Packet Header Anomaly Detection Using Bayesian Belief Network

Mongkhon Thakong[1] and **Satra Wongthanavasu**[2], Non-members

ABSTRACT

This research paper presents a packet header anomaly detection approach by using Bayesian belief network which is a probabilistic machine learning model. A DARPA dataset was tested for the performance evaluation in the packet header anomaly detection or DoS intrusion-type. In this respect, the proposed method using Bayesian network gives an outstanding result determining a very high detection rate of reliability at 99.04 % and precision at 97.33 % on average.

Keywords: Keywords: Bayesian network, Bayesian learning, packet anomaly detection.

1. INTRODUCTION

There are several threats on the network on nowadays which may harm the data such as viruses, worms, etc. Therefore, it is necessary to provide the intrusion detection in order to increase the efficiency of network security. In this regard, there have been many different models from previous studies related to intrusion detection, statistics and machine learning model, for example. For each of them, statistical data of the detection were collected and then were analyzed to determine the relation of variables that caused the intrusion.

In order for determining the rule relationship of a intrusion detection system, the studies of the structure of data and properties of intrusion are needed. However, almost of the data on the network were based on 'Header' that the researcher was interested in by observing their relation and the effect that causes the abnormal model on the network.

In this research, we have constructed a model of Bayesian belief network, simply called Bayesian network, to determine the structure of data relationship, more specifically a field factor of header, to detect and analyze if the data is abnormal and probably intrudes into the computer network.

Manuscript received on December 12, 2006 ; revised on March 21, 2007.

[1]The authors are with the 64 UdonThani Rajabhat University, Udonthani, Thailand, 41000; E-mail : mong_khon@hotmail.com

[2]The author is with the Department of Computer Science, Khon Kaen University, Thailand, 40002; E-mail : wongsar@kku.ac.th

2. BAYESIAN NETWORK

It is a method of learning network construction, using a probabilistic theory based on Bayes Theorem to assist the hypothesis learning of condition independent between the variables or the properties, with prior knowledge and teaching examples for learning process effectively [1].

The study network can set the prior knowledge to the network structure on the Bayesian network and show the result in a directed acyclic graph-DAG, which can indicates the variables relied or relied not on others, and Conditional Probability Table - CPT). An example of Bayesian network and CPT is shown in fig. 1.

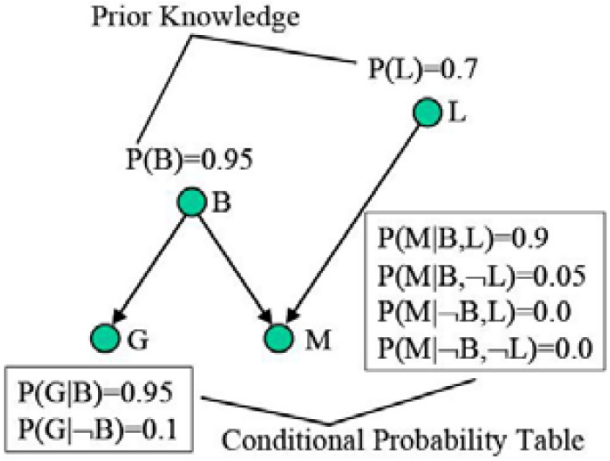

Fig.1: *An example of Bayesian network [1]*

2.1 Bayes Theorem

Let A and B be any events. Probability of A when knows B (probability of event A happened under condition of event B) equals P(A | B).

$$\mathrm{P(A \mid B)} = \frac{P(B \mid A)P(A)}{P(B)} \tag{1}$$

From Formula 1, P refers to prior probability and P(A|B) refers to posterior probability. The prior probability was the value from primary data. The posterior probability was the probability value adjusted by changing values. Due to Bayes Theorem, each probability was able to be analyzed, when the set of teaching examples were true, that would helped making the best choice of hypothesis that was Maximum A Posterior hypothesis (MAP).

$$h_{MAP} = \arg\max_{h \in H} P(h|D) = \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \quad (2)$$

$$= \arg\max_{h \in H} P(D|h)P(h) \quad (3)$$

When H was the vector space of all hypotheses, arg max f(x) was function which returned the f(x) value from Bayes Theorem as in Formula 2. Due to every hH had the same P(D) value, this can leave P(D) and got the Formula 3, that was hMAP which volumed the P(D|h)P(h) value up to the best choice of hypothesis.

There is another method by Bayes, or hML (Maximum Likelihood hypothesis), as shown in Formula 4, which represented the posterior probability that the probability value was adjusted without the prior probabilty.

$$h_{ML} = \arg\max_{h \in H} P(h|D) \quad (4)$$

2.2 Bayesian Learning

It is the network structure finding and/or CPT tables which are likely correlated to the teaching examples. Problems of Bayes learning network are 2 cases:

2.2.1 Structure Learning

It is the process of learning value adjustment of Bayesian network structure. The 2 ways of making the structures are constrain-based and search-and-score. [2, 10]

1. Constrain-based is a Bayesian network constructing made by connecting all nodes (full connected) and then erasing unconditional independent relations.

2. Search-and-score is a Bayesian network constructing that finds all directed acyclic graphs (DAG), then analyses each graph value and selects the best graph.

2.2.2 Parameter Learning

Structure from structure learning part was adjusted by each variable, called node, in the Bayesian network. Results of each node are in probability table, called Conditional Probability Table - CPT.

3. EXPERIMENTS

3.1 Data preprocessing

It was the process of transforming raw data to the data used when analyzing. In this research, the raw data was taken from 1999 Dataset of DARPA Evaluation offline [5] which was the data recorded by the simulated network in DARPA. The structure of data used in the analyzing was the packet data in tcpdump pattern as shown in fig. 2.

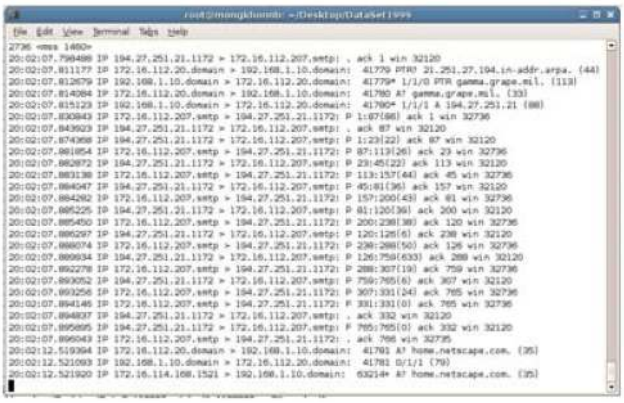

Fig.2: *Example of data in tcpdump model*

Due to the tcpdump model of data contained high capacity of memory and was not able to applied to the experiment, the researcher had changed the form of data and then filtered them for the specific data used in this research, or called field of packet header, to determine the structure of Bayesian network from the field of packet header variable of PHAD [6] which consisted of 33 fields and was divided into 5 types as followed:

- Ethernet Header consisted of Ethernet Size, Ethernet Destination High, Ethernet Destination Low, Ethernet Source High, Ethernet Source Low and Ethernet Protocol

- IP Header consisted of IP Header Length, IP TOS, IP Length, IP Fragment ID, IP Fragment Pointer, IP TTL, IP Protocol, IP Checksum, IP Source and IP Destination

- TCP Header consisted of TCP Source Port, TCP Destination Port, TCP Sequence, TCP Acknowledgement, TCP Header Length, TCP Flag UAPRSF, TCP Window Size, TCP Checksum, TCP URG Pointer and TCP Option

- UDP Header consisted of UDP Source Port, UDP Destination Port, UDP Length and UDP Checksum

- ICMP Header consisted of ICMP Type, ICMP Code and ICMP Checksum

When studying the Protocol TCP/IP and the property of data frame encapsulated in the DARPA computer network that was important to each header, the researcher had selected those fields of header, or variables, and studied the relation of the headers. After that, the selected variables were brought to construct Bayesian network which were considered by the relations of variables, and then were brought to the

experiment. They are IP TTL, IP Lenght, IP Fragment Pointer, IP Protocol, Checksum, Source Port, Destination Port and Attack_type, shown in example of data in Table 1.

In the data preparing process, there were 4,895 sets of data; 3,302 of them were normal and other 1,593 were abnormal, that can be divided into 10 categories that were apache2, back, crash, dosnuke, mailbomb, neptune, processtable, smurf, udpstrom and teardrop [4, 9]

In this research, the data were divided into 2 groups that were:

- Data group of training (80% of all data)

- Data group of testing (20% of all data)

Table 1: *Example of data in the experiment*

IP				Transport Layer			Attack_Type
Len	FrgPtr	TTL	Proto	PortSrc	PortDst	CkSum	
44	4000	255	6	32973	80	TCPTrue	normal
40	4000	60	6	20564	21	TCPTrue	normal
44	0	63	6	1052	25	TCPTrue	processtb
44	0	62	17	7	7	UDPFalse	udpstorm
528	0	254	1	0	0	ICMPFalse	Smurf
40	4000	64	6	24384	80	TCPTrue	Crashiis
62	4000	63	6	17464	25	TCPTrue	mailbomb
44	4000	127	6	1216	139	TCPTrue	dosnuke
1064	4000	62	6	1491	80	TCPTrue	apache2
44	0	62	6	2164	80	TCPTrue	back
40	0	253	6	2855	389	TCPTrue	neptune
56	0	62	6	24891	23	UDPTrue	teardrop

3.2 Bayesian Learning

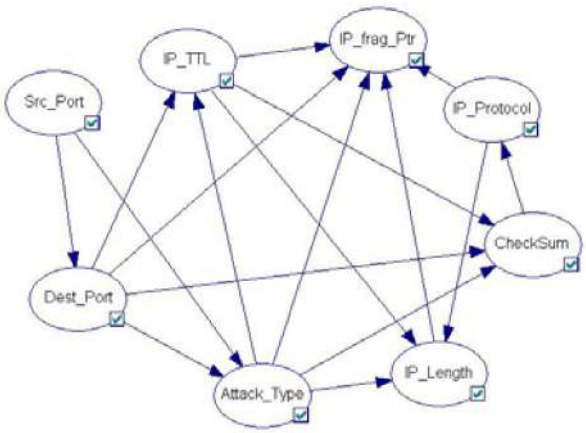

Fig.3: *Bayesian network structure*

3.2.1 Structure Learning

It is the process of adjusting the value of prepared data in table 1. In some cases, variables were continuous; the data were adjusted to discrete by using the range adjusting to the relation of data.

The process of adjusting the learning structure used the greedy algorithm [8] with computing score of K2- algorithm [2, 8] and the properties of field of packet header that previously determined the relations to gain more efficiency. The experimental outcomes are shown in Bayesian network structure in fig. 3.

3.2.2 Parameter Learning

It is the process of taking data and Bayesian network structure from the previous structure learning to adjust the value of parameter learning for learning of each variable, or node, using the Expectation Maximization Algorithm (EM) [3, 7] in the adjusting value of parameter learning and resulting in the conditional probability table (CPT) of each node in Bayesian network as shown in fig. 4.

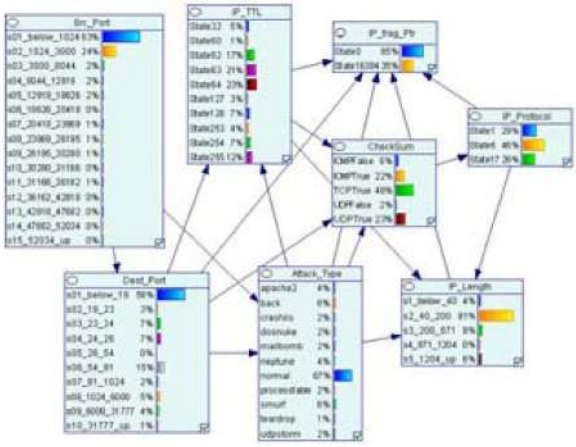

Fig.4: *Bayesian network and CPT table*

Table 2: *Probability of variables (of each node)*

IP Protocol	
Status	Probability
ICMP (1)	0.285896
TCP (6)	0.458812
UDP (17)	0.255292

IP Fragment Pointer	
Status	Probability
0x000	0.654448
0x4000	0.345552

IP Length	
Status	Probability
Less than 40	0.0390207
40 – 200	0.810508
200 – 871	0.089773
871 – 1204	0.0048457
Greater than 1204	0.0558531

Check Sum	
Status	Probability
ICMPFalse	0.0614639
ICMPTrue	0.224433
TCPTrue	0.458812
UDPFalse	0.020403
UDPTrue	0.234889

Source Port	
Status	Probability
Less than 1024	0.630961
1024 – 3000	0.2405
3000 – 8044	0.0234634
8044 – 12919	0.0201479
12919 – 18626	0.0175976
18626 – 20418	0.0048457
20418 – 23969	0.0127518
23969 – 26195	0.0124968
26195 – 30280	0.0122418
30280 – 31166	0.00357052
31166 – 36162	0.0102015
36162 – 42818	0.00331548
42818 – 47682	0.00306044
47682 – 52034	0.00306044
Greater than 52034	0.00178526

Destination Port	
Status	Probability
Less than 19	0.558531
19-23	0.0252487
23-24	0.0749809
24-26	0.06886
26-54	0
54-81	0.152002
81-1024	0.020403
1024-6000	0.0525376
6000-31777	0.0397858
Greater than 31777	0.00765111

Attack_type	
Status	Probability
Apache2	0.0448865
Back	0.0614639
Crashiis	0.020403
Dosnuke	0.020403
Mailbomb	0.0232084
Neptune	0.0438664
Normal	0.673808
Processtable	0.0198929
Smurf	0.0614639
Teardrop	0.0102015
Udpstorm	0.020403

IP TTL	
Status	Probability
32	0.0453966
60	0.00561081
62	0.165519
63	0.214741
64	0.234124
127	0.0321347
128	0.0696251
253	0.0438664
254	0.0711553
255	0.117827

3.3 Experimental Outcomes

According to this research, 8 variables and 3,921 set of training data; 2,642 are normal and 1,279 are abnormal. The outcomes are shown in Table 2.

3.4 Summary

From the experiment, it was found that Bayesian network structure constructed was able to represent the less using of network systems. To examine the correctness of intrusion detection or anomaly detection, Confusion Matrices [10] with reliability value and precision value were used in the efficiency test. In this research, the data examined were 974 sets of data prepared; 660 of them were normal and 314 sets from 10 categories were abnormal. The results of the efficiency test are shown in Table 3.

From Table 3, types of intrusion or anomaly like apache2, crashiis and abnormal data were examined for the precision value which resulted to 93.18, 78.95 and 98.48 percent as in order. In conclusion, the efficiency test of intrusion detection or anomaly detection of Packet on the DARPA network system had the average reliability value of 99.04 % and average precision value of 97.33 %.

Table 3: *The efficiency test: reliability value and precision value*

Attack_type	Predicted	Real	Reliability (%)	Precision (%)
apache2	41	44	100.00	93.18
back	66	59	89.39	100.00
crashiis	15	19	100.00	78.95
dosnuke	20	20	100.00	100.00
mailbomb	21	21	100.00	100.00
neptune	42	42	100.00	100.00
processtable	20	20	100.00	100.00
smurf	59	59	100.00	100.00
udpstorm	20	20	100.00	100.00
teardrop	10	10	100.00	100.00
normal	650	660	100.00	98.48
Average			**99.04**	**97.33**

4. CONCLUSIONS

In this research study, researcher has experimented by taking parts of the data on a real DARPA network system under the intrusion of DoS and normal data. In the next research, the following studies are recommended.

1. Use sets of data of other different intrusions, besides those already examined, to increase the efficiency of working. Then, involve other variables to the fundamental variables to gain more efficiency of Bayesian network constructing.

2. Time when analyzing the data on the Bayesian network constructing is not studied. In case of huge amount of data, the system may be slow down.

References

[1] Bunserm Kitsirikul (2003), Document to Artificial Intelligence course, Department of Computer Engineering, Chulalongkorn University.

[2] Cooper and Herskovits (1992), A Bayesian Method for the Induction of Probabilistic Networks from Data, Machine Learning, 9, 309-347.

[3] Dempster, A.p., Laird, N. M. and Rubin D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B, 39 (1), 1-38.

[4] J. Krister and S Lee (2003), Bayesian Network Intrusion Detection (BNIDS), CS424 Network Security. May 3, 2003.

[5] Lippmann, R., et al. (2000), The 1999 DARPA Off- Line Intrusion Detection Evaluation, Computer Networks 34(4) 579-595, 2000.

[6] M. Mahoney & P. K. Chan (2001), PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic, Florida Institute of Technology Technical report CS-2001-04.

[7] McLachlasn, G. J. and Krishman, T. (1996) , The EM Algorithm and Extensions, Wiley Interscience.

[8] Murphy K. (2004), Bayes Net Toolbox for Matlab, retrieved August 23, 2006, http://bnt.sourceforge.net/

[9] N. S. Abouzakhar, A Gani and G Manson (2003), Bayesian Learning Networks Approach to Cybercrime Detection, The Centre for Mobile Communications Research (C4MCR).

[10] DNasser S. Abouzakhar, Gordon A. Manson, (2006), Evaluation of Intelligent Intrusion Detection models, retrieved August 23, 2006, http://www.ijde.org.

Mongkhon Thakong received B.S. in Mathematics and the M.S. in Computer Science from Khon Kaen University. His research interests include Computer Network and Artificial Intelligence. Currently, he serves as a lecturer in the Faculty of Science, Udonthani Rajabhat University, Thailand.

Sartra Wongthanavasu received M.S. in Computer Science from Illinois Institute of Technology (IIT) inUSA and Ph.D. in Computer Science from Asian Institute of Technology (AIT), Thailand. His research interests cover Machine Learning, Image Processing.