International Research Journal of Computer Science (IRJCS)
Volume 1 Issue 1 (September 2014)
ISSN: Applied
www.irjcs.com

Improve IDS Alert Result By Using Decision Support Techniques

Nanaso S. Bansode[1], Anil B. Pawar[2], Thaksen J. Parvat[3]

Department of Computer Engineering, University of Pune
Sinhgad Institute of Technolgy,Lonavala,Pune,Maharashtra,India

Abstract—Intrusion Detection System (IDS) is used to monitor all activities which are running on particular machine or network. Also it will give you alert regarding to any attack. But these alerts are very huge. It is very difficult to analyze whether given alert is true or false. That's why we create new concept which will decide given alert is true or false. We can classify given alert by using three phases alert preprocessing, model construction and rule refining phase which will give you true alert.

Keywords—IDS; true alert; false alert

I. INTRODUCTION

When number of computer to each other connected by some medium which may be wired or wireless and they can share information between them then we call these computers are in network. Some attackers attack to particular computer or network which is called as unauthorized user. These authorized users are not valid user. They have no permission to access computers. Only authorized user can use the utilities which are on particular computer to access.

Attacker is nothing but unauthorized user which will attack your computer system and take confidential data from your computer. That's why IDS is generated [1-4]. IDS will give you alert regarding to any attack also monitor all the activities which are on particular computer or network. Our idea is to create a decision support system which will help to expert construct an alert classification model for online intrusion detection of IDS alert [5-8].

Our idea is to collect all alert transactions in an environment, which is logical intranet with number of hosts in laboratories to represent real internet behavior. As per our proposal we consider all incoming alerts are false alert in logical intranet and number of time occurred alerts are treated as normal behavior or may be false alert. We can say that use this technique to create normal behavior pattern remove false alerts.

The proposed IDS alert result improvement model consist of three phases: Alert preprocessing phase, Model construction phase and Rule refining phase to create three rule classes (normal behavior rule class, Intrusion rule class and Suspicious rule class) to remove false alert and identify each unknown alert pattern and each rule class contains a set of classification rules. A least recently used (LRU)

Rule replacement policy is used to replace rules which are less used in each rule class.

II. IDS ALERT RESULT IMPROVEMENT ARCHITECTURE

There are number of methods to analyze IDS behavior but it is very difficult get desired intrusion pattern. Most of these methods are independent. Each method has its own limitation so integrating advantages of different methods which will give us better result than individual one. Hence we propose IDS alert result improvement model help experts easily to create alert classification model using large number of alerts.

The important purpose of IDS alert is that collection and identification are finding more meaningful alert information and give the information of relation between real alert to verify system attacks. Some issues are derived from these purposes.

(a) How to choose particular analysis targets and data format.

(b) How to filter false alert correctly.

(c) How to discover attack methods and display particular data types for administrator to make policies.

© 2014, IRJCS- All Rights Reserved

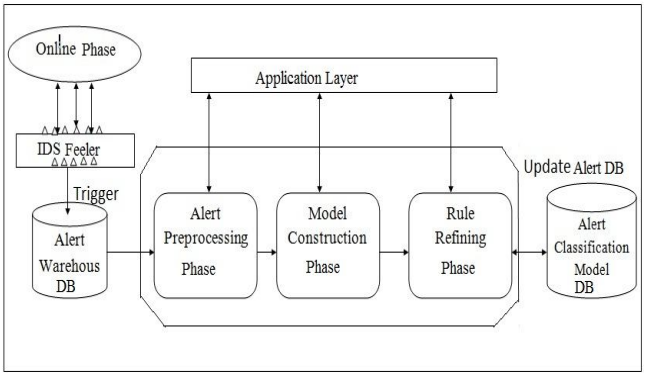

Fig1. IDS alert result improvement architecture

In the online stage, IDS alert result improvement model for creating an alert result classification model consists of three phase alert preprocessing, is shown in figure 1. In alert preprocessing phase we can collect all alert which are triggered by IDS sensors. We can store these alert into alert warehouse database, for alert preprocessing, we take one by one alert and convert it into alert sequence. In model construction phase, filtering and identification methods are proposed for creation of classification rule classes to remove false alert and identify each existing alert method. The normal alert behavior will take firstly in online stage by sequential mining algorithm and used to reduce affect of noise, because normal behavior method occur periodically and frequently suspicious behavior method will occur in attacking environment.

A. Alert Preprocessing

In IDS alert transaction, alert warehouse are large in online stage. It is required to convert raw alert into specific format for next phase analysis. Some of the characteristics of these alert transactions are same [9].

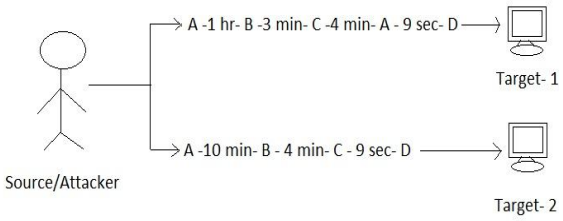

Fig2. Attack Tool

Alert sequence generated from alert transaction can be represented as particular characteristics of specific attack tool that's why we select as our target data for analysis. We collect all IDS alert transaction by using IDS sensors as our online data source [10] and set appropriate time period for alert transaction Gap_Time for batch processing. The Gap_Time value may be a month, a week, a day, an hour, a minute or a sec. But we assume that the value of Gap_Time as one day will be better in our experience.

Most attack will not continue for very long time in its own attack lifecycle, that's why we select short period time. We assume every alert transaction complete in short period may be within 1 minute.

First, for a single alert transaction, it is difficult to identify whether given alert is true alert or false alert because some alert are used to triggered, second if the alert sequence is too long because long sequential method are not easy. Some policies of alert sequence for model construction phase, these policies are used to divide alert sequence into number of partitions in each Gap_Time.

There are following policies which are used to divide one alert sequence into number of sub alert sequence as follows.

CASE- 1: Left to Right Policy.

Step 1: Scan every alert sequence from left to right Sequence.
Step 2: Divide sequence into scanned part and unscanned

© 2014, IRJCS- All Rights Reserved

Part; if next sequence to be scanned is equal to Some alert in the scanned part. Then do partition from first element to present element and set the un scanned as new sequence.

Step 3: Still any element is remaining in alert sequence then goto step1.
CASE- 2: Right To Left Policy
Step 1: Scan every alert sequence from right to left Sequence.
Step 2: Divide sequence into scanned part and un scanned Part; if next sequence to be scanned is equal to some alert in the scanned part. Then do partition From first element to present element and set the unscanned as new sequence.
Step 3: Still any element is remaining in alert sequence then goto step1.

CASE- 3: Equal Length Policy
Step 1: Administrator assigns value of subsequence
Step 2: Divide alert sequence into several subsequences with fixed length.
Now we use first policy which is left to right policy for partition purpose, suppose there is an alert sequence of sensor D1 in short_term as follows.

Sensor ID	Alert Sequence
D1	ABCXADCD

Firstly let AS[7] be alert sequence is ABCXADCD; because AS[4]='A' and AS[0]='A', given sequence is divided into two parts scanned part and unscanned part. Scanned part is ABCX and unscanned part is ADCD. Compute unscanned part again then it is divided into two part ADC as scanned part and D as unscanned part. Then whole alert sequence is to be scanned from left to right we get three sub-sequences as ABCX, ADC and D.

Sensor ID	Alert Sub-sequence		
D1	ABCX	ADC	D

B. Model Construction

There are some techniques which gives idea about how to filter false alert efficiently. We use particular method to construct filter model and use this model to identify false alert to get clear data.

Here we overcome alert pattern for two purposes: false alert filtering and true alert discovering. If alert is false then it is compulsory to filter that alert before discovering because noisy alert will affect the result of IDS alert. Our idea is to discover false alert first and then continue for true alert.

In online stage, we design some rule classes to create particular behavior rule classes. We construct three types of rule classes which are normal behavior rule class, intrusion detection rule class and suspicious behavior rule class. Each of these classes is constructed using individual method with their input.

The procedure for constructing rule classes as follows

1) Construct normal behavior rule class
2) Construct intrusion behavior rule class
3) Construct suspicious behavior rule class

These behavior rule classes are used to monitor IDS alert behavior in online stage. IDS sensors are will trigger all alert, it is very difficult to analyse for expert whether it is true alert ar false. Before storing it into alert warehouse each sequence is mapped by rule classes first. The normal behavior rule class then it is forwareded to next rule class which is used to detect true alert. Lastly remaining alert is classified into known supicious behavior as unknown suspicious alert.

© 2014, IRJCS- All Rights Reserved

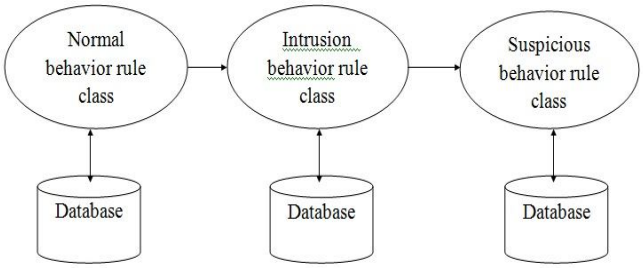

Fig3. Mapping of alert in online stage

C. Rule Refining

If any alert sequence match with normal behavior rule class then filter it immediately, if it is not match with normal behavior rule class then pass to next rule class which is intrusion rule class if it is match that then notice to expert, still no match then it is stored in alert warehouse as new alert.

These rule classes are used for administrator to identify alert in online stage. If any new rule class is generated then it is integrated to rule classes for efficient result. We require maximum number of classification rule classes, if any new rule class is generated then the performance of online monitoring is decreased due to huge number of rule classes. Initially we set maximum length of rule class is 200.

If storage capacities of rule classes are full, it is necessary to find replacement strategy. We use LRU policy is used as page replacement algorithm and LRU is used for replacement policy in rule classes. LRU will choose that rule which has not been mapped for longest period of time.

III. DESIGN OF EXPERIMMENTAL MODEL

Her we design experimental model which contains one server which acts as IDS alert analysis server, including alert warehouse. Number of Hosts plays the role of IDS sensors to trigger alert.

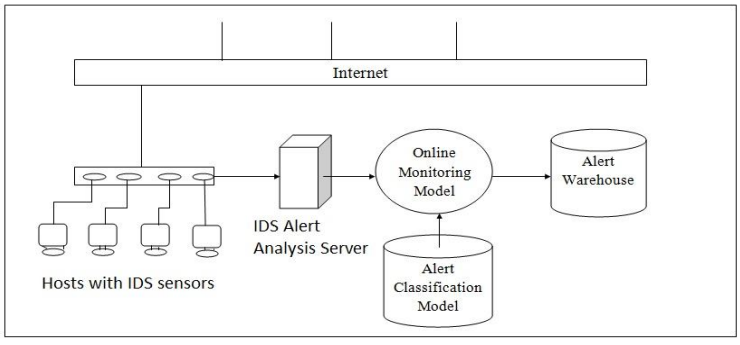

Fig4. Experimental Model

IV. CONCLUSION

In this paper, we propose IDS alert result improvement model which consist of three phases; Alert preprocessing, Model construction and Rule refining. In alert preprocessing all alerts are collected into alert warehouse. Take one by one alert for processing. Convert alert transaction into alert sequence and map with model construction phase which containing three classes' normal behavior rule class, intrusion behavior rule class and suspicious rule class. By Using these three rule classes administrator will analyze whether given alert is true or false. This model is useful for expert in online stage to discover intrusion quickly and precisely.

© 2014, IRJCS- All Rights Reserved

REFERENCES

[1] Yan Zhang, Shuguang Huang, Yongyi Wang, "IDS alert classification model construction Using Decision Support Techniques," in International conference on computer science and electronics engineerin, DOI 10.1109/ICCSEE.2012.242.
[2] Y. Hsin, S. S. Tseng, S. C. Lin, "A study of alert-based collaborative defense," in Proceedings of The 8th International Symposium on Parallel Architectures, Algorithms & Networks (ISPAN 2005), Las Vegas, Nevada, U.S.A., pp. 148-153, 2005.
[3] S. R. Madden, M. A. Shah, and J. M., "Hellerstein continuously adaptive continuous queries over streams," in Proceedings of ACM SIGMOD 2002, pp. 49-60, 2002.
[4] B. Morin and H. Debar, "Correlation of intrusion symptoms: an application of chronicles," in Proceedings of the 6th symposium on Recent Advances in Intrusion Detection (RAID 2003), pp. 92-112, 2003.
[5] P. Ning, Y. Cui, and D. S. Reeves, "Constructing attack scenarios through correlation of intrusion alerts," in Proceedings of 9th ACM Conference on Computer and Communications Security, pp.245-154, 2002.
[6] P. Ning, D. Xu, C. G. Healey, and R. A. St. Amant, "Building attack scenarios through integration of complementary alert correlation methods," in Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS'04), 2004.
[7] Alharby and H. Imai, "IDS false alarm reduction using continuous and discontinuous patterns," in Proceedings of ACNS 2005, pp.192-205, 2005.
[8] P. A. Porras, M. W. Fong, and A. Valdes, "A mission-impact-based approach to INFOSEC alarm correlation," in Proceedings Recent Advances in Intrusion Detection, pp.95-114, 2002.
[9] M. S. Shin, E. H. Kim, and K. H. Ryu, "False alarm classification model for network-based intrusion detection system," in Proceedings of IDEAL 2004, pp.259-265, 2004.
[10] Symantec Corp., "Symantec Internet Security Threat Report: Trends for July 05-Decamber 05," http://www.symantec.com/index.htm, 2005.
[11] K. Julisch and M. Dacier, "Mining intrusion detection alarms for actionable knowledge, in Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining, pp. 366-375

© 2014, IRJCS- All Rights Reserved