An Efficient Network Anomaly Detection Method Using Management Information Base

Lei Zhao

School of Electronic Information, Shanghai Dianji University, Shanghai 201306, China.

Abstract

In this paper we have developed an entropy-based method that is real-time capable and can reports entropy contents of data provided by Management Information Base (MIB). Change of the entropy value indicates a massive network event and an network anomaly may happen. Experiments show that the method is effective and efficient for a wide variety of computing environments.

Keywords

Network Anomaly Detection, Management Information Base, Entropy.

1. Introduction

All As network play increasingly vital roles in modern society, they also become the targets of enemies and criminals. Therefore we need to do research on techniques and measures to protect our network system.

Anomaly detection is an important component of layered network security mechanism. For example, an anomaly detection system collects data from network and system e.g., tcpdump and BSM [1] , and analyze them to determine weather is an attack taking place. Anomaly detection systems, such as IDES , establish normal usage patterns called profile as statistical measures on system features[2]. Anomaly detection can be effective against new attacks. IDIOT[3] and STAT[4], use the "signatures" of known attacks to identify a matched activity as an attack instance. Anomaly detection can be effective against new attacks. In practice, reports of anomaly detection are often sent to security department for investigation and to take appreciate actions.

In this paper we have developed an entropy-based method for anomaly detection using Management Information Base. The key advantage of the approach is that it can automatically generate concise and accurate detection models from large amount of audit data. Its computation is also very fast and real-time without complex statistical analysis. The method is general and mechanical, and therefore can be used to build anomaly detection systems for a wide range of computing environments.

2. Network Anomaly Detection Method Using MIB

2.1 Network Anomaly Detection Model

Table 1 Indicators of interface class in MIB

Object indicator	Data type	Description
ifInOctets	Counter32	Number of bits received by interface
ifInUcastPkts	Counter32	Number of unicast packets received by interface
ifInNUcastPkts	Counter32	Non-unicast packets received by interface
ifInDiscards	Counter32	Number of packets discarded by the interface
ifInErrors	Counter32	Number of error packets received by the interface
ifInUnknownProtos	Counter32	Nnumber of unkown proctocol packets received by interface
ifOutOctcts	Counter32	Number of bits sended by the interface
ifOutNUcastPkts	Counter32	Number of non-unicast packets sended by the interface
ifOutDiscards	Counter32	Number of packets discarded by the interface
ifOutQLcn	Unsigned32	Length of packet list

Our goal is to perform network anomaly detection in the challenging real-world conditions within a dynamic and complicated environment. The method uses the Management Information Base (MIB) counter database provided by a large international network Company such as Cisco and Huawei. The MIB database contains 11 categories of classes, including some basic system information such as the system class or network-related information such as the IP class and TCP class, etc. Most of the classes have non-numeric values or strongly associated with special network applications therefor not suitable for general analysis. We investigate the interfaces class in MIB, which is often be used as the identification of the network interface and is not related with specific network protocol. Table 1 shows the 12 indicators contained in the interface class.

Fig. 1 shows the model of our method. It can be seen from Fig. 1 that the detection process includes three units: the data pre-processing unit, the data processing unit and anomaly detection unit.

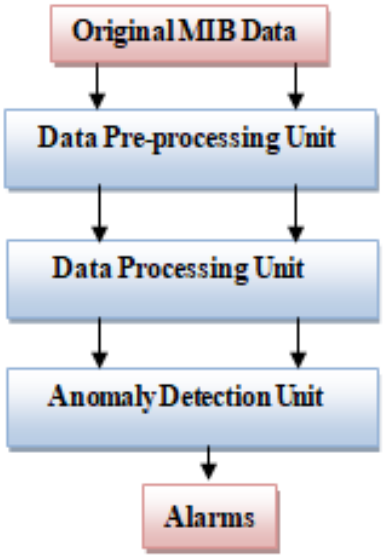

Fig. 1 Model of the Method

2.2 The Method

The data pre-processing and processing stage are to select appropriate indicators for analysis from MIB. For example, we can combine the change of indicator ifInOctets with time $x(t)$ and the change of indicator ifInDiscards with time $y(t)$ together to a new variable

$$z(t) = x(t) + \alpha y(t) \tag{1}$$

where $\alpha = 3000$ is the weight for indicator ifInDiscard.

We use the concept of entropy to fining uncertainty in network traffic**.** Entropy is an important component in information theory for measuring the uncertainty or impurity of a collection of data items [5]. For a dataset X where each data item belongs to a class X , the entropy of X relative to the class is defined as

$$H(X) = -\sum p(x)\log(p(x)) \tag{2}$$

where $p(x)$ is the probability of x in X

Following is a piecewise window dividing method for $z(t)$ .We divide the time axis of $z(t)$ into equally 10 minutes time per window and each has 5 data points. Then we compute the entropy of each window supposing their data conform to some kind of probability distribution such as Gaussian

or Parzen Window. An anomaly is likely to happen if the entropy or entropy ratio computed is larger than threshold u. Here the entropy ratio is the entropy of current window divided by the average entropy of previous windows. Fig. 2 gives an example of the entropy computation for 9 data windows.

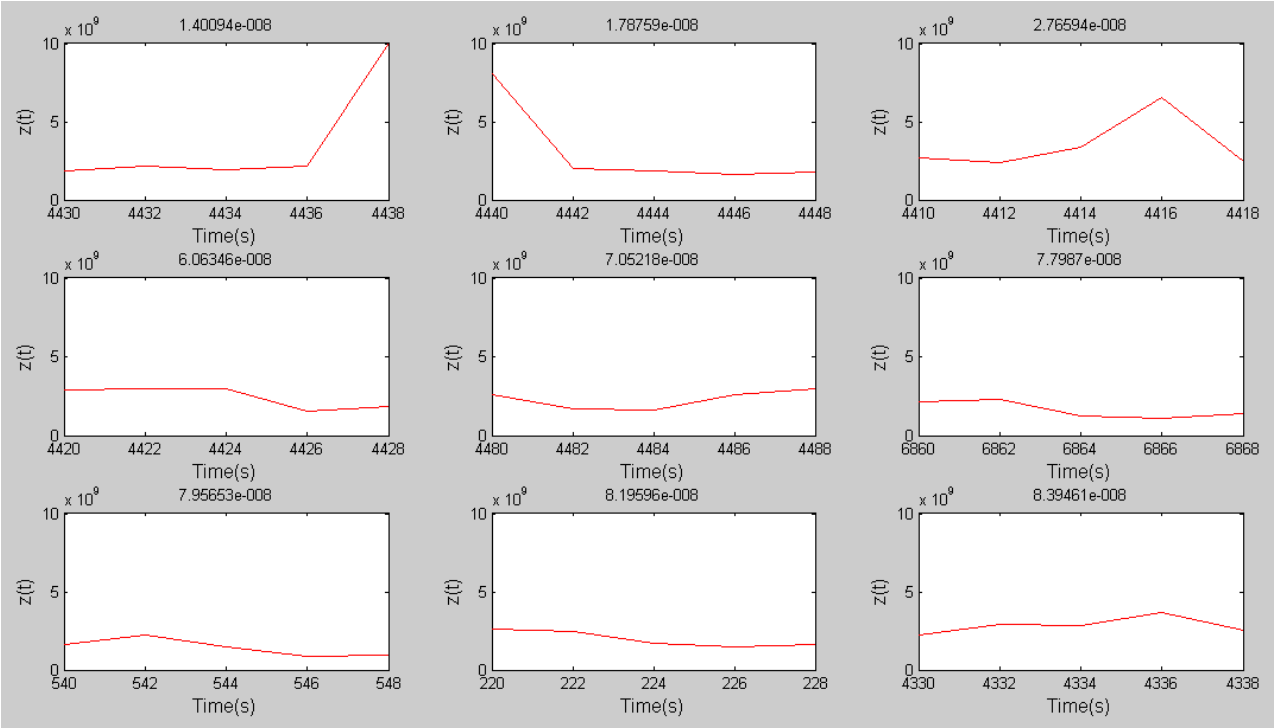

Fig. 2 Examples of entropy computation for 9 data windows

Supposing the data of MIB data satisfy the Parzen Window distribution which is a non-parametric way of estimating the probability density function for random variables. If $x_1, x_2, ..., x_n$ are independent and identically-distributed random variables, then the kernel density approximation of its probability density function for the Parzen Window distribution is

$$\widehat{f}_h(x) = \frac{1}{Nh}\sum_{i=1}^{N} K(\frac{x - x_i}{h}) \tag{3}$$

where K is some kernel and h is the bandwidth (smoothing parameter). Often K is a standard Gaussian function with mean zero and variance 1. We then use a self-implemented function to implement the Parzen Window. For example, we use previous 5 windows including 25 data points to estimate current window's probability by setting N=5*5=25 and h=1 in Eq. 3.

After getting each data point's probability in the current window, we use Eq. 2 to compute its entropy. In order to detect whether there is an anomaly occurred, we developed a new concept called entropy ratio for each window. First we define a variable d as the distances of previous windows to the current window. If $d = 1$, the entropy ratio is the ratio between previous one window's entropy and current window's entropy. If $d > 1$, the entropy is the ratio between previous d window's mean entropy and current window's entropy. We then can conclude weather there are some anomaly occurred in the window by compare the entropy or entropy ratio with a threshold u.

3. Experiment Results

Fig. 3 (a) is the processed curve for the combined variable $z(t)$ provided by the interface of router gw2. Fig.3 (b) is the entropy computed for each data window. Fig. 3(c) is the entropy ratio computed

for each data window. We can clearly see that an anomaly event is detected in the range of [4400s,4600s].

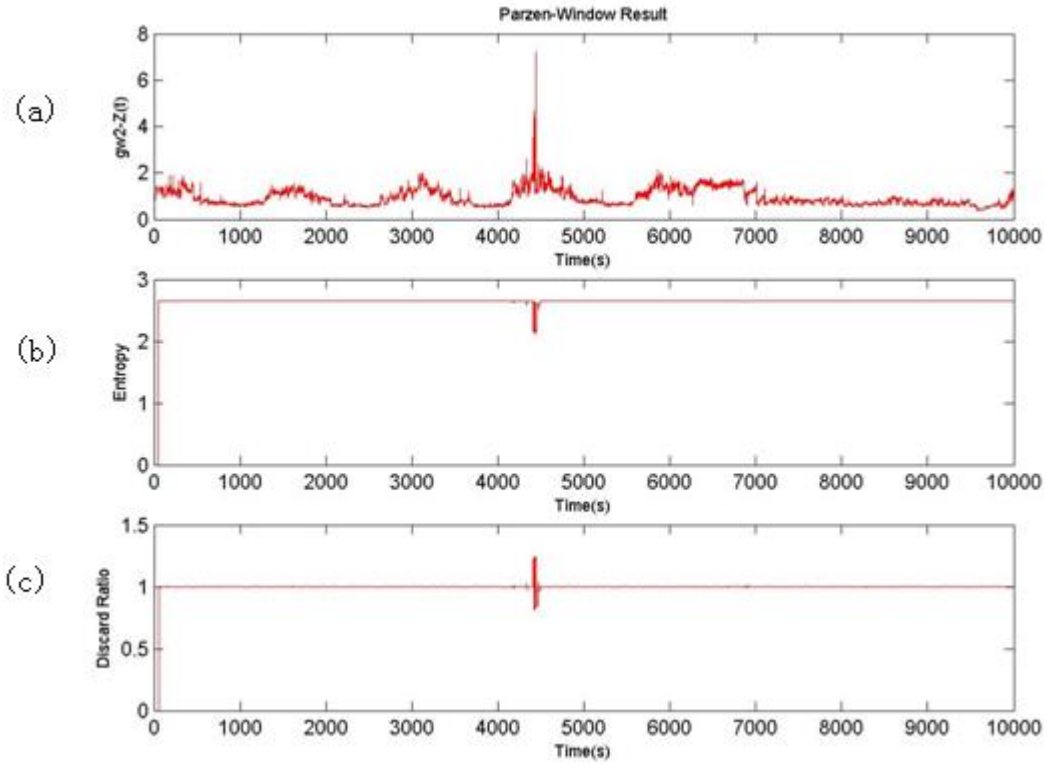

Fig. 3 Anomaly detection with a self-implemented function

4. Conclusion

In conclusion, we have developed an entropy-based method for network anomaly detection using Management Information Base. Experiments show that the method works well on the MIB data and can effectively report the anomaly event supposing the data satisfies the Parzen window distribution. More work should be done on this system using other distribution probabilities in the future.

Acknowledgements

This work was financially supported by the Major Program of 863 Plan (2012AA01A403), Important Subject Construction Program for Computer Application Technology (13XKJ01) and Doctoral Start Fund of Shanghai Dianji University (13QD03).

References

[1] SunSoft: SunSHIELD Basic security Module Guide. (Sun-soft, Mountain View, CA, 1995).

[2] S. Kumar, E. H. Spafford, in: Proceedings of the 18th National Information Security Conference (1995), p. 194-204.

[3] K. Llgun, R. A. Kemmerer, and P. A. Porras, in: IEEE Transactions on Software Engineering, Vol. 2 (2018) p.40-42.

[4] Information on http://securityresponse.symantec.com.

[5] Information on http://www. wikipedia.org.