JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 22, 1447-1464 (2006)

Effective Intrusion Detection Model through the Combination of a Signature-based Intrusion Detection System and a Machine Learning-based Intrusion Detection System*

ILL-YOUNG WEON, DOO HEON SONG+ AND CHANG-HOON LEE
Department of Computer Science and Engineering
Konkuk University
Seoul 143-701, Korea
E-mail: {clcc; chlee}@konkuk.ac.kr
+*Department of Computer Game and Information*
Yong-In SongDam Colleage
5771 Mapyong Dong Young-In Kyungki, Korea
E-mail: dsong@ysc.ac.kr

In the field of network intrusion detection, both the signature-based intrusion detection system and the machine learning-based intrusion detection system possess advantages and disadvantages. When the two discrepant systems are combined in a way that the former is used as the main system and the latter as a supporting system, the machine learning-based intrusion detection system measures the validity of alarms determined by the signature-based intrusion detection system and filters out any false alarms. What is more, such an approach can also detect attacks that the signature-based system by itself cannot detect.

The objective of this paper is to propose a combined model of the signature-based and machine learning-based intrusion detection systems and to show that the combined system is more efficient than each individual system. We used the DARPA Data Set in experiments in order to show the usefulness of the combined model. Snort was used for the experiment as a signature-based intrusion detection system and extended IBL (Instance-based Learner) was used as the principal learning algorithm for the machine learning-based intrusion detection system. To compare performances of the algorithms, C4.5 was used.

Keywords: network intrusion detection system, machine learning, combined model, false alarm, detection of new attack, instance-based learner

1. INTRODUCTION

Most of the network-based intrusion detection systems that are recently being used are misuse systems based on the signatures created by network and hacking experts. This signature method has weaknesses in detecting unregistered types of attacks and/or the variations of existing attack types. When the experts detect any the attacks, which have not been detected by the provided signature, they extract the characteristics of the attacks by the analysis of a log-in process and various statistics to make a detection rule, which requires a lot of time and human resources [1].

Received October 8, 2004; revised May 5, 2005; accepted June 8, 2005.
Communicated by Ja-Ling Wu.
* This research was supported by the Ministry of Information and Communication (MIC), Korea, under the Information Technology Research Center (ITRC) support program supervised by the Institute of Information Technology Assessment (IITA).

In order to reduce the efforts of experts while improving performance of intrusion detection systems, various methods of artificial intelligence have been introduced. Among them and that which receives much attention is the machine learning-based or data mining-based intrusion detection method [2-5] which, unlike the misuse method, models the status of networks, creates knowledge on the status of the network using an artificial intelligent machine learning technique, and determines the normality of the network in the real environment. In this sense, a network that is "normal" means that attacks are not included in the flow of packets, while when a network is said to be "abnormal", this indicates that there are attacks included in the flow of the network or it is suspected to be so. The advantage of this method is that it requires no human support in the process of creating, learning, and judging the original data. The problem with this method is that performance depends on the attributes that are used as the basis of learning. When attributes are organized that focus on statistics used in the signature-based method, the reaction to new attacks decreases. When they depend on packet header information, most attacks occur as the result of the sequence of different packets, creating the problem of context sensitivity in which identical packets produce different results depending on the status of preceding packets. Hence, the perception of abnormal states is no better than that of the misuse method; instead, the perception rate for normal states is rather higher.

One of the trends in recent studies of network intrusion detection systems is to complement the different features of signature-based IDS (intrusion detection systems) and machine learning-based IDS; that is, the high attack perception rate of signature-based IDS and the 'normal' modeling of machine learning-based IDS [19, 20]. One important thing that should be noted in this study is that the signature-based method, adopted by most common IDS, has a high perception rate of attacks, but false positive rates increase exponentially proportionate to the increase in the detection rules, which is hard to deal with by administrators [4, 6]. According to a recent study, among the thousands of alarms created in a day, only five or six are worthy of being noted and others are alerts that accompany those alarms [7]. Moreover, it has been found that it is possible to deliberately create packets that continuously detect intrusions aiming at the known attack detection algorithm of the target signature-based IDS. Thus, the issue of removing false alarms is the core problem that must be solved for performance improvement in misuses of IDS [1].

Therefore, it is important to improve efficiency by reducing false alarms while not affecting the attack detection rate of the signature-based IDS. In order to solve this problem, studies have been performed in two directions. One is to use a data mining method, which makes the knowledge-based preceding filters treat only important alarms. When applying this method, it is reported to bring a 30% decrease in false alarms in real systems [7, 8]. However, this method inevitably neglects attacks that do not frequently occur, due to its structure, causing a loss of perception in abnormal states and unsatisfactory reaction to new attacks. The other is a method called a behavior-based analysis method. It determines whether it is necessary to inform the administrator through a learning or a clustering method when the signature-based IDS gives off alarms. The system load for this method is higher than that of the preceding filter method, albeit, there is no loss of information. Therefore, it will be effective if learning-based and data mining-based methods are adequately combined.

This paper is a study of a model that combines signature-based IDS and machine learning-based IDS into a behavior-based analysis method. Snort (1.8.1) [10, 13] was used as a signature-based system and XIBL (eXtended Instance Based Learner) which was extended from IBL (Instance Based Learner) [11] was used as a machine learning-based method. The combined model proposed in this paper performs back-tracking on the packets in certain amounts when Snort gives off alarms, starting from the time when the alarm is given. The alert patterns for XIBL on these packets are then investigated in order to measure accurately the reliability of Snort on alerts. It was also designed to alert of attacks that Snort did not alert to through judgment by the learning-based IDS itself. This model conforms to the demand of the real environment in which signature-based IDS is to be used as the main system with learning-based IDS as the support system. It was also shown through the DARPA 1998 data set [12] that performance would be better than that when signature-based IDS were used alone.

The organization of this paper is as follows: In section 2, we have explained the combined model of the signature-based and the machine learning-based intrusion detection systems. In section 3, XIBL, a learning algorithm is briefly mentioned and the validity of the model that reduces errors in the signature-based intrusion system is reviewed. In section 4, you will find the validity of the combined model that detects the attacks that cannot be detected by the signature-based intrusion detection system alone. In section 5, the tests on the performance of the combined system that were conducted are reported. In section 6, the conclusions of this paper and the tasks to be done in the future are mentioned.

2. COMBINED INTRUSION DETECTION SYSTEM MODEL

2.1 XIBL

In an IBL algorithm, the method for extracting knowledge from training data is represented by the data itself; that is, the method is not based on the conceptual method that focuses on attributes as shown in C4.5 [11]. The knowledge acquired through this method is called Partial Concept Description (PCD), each of which has the property of a representative that represents a specific and different class, respectively. The IBL algorithm for class determination finds the k number of instances in PCD, which are most similar to the newly input data and then determines the classes of instances according to the distribution of their classes. This algorithm is accurate in many aspects and could be compared to that of C4.5. Although the learning speed is slower than that of C4.5, the validation speed of the new data according to the learned results and the stability according to data change are better than those of C4.5.

The two biggest features of the IBL algorithm are the method for determining the instances to be stored in PCD and the method for measuring the distances between instances. Aha, in his doctoral dissertation, suggested four versions regarding the method for determining instances in PCD. The method (IB1) that stores all the instances contains a overflow problem in memory for mass data. Hence, another method (IB2) was presented. This method stores only the instances having a representative property, but it was vulnerable to noise. To complement these problems, the IB3 algorithm and the attribute-

weighted algorithm (IB4) were developed. IB3 has the function of removing noise and IB4 weakens the disturbances from unnecessary attributes. In practice, IB3 is used more frequently because IB3 is superior to IB4 in many aspects.

In the measurement of distances between instances, Euclidean or Manhattan Distance is used for real number data. As for discrete type data, 0 is set in case of identical attributes and 1 is set in the case of the different attributes in the measurement of distances.

Our algorithm, XIBL has some more functions added to the original IBL.

First, since the distance between instances of discrete type data is represented by 0 or 1, the actual weight is higher than for those of real number type data which show a value between 0 and 1 according to normalization. In order to reflect these differences, we have applied the Value Difference Metric (VDM) [15] that converts the distances between discrete type data in XIBL into continuous values between 0 and 1.

Second, IBL is sensitive to noise, and in order to solve this problem, we have applied a Leave-one-out noise filter [16] which is based upon mathematical statistics.

Third, in setting the weighted value for attributes, we have applied the method of estimating attribute value according to a backward stepwise regression, which was statistically validated, instead of a reward/penalty method from among the learning-based methods adopted by IB4.

The performance of XIBL, which has been developed in this method, is better than those of C4.5 and the original IBL [3, 17]. The instances in PCD are 8-10% of the learned instances and they have stability especially in reacting with mass data [17]. A detailed discussion on XIBL and an analysis of the characteristics are beyond the scope of this thesis and thus shall not be mentioned here; however, we have already mentioned it in the thesis [3, 18].

2.2 System Model

Snort [9] sequentially processes the packets that occur from the network according to each protocol specification, compares the attack signatures, and then signals when they conform to the corresponding stages. The machine learning-based intrusion detection system, on the other hand, creates an event for each packet from the network and learns the normal and abnormal patterns using the events. In real detection, they are classified into normal or abnormal patterns based on the learned knowledge.

The basic ideas of the proposed model are as follows:

In order to examine the truth of the Snort attack signals, the rate of packets that are classified into abnormal patterns is obtained through examining a certain number of packets that are classified in machine learning-based intrusion detection by back tracking at the time of giving the alarm. The signal is given to the administrator when the obtained rate for the abnormal pattern exceeds the standard value. Further, in order to detect the attacks not perceived by Snort, when the consecutive abnormal frequency exceeds a certain limit after examining the classified events of the machine learning-based intrusion detection system, the administrator will be informed of the attacks that are not perceived by Snort. We shall define the former abnormal rate as “Alpha-Cut” and the latter consecutive abnormal numbers as “Beta-Pick”. Fig. 1 shows the concepts in detail.

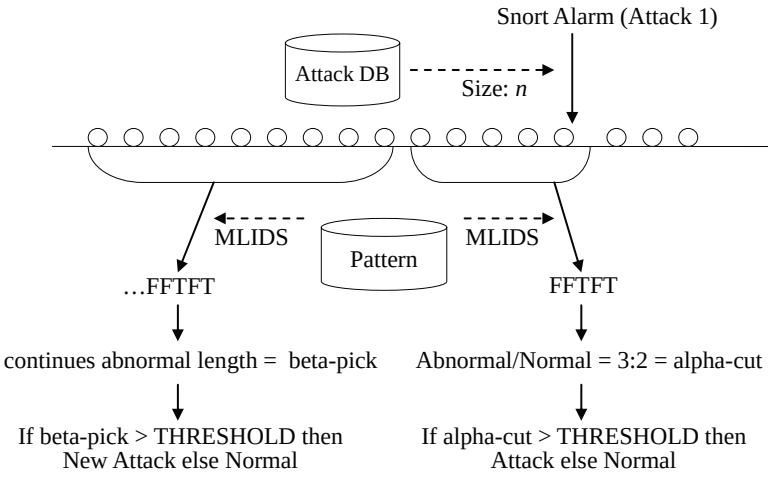

Fig. 1. System model.

The signals of the intrusion detection system are generally classified as the following: true negative (when a normal state is to be noticed as normal), false positive (when a normal state is to be noticed as abnormal), false negative (when an abnormal state is to be noticed as normal), and a true positive (when an abnormal state is to be noticed as abnormal). Alpha-Cut is related to false positive and true positive and its objective is to increase the true positive and reduce the false positive. In other words, there are two types of attack signals that Snort alerts. One is the signal that alarms of an attack for a true attack and the other is the signal that alarms of an attack for the normal state. By applying the Alpha-Cut we can estimate the reliability of Snort's response to the attack signals. The objective of Beta-Pick is to reduce the false negative. When Snort perceives a normal state (Snort does not give off any alarms in this case), the signals are classified into alarming normal state for attacks and alarming normal state for normal state. By applying Beta-Pick, the reliability of the packets that turned out to be normal could be estimated.

The main idea of the above proposed model is based upon the hypothesis that there exists a method that could model the adequate states of normal and abnormal that occurs from the network and that the modeled data could be learned and classified by using machine learning. We will try to prove the validity of the hypothesis through the following experiment.

For the experiment, we organized the system as the following: Snort (1.8.1) was used as the signature-based intrusion detection system and the environment for the experiment was set according to the basic environment setting distributed by the website, www.snort.org. Machine learning-based intrusion detection system was developed for ourselves and used. The learning algorithm used in the machine learning-based intrusion detection system was XIBL [12] that has been expanded from the IBL and the C4.5 [13] was used for the comparative experiment of learning algorithms. Fig. 2 shows the organization of the system.

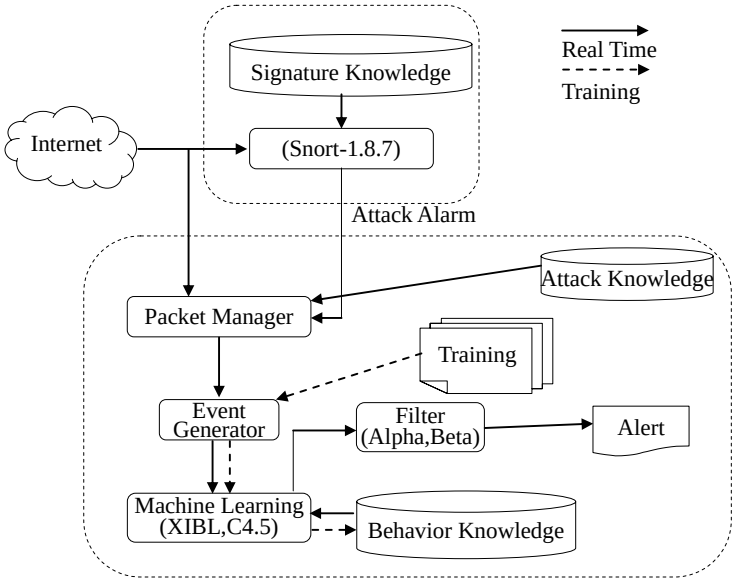

Fig. 2. Experiment system.

In Fig. 2, the average number of packets that form the known attack types are stored in Attack Knowledge. The average number of packets of registered packets is used for the attack types that are not registered. Using the cases of existing normal and abnormal states, the learning engine automatically forms Behavior Knowledge. Then, whenever Snort gives off signals, Alpha-Cut is calculated and when there are no Snort signals, Beta-Pick is measured and the signal will be made if the value exceeds a certain level.

3. ALPHA-CUT

In this section, we will try to prove the existence of the proposed Alpha-Cut through an experiment. For the experiment, we used the DARPA Data Set of 1998. For consistency in the meaning of terms, we define *true positive* for cases where what is estimated to be abnormal actually turns out to be abnormal, *false positive* for cases where what is estimated to be abnormal turns out to be normal, *true negative* for cases where what is estimated to be abnormal actually turns out to be abnormal, and *false negative* for the cases where what is estimated to be normal turns out to be abnormal.

3.1 XIBL Training

For learning knowledge of XIBL, the packets related to corresponding attacks were separated according to each day of the week, from the Training Data Set of 1998 and the normal packets were separated in the same proportion. The data set for the experiment was composed of normal and abnormal packets of the same proportion. Then, the data was divided in a proportion of 30:70. 30% of all data was used for the measurement of the characteristics and performance of XIBL and the remaining 70% was used for XIBL

training. The ratio of normal and abnormal packets within a single set of separated data was 50:50. Training data were divided into 10 folders and then 10CV (Cross Validation) was conducted. Among the data, the PCD with the best performance was specified as the learning knowledge. Using the selected PCD and the entire data, the experiment on the possibility of the proposed model was conducted.

The objective of this experiment was to analyze the behavior of XIBL responding to the Snort alarm and to determine whether a distinctive pattern between true alarms and false alarms existed. If there were any, it could be inferred that the proposed model is valid theoretically.

The factors that were used in the experiment are as follows:

The number of instances that were referred in order to judge normal and abnormal was set to 1 to remove other variables such as voting. The number of back-tracking packets of the combined model was set to 100 and was based upon the average number of packets according to prior attack types. The number of abnormal packets and normal packets used in learning was 44,726 and 45,453, respectively; 8% of which was the optimal PCD (normal: 365, abnormal: 294), and the fitting rate was 99.4%.

In order to test the unique detection ability of XIBL, training was conducted for only 70% of the data. Then performance was tested with the remaining 30% of the data. The data extracted from the attack log-in were all labeled attack; that is, if a certain attack was composed of 10 packets, these 10 packets were all labeled attack. The data from normal log-ins were all labeled normal. When the classified experiment data were tested in XIBL, 68.93% of the attacks that had been labeled attack were actually detected as attacks and 88.05% of normal data that had been labeled normal were actually detected as normal. The 11.95% of normal were mistakenly perceived as attacks and 31.07% of attacks were mistakenly perceived as normal.

3.2 Analysis of True Alarm

In this section, we conducted an experiment on true alarms when various attack data were sent to Snort. Table 1 shows the part of the alert results of XIBL regarding 35 attacks that properly signaled alert, out of 82 types of attacks.

For instance, among the back attack types, in the attack on Friday of the second week (2_fri_back), the observed number of packets was 219. Snort detected these attacks and 195 of those were detected as normal and 24 were detected as abnormal. Hence, the percentage of the abnormal detected in XIBL was approximately 10%. The objective of this analysis was to observe the behavior patterns of XIBL according to Snort detection. The sizes of the observed packets in the attacks are not consistent because the start and the end of each attack are not clearly defined and the duration of attacks was arbitrarily determined by the attacker.

3.3 Analysis of False Alarm

In this section, we conducted an experiment on the results of the XIBL alert which Snort mistakenly perceived as attack when normal data were sent to Snort. The XIBL alert was measured according to the corresponding length after examining the corresponding attack type when an alert by Snort occurs.

Table 1. Alert results of XIBL for true alarms.

Attack Type	Attack Name	# of Packet	XIBL Result		Abnormal rate(%)
			# of normal	# of abnormal	
Back	2_fri_back	219	195	24	10.9589
	3_wed_back	222	5	217	97.74775
	6_wed_back	4350	3	4347	99.93103
	7_fri_back	4195	5	4190	99.88081
Phf	3_mon_phf	9	8	1	11.11111
	6_mon_phf	7	2	5	71.42857
	7_mon_phf	7	2	5	71.42857
	7_wed_phf	7	6	1	14.28571
Portsweep	2_mon_portsweep	216	0	216	100
	3_mon_portsweep	336	0	336	100
	4_thurs_portsweep	1129	0	1129	100
	5_fri_portsweep	199	0	199	100
	…	…	…	…	…
Satan	3_mon_satan	84	5	80	94.11765
	4_tues_satan	226	215	11	4.867257
	6_mon_satan	12	6	6	50

For instance, when the normal data from Friday of the 7th week (7_fri) was sent to Snort, Snort misinterpreted the data as a back attack and gave a false alarm. At the point when the same data were sent to XIL and Snort alarmed of an attack, we observed the reaction of XIBL to 4881 average packets of the back attack type that were observed in the attack data and found that the rate of abnormal data was 0.67%. As for the calculation of the average length of packets of each attack type, we considered both cases where Snort perceived the attack and where Snort did not. The objective of this analysis was to observe the behavior pattern of XIBL when a false alarm was given by Snort.

3.4 Analysis of Combined Model

Analysis of the results of the experiments conducted in the previous sections shows that when the attacks consist of packets of adequate lengths, XIBL reacts to true alarms over 50% of all alarms, whereas it reacts to false alarms less than 50% of the alarms. This indicates that when the flow of packets, which consist of attacks, are analyzed with the standard level set at 50%, the differences between true and false alarms could be perceived. Although there were some exceptions (e.g. 2_fri_back, 3_ mon_phf), they could be classified under the standard in most cases. Among the overall experiment data set, there were 4 different types that had both true and false alarms in the DARPA data set of 1998. The reactions of each type are shown in Fig. 3. In other words, Fig. 3 includes the results of the experiments of both Tables 1 and 2 in order to consider both cases of Tables 1 and 2 at the same time.

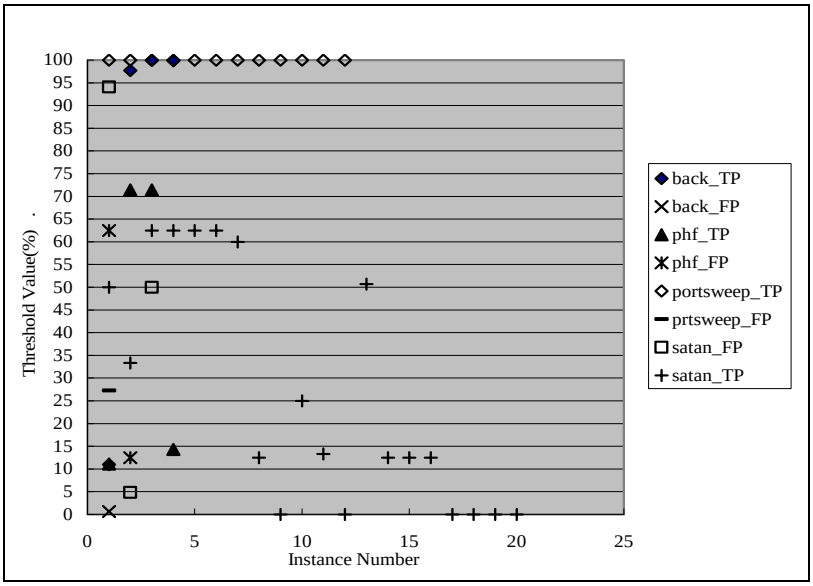

Fig. 3. Threshold value of combined model.

Table 2. Alert results of XIBL for false alarms.

Attack Type	Normal type file Name	XIBL Result		Abnormal rate (%)
		# normal	# abnormal	
Back	7_fri	4881	33	0.671551
Phf	3_wed	3	5	62.5
	3_fri	7	1	12.5
Portsweep	3_fri	16	6	27.27273
Satan	1_mon	4	4	50
	1_thur	10	5	33.33333
	2_fri	3	5	62.5
	3_mon	3	5	62.5
	3_wed	3	5	62.5
	…	…	…	

The vertical axis shows the rate of abnormality using XIBL in percentages, while the attack numbers of each attack type are depicted on the horizontal axis. The postfix TP attached at the end of the title of attack types implies a true positive and FP stands for false positive. The abnormalities of XIBL on the Snort alarm were measured 5 times for "Back" type, 6 times for "phf" type, 13 times for "Portsweep" type, and 23 times for "Satan" type.

As shown in Fig. 3, applying the results of XIBL alerts, there exist patterns that distinguish false positive from true negative when Snort alarms of an attack. For instance, when Snort alarms "Satan" and the abnormality at that point is measured by using XIBL, let us suppose that the administrator perceives the alarm as attack when the abnormality by XIBL is 65% or higher. According to the results above, all the true alarms given by

Snort related to Satan would be alarmed while all the false alarms given by Snort would not be alarmed at all. Thus, the false alarms could be completely avoided while maintaining the rate of true alarm at the same level as that in the cases when alarming by Snort only.

However, it cannot be said that an accurate judgment of all Snort alarms is possible. For instance, when the standard abnormality is set to 30%, there will be some misjudgment in the case of the Satan-type. In addition, there are various methods for applying abnormalities; applying abnormalities separately for each attack; and applying a single value for all attacks. Since Snort detects about 200 unknown types of attacks in the DARPA data, the issue of the application of abnormalities to each type is a matter of implementation rather than theoretical basis. In section 5, there is detailed explanation on the experiment that measures the effectiveness of these methods quantitatively in the real environment.

4. BETA-PICK

Beta-Pick refers to the number of consecutive packets that are classified as abnormal packets by the learning engine and are based upon the judgment of the machine learning-based intrusion detection system of the packets that Snort perceived as normal packets. When the Value of Beta-Pick is 90, it means that among the packets that are perceived as normal packets by Snort, there are 90 consecutive packets that are classified as abnormal packets by the machine learning-based intrusion detection system.

Table 3. A part of the values of Beta-Pick of the 46 attacks that were not detected by Snort.

ID	Attack_Name	Beta-Pick
5	Dict	90
12	Eject	364
16	Ffb	65
20	Ffb	17
24	Format	117
28	Guest	3
37	Loadmodule	88
44	Neptune	138
45	Neptune	160
46	Neptune	119
51	perlmagic	49
59	portsweep	216
71	Rootkit	228
72	Rootkit	200
76	Spy	1
…	…	…

Table 4. An analysis of Beta-Pick distribution of normal data for the year 1998.

Beta-Pick	# of Pattern
1	1636
2	458
3	75
4	30
5	13
6	10
7	8
8	7
9	4
10	4
11	4
12	3
13	3
14	2
15	1
16	1
17	0

The data and environment for the experiment are identical to those in section 3.1. From the data set of the year 1998, a total of 81 attacks were selected and used in the experiment. When these data were sent to Snort and examined, 35 were detected while 46 were not detected under the Snort environment setting.

Table 3 shows a part of the values of Beta-Pick of the 46 attacks that were not detected by Snort and measured in the machine learning-based intrusion detection system. ID numbers in Table 3 are unique numbers assigned to each attack for the convenience of the experiment. Attack_Name represents the name of the attack. Table 3 shows that most of the attacks have the values of Beta-Pick exceeding the certain level.

Beta-Pick could increase true positive and false positive at the same time. Therefore, it is necessary to study the distribution of Beta-Pick in normal packets. Table 4 below represents an analysis of Beta-Pick distribution of normal data for the year 1998 that were used in the experiment. When the value of Beta-Pick is 5 and the number of patterns is 13, it indicates that when the Beta-Picks of the normal data of the year 1998 were measured, there were 13 cases when Beta-Pick was classified as abnormal packets 5 consecutive times.

Comparing Tables 3 and 4, it was found that there exists a proper Beta-Pick threshold which increases the true positive but does not increase the false positive significantly. It increased in the DARPA Data Set of 1998. That is, when an adequate standard of Beta-Pick value was applied, it is possible to detect all the attacks that were not detected by Snort in the combined system. It does not mean; however, that it is possible to detect all the attacks in the combined system, which has not been detected by Snort by simply applying Beta-Pick. Beta-Pick has no significant meaning for the attack types that consist of a small number of packets.

There are some reasons of the great variation of perceived rate. First the reprocess of data has some mistakes. Second the size of packets is not suitable for Alpha-cut and Beta-pick. So, through the more improved method of experiment, we need to search for the suitable packet size for Alpha-cut and Beta-pick by reducing preprocess errors.

5. PERFORMANCE TEST OF THE COMBINED MODEL

In this section, we shall analyze the performance and characteristics of the proposed combined model intrusion detection system by comparing the results with those of the signals of the signature-based intrusion detection system.

5.1 Experiment Data and Methods of Alarming

For the experiment, we used the DARPA Data Sets of 1998 and 1999. The 1998 data were used for learning the intrusion detection engine and the 1999 data were used for measuring the performance of the model in each case. The 1998 data were created by using the method described in 3.1 and used in learning. The 1999 data [16] were classified into attack data and normal data. In the case of attack data, 85 different types of attack data were selected randomly and 137 attack data were sampled (consisting of 127,405 packets in total). All of the data from the Monday of the 1st week were used as normal data (consisting of 1,369,134 packets in total). Table 5 shows parts of the abnormal data we used.

Table 5. A part of abnormal data from 1999.

Attack Type	Attack Name
Denial of Service Attacks	arppoison (New in 1999 test) dosnuke (New in 1999 test) selfping (New in 1999 test) …
User to Root Attacks	anypw (New in 1999 test) casesen (New in 1999 test) ntfsdos (New in 1999 test) …
…	…

Table 6. The process of signaling final alarms.

```
Key:
AC: Alpha-cut value
BP: Beta-pick value
K: Packet size of attack
AK : Average packet size of known type attack (for unknown type attack)
LA: Learning algorithm

DECISION RULE()
IF   SNORT triggers an alarm on input packet x   THEN
     IF   the type of attack is "known"   THEN
          t race back to K packets
            examine K packets by LA
     ELSE
           t race back to AK packets
           examine AK packets by LA
     ENDIF
     get   AC
     IF   AC >= Alpha-threshold   THEN
          send alarm
     ELSE
          pass that packet x
   ENDIF
ELSE
     examine BP with the same source IP
     IF   BP >= Beta-threshold THEN
          send alarm
   ENDIF
ENDIF
```

In order to enable the machine learning-based intrusion detection system to model the state of the network, we used the method that creates one event (learning material) per network packet. The attribute set that constituted one event was sampled from a TCP/IP packet header and details related to this are explained in [12].

In Table 6, we have described the process of signaling final alarms in mnemonic

code by applying the Alpha-Cut and Beta-Pick in the combined model. The process has the Alpha-Cut and Beta-Pick threshold as a parameter. The maximum number of packets (AK) for unknown attack types in the measurement of the Alpha-Cut was set at 100. This value was based upon the observation of Alpha-Cut and Beta-Pick using the 1998 data. We determine the *K* with the average number of packet sizes of every attack type. But some attacks have different number of packets according to the attacker's intention. In this case, we use average number of packets.

5.2 Results of the Experiment

The results of the experiment were measured by the Alpha-Cut and Beta-Pick, respectively, on attack data and normal data. The Alpha-Cut was measured from 10 to 90 with an interval of 10 and the Beta-Pick was measured on 5, 10, 15, 20, 30, 50, 60, and 70. The number of attacks detected by the system was recorded on the attack data where more than two alarms on the same attack were recorded as a one-time detection. On the other hand, since the alarms that occurred from the normal data were all false alarms, the results were measured mainly focusing on the number of alarms. Fig. 4 recorded the number of attack detections when the Alpha-Cut was applied in the combined system on the attack data that Snort detected as attacks among the data of 137 attacks. Fig. 5 recorded the number of attack detections when the Beta-Pick was applied in the combined system on the attack data that Snort could not detect as attacks among the 137 attacks.

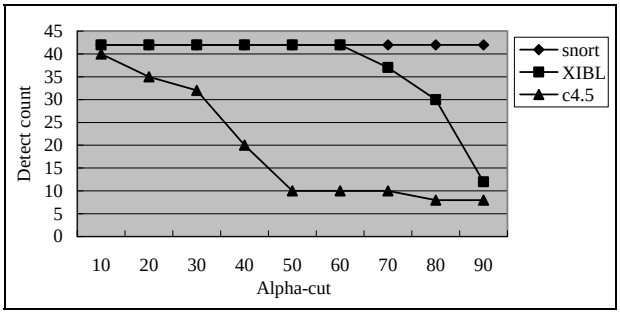

Fig. 4. Application of the Alpha-Cut on the attacks detected by Snort.

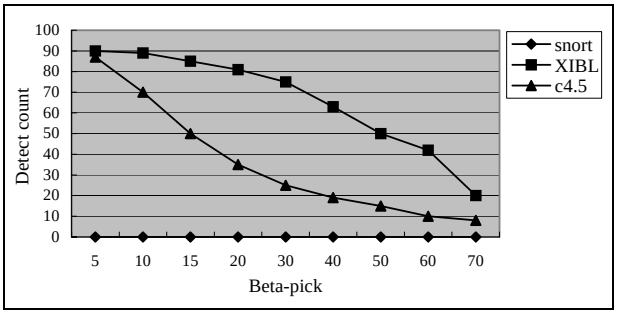

Fig. 5. Application of the Beta-Pick on the attacks that were not detected by Snort.

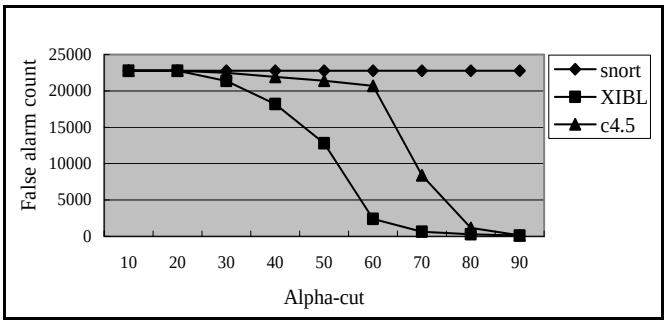

Fig. 6. Application of the Alpha-Cut on the false alarms given by Snort on normal data.

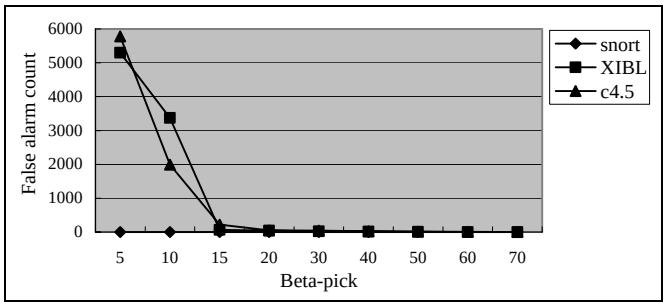

Fig. 7. Application of the Beta-Pick on the false alarms on normal data.

Fig. 6 recorded the number of alarms to be decreased by applying the Alpha-Cut on the false alarms that Snort gave in the normal data. Application of the Beta-Pick caused additional false alarms that Snort did not give off in the normal data and the results were recorded in Fig. 7.

In order to set a proper threshold for the Alpha-Cut in the combined system, Figs. 4 and 6 should be considered at the same time. For this purpose, it is important to maintain the number of alarms detected by Snort at maximum in Fig. 4 while reducing the number of false alarms at maximum in Fig. 6. On the other hand, in order to set a proper threshold for the Beta-Pick in the combined system, Figs. 5 and 7 should be considered at the same time. For this purpose, the number of detections should be increased at maximum in Fig. 5 while the number of false alarms should be reduced at maximum in Fig. 7.

Table 7 is an example of an integrated analysis when the threshold of the Alpha-Cut was set to 50 and the threshold of the Beta-Pick was set to 20. These threshold values were selected as they were regarded to be appropriate under the universal operating environment based on the experiment results shown in the previous page and [1]. On the 137 attack trials, a single Snort system detected 42. In case of the combined system, XIBL detected 100% when the Alpha-Cut was applied to the attacks detected by Snort. As for C4.5, it only detected 10 times, which showed a 23.8% detection rate compared to a single Snort system. Snort gave off 22,787 false alarms on the normal data for the test. The XIBL, in which the Alpha-Cut was applied to the false alarms, gave off 12,791 false alarms, which was reduced 43.3% compared to Snort.

Table 7. An integrated analysis example of performance test.

Original Data	Single System	Combined System		
	Snort	Algorithm / Filter	XIBL	C4.5
Abnormal (137times try)	Abnormal (42 times detect)	Alpha-cut	Abnormal (42times detect)	Abnormal (10 times detect)
			Normal	Normal
	Normal (95 times miss)	Beta-pick	Abnormal (81 times detect)	Abnormal (35 times detect)
			Normal	Normal
Normal	Abnormal (22,787 alarm)	Alpha-cut	Abnormal (12,791 alarm)	Abnormal (21,403 alarm)
			Normal	Normal
	Normal	Beta-pick	Abnormal (43 alarm)	Abnormal (53 alarm)
			Normal	Normal

However, the false alarms that were not given by Snort occurred in XIBL to which the Beta-Pick was applied 43 times and it was relatively low. In the case of C4.5, when the Alpha-Cut was applied to the 22,787 false alarms given by Snort, C4.5 gave 21,403 alarms, which reduced the false alarms by about 6%. When the Beta-Pick was applied, the false alarms that were not given by Snort occurred 53 times.

In terms of the number of alarms, the results of the experiment above show that the number of alarms in the combined models decreases to the level of 56.75% of those of the Snort model. In terms of the quality of alarms, the alarms detected by the combined models include most of the alarms detected by the Snort model and they also include the alarms not detected by the Snort model. This, however, inevitably requires sacrifice of the new false alarms those are not found in the Snort model. However, considering the fact that the new false alarms are not as frequent and that it is possible to detect new attacks that were previously undetected, the application of this model depends upon the circumstances of the system of the administrators.

5.3 Discussions

In the results of experiment, the reason for lower accuracy of C4.5 than XIBL is as following. C4.5 is a suitable algorithm where with a large data it makes out rules in a quickly with relatively correct accuracy. But each learning time it makes different rule trees. C4.5 is suitable for intrusion detection area in that intrusion detection needs on-line process but for the dynamics it is rarely used in intrusion detection. That is, in the rules cluster where the score of change is very small the accuracy is high, but in case of handling of packets C4.5 is not suitable for the dynamics. On the other hand, XIBL has a long training time, but in dynamic data it can make rules with stability. So, it has higher accuracy than C4.5.

If there are small number of packets, then the ability of learning rate is reduced so the performance is not good. The other way, if there is large number of packets, then the ability of learning rate will be increased and the performance is high. But the learning

time and memory size for saving knowledge are increased in proportion to performance. And also detection time is increased in proportion to size of knowledge.

The main limitation of combined system is that it depends on learning algorithm, so according to the learning algorithm, the detection accuracy is various. And it makes alarms when misused system does. Therefore it has no meaning without misused system.

6. CONCLUSIONS

In this paper, we have proposed signature-based and machine learning-based intrusion detection systems. The combined model of the intrusion detection system conducts self-classification in both the signature-based and the machine learning-based intrusion detection systems simultaneously. The machine learning-based intrusion system in the combined model corrects the accuracy of intrusion alarms, reduces false alarms, and detects types of intrusions that are not detected by the signature-based intrusion detection system. The verification of the usefulness of this combined model was done by using the DARPA Data Set. The results show that the combined model has various advantages compared to a single-system model. The main learning algorithm for the combined model was XIBL. C4.5 also was used for comparison but it produced worse results than did XIBL.

The objective of the combined model is to improve performance and in this paper it does not merely mean numerical increase in the overall accuracy. The errors in intrusion detection are classified into false positive and false negative and they have symmetrical structure with true negative and true positive, respectively. When false positive is reduced due to a certain factor, it also affects true positive. The reduction in false negative also affects true negative. Therefore, the ideal way of improving performance is to reduce false alarms at maximum while sacrificing true alarms at minimum. The application of these standards depends on the environment of an administrator who operates the system. We tried to explain the characteristics of the variables that determine the properties of the combined model system, and leave the determination of threshold to administrators.

Beta-pick aims at finding more abnormal packets. Moreover with Alpha-cut we will show that there are relatively fewer false alarms than single misused system. That is, our paper's main contribution is to show the usefulness of combined system between machine learning anomaly system and misused system to reduce false alarms in the experiment level considering that most IDS is misuse system in real field.

There are two ways to combine the signature-based and the machine learning-based intrusion detection technologies. One is a positive combination that creates a new algorithm by combining the advantages of each algorithm. The other is a passive combination where one system handles the results already handled by the other system. The latter is easy to implement; however the combination of these methods ultimately has limits in improving performance. It is necessary, therefore, to study a new intrusion detection system that adopts the advantages of both technologies. And it is interesting to provide more experiment data with respect to different attack benchmarks, for example attack packets generated by some program with adjustable parameters, so that we can see how the system fares to more dynamic attack patterns. In addition, it would be good to see a comparison of other major IDS system, not just Snort, Snort + C4.5, and Snort + XIBL.

REFERENCES

1. S. Patton, W. Yurcik, and D. Doss, "An Achilles' heel in signature-based IDS: squealing false positives in SNORT," in *Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection* (*RAID 2001*), 2001.
2. W. Lee, "A data mining framework for constructing features and models for intrusion detection systems," Ph.D. Dissertation, Department of Computer Science, Columbia University, New York, 1999.
3. I. Weon, D. Song, C. Lee, Y. Heo, and J. Jang, "A machine learning approach toward an environment-free network anomaly IDS – a primer report," in *Proceedings of the 5th International Conference on Advanced Communication Technology*, 2003, pp. 813-817.
4. C. Kruegel and T. Toth, "Using decision trees to improve signature-based intrusion detection," in *Proceedings of the 6th Symposium on Recent Advances in Intrusion Detection*, LNCS 2820, Springer-Verlag, 2003, pp. 173-191.
5. M. Mahoney and P. Chan, "PHAD: packet header anomaly detection for identifying hostile network traffic," Technical Report No. CS-2001-04, Department of Computer Sciences, Florida Institute of Technology, 2001.
6. R. Lippman *et al*., "Evaluation intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation," in *Proceedings of DARPA Information Survivability Conference and Exposition*, 2000, pp. 12-26.
7. K. Julisch, "Mining alarm clusters to improve alarm handling efficiency," in *Proceedings of the 17th Annual Computer Security Application Conference*, 2000, pp. 12-21.
8. 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*, 2002, pp. 366-375.
9. S. Christensen, M. Manganaris, D. Zerkle, and K. Hermiz, "A data mining analysis of RTID alarms," in *Proceedings of the 2nd Workshop on Recent Advances in Intrusion Detection*, 1999, pp. 571-577.
10. SNORT, http://www.snort.org, 2006.
11. D. Aha and D. Kibler, "Noise-tolerant instance-based learning algorithms," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, 1989, pp. 794-799.
12. J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln laboratory," *ACM Transactions on Information and System Security*, Vol. 3, 2000, pp. 262-294.
13. M. Roesch, "SNORT – lightweight intrusion detection for networks," in *Proceedings of the USENIX/LISA Conference*, 1999, pp. 229-238.
14. J. R. Quinlan, "Probabilistic decision trees," in Y. Kodratoff and R. Michalski (eds.), *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann Publishers, Inc., San Mateo, California, Vol. 3, 1990, pp. 140-152.
15. C. Stanfill and D. Waltz, "Toward memory-based reasoning," *Communications of the ACM*, 1986, pp. 1213-1228.
16. S. Cost and S. Salzberg, "A weighted nearest neighbor algorithm for learning with symbolic features," *Machine Learning*, Vol. 10, 1993, pp. 57-78.

17. D. Kim, "A study on evaluation model and network based IDS using IBL," M.S. Dissertation, Department of Computer Science, Konkuk University, 2003.
18. I. Won, D. Song, and C. Lee, "The architecture of network intrusion detection systems (written in Korean)," *Magazine of the Korean Institute of Communication Sciences*, Vol. 19, 2002, pp. 41-51.
19. NetDetector, http://www.niksun.com/documents/ND_20_Final_ version.pdf.
20. MatrixMonitor, http://www.cs-inc.com/CSI/pdf/matrixmonitor- datasheep.pdf.

Ill-Young Weon received his B.S. degree in Computer Science from Kyung Won University, Korea, in 1997 and the M.S. degree in Computer Science from Konkuk University, Korea, in 2000. He received his Ph.D. degree in Computer Science from Konkuk University, Korea, in 2006. His research interests are artificial intelligence, network security, and complexity theory.

Doo Heon Song received his B.S. degree in Calculus Statistics from Seoul National University, Korea, in 1981 and the M.S. degree in Computer Science from Korea Institute Science Technology University, Korea, in 2000. He received his Certificate Ph.D. from University California, U.S.A, in 2006. He was a researcher at Korea Institute Science Technology University, Korea, from 1983 to 1986. Currently he is a Professor at Yongin Songdam College, Korea. His research interests are data mining, machine learning, database, and security

Chang-Hoon Lee received his B.S. degree in Mathematics from Yonsei University, Korea, in 1975 and the M.S. degree in Computer Science from Korea Institute Science Technology University, Korea, in 1977. He received his Ph.D. degree from Korea Institute Science Technology University, Korea, in 1993. He was Head of Information Technology Center at Konkuk University, Korea, from 1996 to 2000. He was Rector of Information Technology Center at Konkuk University, Korea, from 2001 to 2002. Currently he is a Professor at Konkuk University, Korea. His research interests are intelligent system, operating system, security, and e-commerce