load. We believe that when a network is under attack, the IDS will issue alarms in a way that is more or less different from usual situation. For example, the IDS may trigger an alarm of the type that is absent in normal situation or issue more alarms of certain types, depending on the attacks it is experiencing. Our method is to determine whether the incoming alarm sequences are deviated from normal situations. If that is the case, it may be a sign of attacks and further investigation is needed. On the other hand, if the incoming alarms possess very similar patterns as in an attack-free situation, the risk of being attacked is low. Below we will describe how to model normal alarm patterns and how to detect deviations from the normal patterns.

Given a large set of alarms generated by an IDS under an attack-free environment with N distinct alarm types, we model the normal alarm patterns with an N-dimensional space. A data point P (with N attributes $<A_1, A_2, A_3 \ldots A_N>$) in the space represents the counts of alarms of different types within a time window of size W. We define these points as "*normal*" because they are created from alarms for normal audit data ex-periencing no attacks. These alarms are said to be "safe" and are treated as false alarms. Now we have a set of data points telling us how the IDS behaves when there is no intrusion attempt.





(a) A normal point **(b)** An abnormal point

Fig. 1. Example of normal and abnormal points in the false alarm model. Numbered points are the 5 nearest normal (black) points from the new (white) point

To detect abnormal patterns from newly arrived alarms, new data points are created for the new alarms in the same way. The new points' distances from the normal points indicate their deviances from normal patterns. If a new point is close to the normal points, it is considered as normal too. Then we say in the time period that the point represents, there is nothing special in alarm distribution and alarms generated in that period are false alarms. Figure 1a shows a simple example of the model and a new normal point (Note that for the ease of illustration it is a 2-D example. In real situations the dimension is much larger than two). On the other hand, we consider a new point as abnormal if it satisfies any one of the following cases:

1. it lies far apart from the normal points (see Figure 1b)
2. it consists of new alarm type(s) that is absent in normal points

An abnormal point tells administrators that in the corresponding time period, the alarm distribution is strange and some intrusive activities may be happening.

We use k-nearest-neighbor (KNN) classifier to classify whether a data point is normal or abnormal. KNN classifier has been first used in intrusion detection area for anomaly detection to learn program behaviors and uncover intrusions from audit data [8]. Here we try to extend its use in false alarm reduction. The KNN classifier measures the distance between two data points *P* and *Q* by Euclidean distance (Formula 1). The distance actually represents their similarity. The shorter the distance between them, the more similar they are.

(1) $$distance(P,Q)=\sqrt{\sum_{i=0}^{N}(p_i-q_i)^2}$$

where p_i and q_i are the values of the i^{th} attribute of points *P* and *Q* respectively. The final similarity score of a data point being classified is the average of its Euclidean distances from the closest *k* normal points. If the similarity score is higher than a threshold *T*, the point is said to be abnormal and the alarms corresponding to it are noteworthy (case 1 as mentioned above). On the other hand, alarms corresponding to low-score data points are false and are filtered.

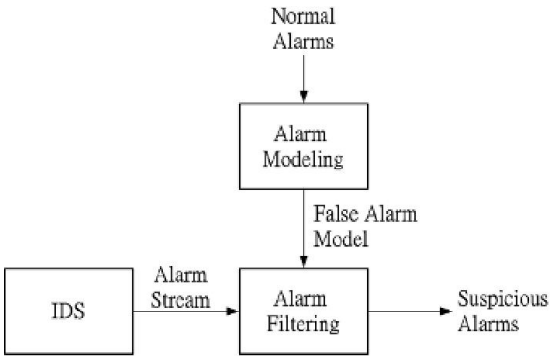


Fig. 2. Relationship between IDS and the proposed alarm reduction processes

Our suggested modeling and filtering processes are independent from the intrusion detection process (Figure 2). Therefore we believe that our model can be applied to most commercial IDSs in use nowadays without changing the existing detection con-figuration. The alarm modeling process makes use of normal alarms, i.e. alarms raised by IDSs under no attacks, to construct false alarm models as described earlier in this section. The normal models are used by the alarm filtering procedure in which con-tinuously incoming alarms are filtered. Only the alarms left out needed to be investi-gated. The entire reduction procedure can be seen as a plug-in to IDSs.

4 Experiments

To evaluate the applicability of our approach, we conducted preliminary experiments with DARPA 1999 dataset. The DARPA Intrusion Detection Evaluation [9] program evaluated intrusion detection technologies with sufficiently large sample dataset which contains network traffic embedded with marked attacks. What we used in our experi-ments are TCPDUMP data collected from an Air Force Local Area Network in 1999. It was actually the captured traffic over the network. There are three weeks (5 days a week) of training data, with crafted attacks in week 2 only and no attack in week 1 and week 3.

4.1 False Alarms from Snort

We examined the network traffic with an open-source signature-based IDS, namely Snort [10]. Snort is a well-known network intrusion detection freeware available on the Internet. It looks for signs of attacks from the network traffic by comparing its own set of attack signatures and incoming packets, and generates an alarm when it finds a match. The Snort recorded each incident to database with the exact timestamp, alarm type, packet header and other relevant information.

Table 1. Top 10 frequent alarm types

Alarm Type	Occurrence
SNMP public access udp	54498
ICMP PING NMAP	7000
FTP CWD	5426
TELNET access	1799
SCAN FIN	1027
ATTACK-RESPONSES 403 Forbidden	726
TELNET login incorrect	554
WEB-MISC /doc/ access	551
WEB-CGI redirect access	542
WEB-MISC apache DOS attempt	480

We observed that the Snort, with default configuration, had issued more than 75000 alarms when processing the 3 week data, with more than 5000 per day on average. There were 76 different alarm types. For the attacks in week 2, Snort was able to detect 18 out of 43 attacks. From intrusion detection's points of view, it is not satisfactory. It may be improved with correct configurations, but improving detection ability is beyond the scope of this study. Among the alarms, we observed that 4 alarm types contributed for more than 68000 (91%) alarms. This observation matches with Julisch's one [6] as mentioned in Section 2. The 10 most frequent alarms types are listed in Table 1.

4.2 False Alarm Model

Since the DARPA 1999 dataset contains no intrusive traffic in week 1 and week 3, alarms triggered from dataset of these two weeks are false alarms. There are totally 60549 false alarms in these two weeks. The modeling of false alarm patterns is de-scribed in the previous section. A data point represents the alarm distribution of dif-ferent alarm types in a time period. Since there are 76 different types of alarms, there are 76 attributes for each data point. Each attribute value equals to the count of the corresponding alarm type within the time period. The length of the time window was determined arbitrarily. Obviously, a shorter time window would tell administrators more precisely the time when a noteworthy data point got identified. Here we set the time window size to be 2 minutes. The order of magnitude of different alarm types may be different in nature and it will affect the accuracy of the classifier, so the attribute values of the points are normalized to eliminate this effect. Another parameter to determine is the sampling rate, i.e. how frequent we construct data points with the 2-minute window size. The time interval of sampling between two successive data points should be shorter than the size of time window that a single data point represents. Otherwise, there will be gaps between data points and some alarms will be missed in the model. So we set the time interval to be 1 minute. The data points of week 1 and week 3 alarms represent how the IDS behaves under attack-free situation and we call them normal points.

References

[1] Paxson, V.:Bro: A System for detecting network intruders in real time. Computer networks 31(23-24) 2435-2463 (1999)
[2] Scarfone, K., Mell, P.: Guide to Intrusion Detection and Prevention System, pp.800-894. NIST special publication (2007)
[3] Davenport, M.A.,Baraniuk,R.G.,Scott,c.D.: controlling false alarms with support vector machines: in: international conference on acoustics, speech and signal
[4] Alharby, A.,Imai,H.: IDS false alarm reduction using continuous and discontinuous patterns. In: Ioannidis,J.,Keromytis,A.D., Yung,M. (eds.) ACNS 2005.pp. 192-205. Springer,Heidelberg (2005)
[5] lee, W.,Stolfo,S.J..:A Framework for constructing features and models for intrusion detection system. ACM Transactions on information and security 227-261 (2000)
[6] WEKA-waikato environment for knowledge analysis (open source) http://www.cs.waikato.ac.nz/ml/weka

[7] wireshark packet capture http://wireshark.com
[8] colasoft packet builder http://www.colasoft.com/packet_builder
[9] lippman,R.,Haines,J.W., Fried,D.J.,Korba,J.,Das,K.:The 1999 DARPA offline intrusion evaluation. Computer networks: the international journal of computer and telecommunication networking , 579-595 (October 2000)
[10] Pietraszek , T.: Using Adaptive Alert Classification to reduce false positives in intrusion detection. In : Jonsson, E.,pp.102-124. Springer , Heidelberg
[11] Davenport , M.A., Baraniuk, R.G., Scott,C.D. Controlling false alarms with support vector machine. In: International Conference on speech and signal (may 2006)
[12] Nitin Mohan Sharma, Kunwar Pal, "Implementation of decision tree algorithm after clustering Through WEKA" International Journal of Computer Engineering and Technology (IJCET), Volume 4, Issue 1, Jan. – Feb. 2013 pp. 358-363 Published by IAEME
[13] Tapan P. Gondaliya, Maninder Singh, Lovely Professional University "Intrusion detection System for Attack Prevention in Mobile Ad-hoc Network", Volume 3, Issue 4, April 2013.