Multimed Tools Appl
DOI 10.1007/s11042-012-1275-x

# A study on cyber threat prediction based on intrusion detection event for APT attack detection

**Yong-Ho Kim · Won Hyung Park**



**Abstract** A number of APT(Advanced Persistent Threat) attack malwares are being detected as of late together with attempts by the state and enterprises to leak personal information. To detect and respond to them, malwares must first be detected by security monitoring system. In particular, availability of a method to detect and predict such malwares in advance will lead to preventing security incidents. This study will propose a method of prediction based on intrusion detection event and a functional configuration to realize the method and will assess the prediction model based on intrusion detection events proposed through a test consisting of the stages of learning, prediction and evaluation.

**Keywords** Cyber threat · Intrusion detection event · APT malware

## 1 Introduction

The cyber threat prediction technologies discussed and studied so far are mostly of predicting changes of numerical values after the unit time of time-series data used in a near future or for input by using numerical data collected on network or under individual system with a statistics-based prediction model. However, the numerical results can only be used as reference data and it is in fact almost impossible in reality to decide which handling measures must be taken against which types of threats in details. Therefore the numerical results cannot be of substantial assistance in strengthening security. The prediction technologies based on intrusion detection event are intended at predicting detailed information

Y.-H. Kim
Department of Information and Communication Engineering, SungKyunKwan University, 867-90 (202) Songcheon-dong, Kangbuk-gu, Seoul 142-816, Republic of Korea
e-mail: porsche0911@paran.com

W. H. Park (✉)
Department of Information Management, Far East University, Wangjang-ri, Gamgok-myeon, Eumseong-gun, Chungbuk 369-700, Republic of Korea
e-mail: whpark@kdu.ac.kr

Published online: 27 October 2012

relating to detailed event occurrence and intrusion in advance by overcoming the problems of statistical prediction model. Due to technological difficulties and limitations, only a theoretical model has been suggested with validity evaluation through a test. Since there are yet to be cases of using the technologies in detail, this study focuses on validating and evaluating effectiveness of the prediction model based on intrusion detection event.

## 2 Related work

In July 2011, NIST (National Institute of Science and Technology) released the "Attack Graph" to assist in predicting the risk of hackers to destroy computer system security and in identifying the most vulnerable resources [2]. NIST anticipates that the attack graph will enable IT administrators to identify vulnerabilities to be handled in order to protect precious data by analyzing and designating possibilities of all routes to be used by hackers in intruding computer system. Network attack graph provides information to security analyzers so that to understand how vulnerabilities of single network service can be the cause of overall vulnerabilities and how intruders can penetrate into system by stages. A variety of approaches have been used to create attack graph so far.

An earlier approach was proposed by Sheyner [6]. It was a model checking approach. In an attack graph, nodes indicate network state and the corners indicate that the state of an attacker's action is changing. When specific information about an attack is given. model checking technology is used to check whether it is an attack model of a system that satisfies the given characteristics.

Philips and Swiler [7] developed a search engine customized to attack graph creation. In general, this approach is subject to problems in terms of scalability when state increase occurs.

Ammann [1] proposed an algorithm based on graph search in order to create an attack graph. The attack graph was used in topological vulnerability analysis tools. This study assumed that an attacker's privilege always increased while analysis was conducted. It also explains that if an attacker obtains privilege of a polynomial number, the proposed algorithm can complete an attack graph creation within the time of a polynomial expression.

Xinming Ou [10] limited approaches to create a logical attack graph of which the logical dependence between attack target and configuring information can be directly explained. The logical attack graph holds a polynomial size in relation to the analyzed network. Another approach to creating an attack graph is to create an attack graph based on vulnerability attack using attack scenario configuration techniques. Ning [5] and Cuppens [3] tested preceding conditions and results of attacks. An attack scenario is configured by adjusting results of previous attacks considering preceding conditions of the next attack. Qin and Lee [4] proposed a warning correlation approach based on statistics in order to configure an attack scenario that recognizes a new warning relationship and is not depending on the previously acquired knowledge of an attack change pattern.

This paper adopted an approach to create an attack graph based on data mining technology for prediction of cyber threats. The concept of using data mining at intrusion detection was proposed in [9]. Thurimella [8] showed that the correlation among combinations of warnings within an intrusion detection log, a result of attackers and the acts of attack, could be verified using the data mining approach.

In this paper, correlation rules are identified from intrusion warning and the identified correlation rules are used to create an attack graph.

## 3 Prediction based on intrusion detection event

The existing time series prediction model was a frequency prediction of cyber threat items and, therefore, difficulties existed in using the prediction results. The prediction model based on intrusion detection event analytically expressed the prediction results in order to improve on the weakness of time series prediction model. As a ground for analytical expression of results according to the principle of five Ws and one H, intrusion detection events were used. In addition, using information extracted from intrusion detection events, the source area/target area of attack can be expressed as well as visual/geographic expression of threat.

The prediction model based on intrusion detection event shows possibility of threat prediction based on correlation of the intrusion detection events. When intrusion detection events are analyzed, it is found that attacks before or after a specific attack exist in a correlation. By extracting intrusion detection events that hold a context as such, an attack scenario is configured. When an intrusion detection event takes place, the next attack in an attack scenario can be predicted by analyzing at which stage of the attack scenario the intrusion detection event takes place. This will resultantly enable prediction of the final threat.

The prediction model based on intrusion detection even includes the functions to collect and pre-treat intrusion detection events, extract threads and sessions, create attack scenarios through correlation analysis, predict intrusions and express analytical results.

The function to collect and pre-treat intrusion detection events is to collect intrusion detection events and the related data, convert the collected data into a uniformed format, identify repetitive intrusion detection events and contract data by integrating the events into a single event. The function to extract attack threads and sessions is to extract events to attempt single-directional and bi-directional intrusions between source area and target area. The function to create attack scenarios through correlation analysis is to identify context of intrusion detection events using sequential association rules and therefore to create sequential rules and extract the time of event occurrence from the sequential rules. The function to predict intrusions is to predict intrusions by searching events on the sequential rules when intrusion detection event occur and by considering the context of events. The function to express analytical results is to express prediction results in GUI environment according to the principle of five Ws and one H.

Figure 1 shows the structure and operation flow of the proposed prediction system based on intrusion detection event.

### 3.1 Collection and storage of intrusion detection events

Intrusion detection event repository provides a function to collect events, the ground for expression of results according to the principle of five Ws and one H, and the related data. intrusion detection event repository collects the following data.

- Collecting intrusion detection event data

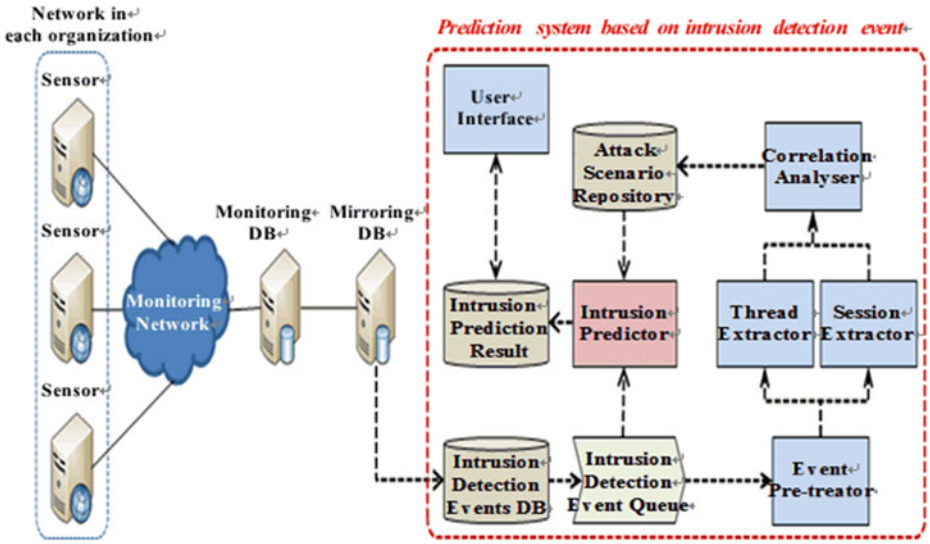


**Fig. 1** Structure and operation flow of prediction system based on intrusion detection event

- Use: To derive attack scenarios and use them in suggesting prediction response info.
- Required Data: Intrusion detection event info., intrusion detection rules info.

• Collecting status data of system and network in each organization

- Use: To identify intrusion detection events and decide weighted values for estimation of threats by organization
- Required Data: IP address band by organization, information on importance of organization

• Collecting harmful website and IP address data

- Use: To decide weighted values for estimation of threats
- Required Data: Information on harmful website and IP addresses used in enterprizes monitoring operation

• Collecting matrix event data

- Use: To decide weighted values for estimation of threats
- Required Data: Information on extensive monitoring events

The collected data are converted into a uniformed format so that to enable mutual reference of the data. In addition, 11 insignificant fields among the intrusion detection events are eliminated. The 12 fields to be used in prediction based on intrusion detection event are listed below.

- Time of occurrence
- Organization code
- IP/port of attacker and victim
- Name of attack

- CVE-ID
- Country code
- Continent code
- Regional code
- In/out field

### 3.2 Intrusion detection event queue

When an intrusion detection event occurs, intrusion detection event queue sends the intrusion detection event to intrusion predictor and pre-treator.

### 3.3 Pre-treatment

Pre-treator contracts and verifies intrusion detection events and also binds the closely related intrusion detection events.

A function to identify repeated intrusion detection events and to unify them into a single event in order to contract intrusion detection events produces the effect of reducing time and resources required in analysis. In addition, when events of which the source area/target area IP, attack name and time of event occurrence are the same take place sequentially in the order of time, this function regards the events as repeated events and integrate them into a single event.

Pre-treator eliminates events of which cyber threat prediction is difficult with intrusion detection events only. For example, when it cannot be identified whether an intrusion detection event is an intrusion detection event targeting a certain organization, analytical prediction is disabled and therefore the event must be eliminated.

### 3.4 Extracting attack thread

Attack thread extractor extracts unidirectional intrusion attempt events between source and target areas. A series of intrusion detection events taking place from a source area to a target area at a time interval were collected and defined as a single attack thread. In other words, attack thread refers to a series of processes through which a specific attacker attacks a specific system of a target organization. All of the collected intrusion detection events are categorized in the unit of attack thread. Since the start and end points of an attack are not known, the size of an attack thread must be decided as an optimal attack thread time window through a test.

### 3.5 Extracting attack session

Attack session extractor extracts bidirectional intrusion attempt events between source and target areas. From an attack thread, intrusion detection events of which the target area of attack is the source area of attack were collected and defined as a single attack session. In other words, an intrusion detection event of which the source area is A and the target area is B and an intrusion detection event of which the source area is B and the target area is A were extracted and combined as a single session. Attack threads are re-categorized in the unit of attack session. Even if an intrusion detection event of which the source area is A and the target area is B and an intrusion

detection event of which the source area is B and the target area is A occurred, it is difficult to conclude that an intrusion took place in B by an attack of A. However, a number of malwares including trojan characteristically attempt connection to other sites after intrusion. Considering a possibility of the site for an attempt of connection being an attacker's site, bi-directional intrusion attempt events between a source area and a target area were regarded as a single attack.

An attack session contains attack scenarios. A series of the acts of attack including information collection, intrusion attempt, intrusion and acts after an intrusion are included in an attack session. The order of intrusion detection event occurrence shows the order of attacks possible to occur for each attack. The order of intrusion detection event occurrence is used when creating sequential rules.

In case of an attack session, the start and end points of an attack are also unknown as of an attack thread. Therefore, for the size of an attack session, the optimal attack session time window must be decided through a test.

### 3.6 Analyzing correlation

Correlation analyzer creates sequential rules by identifying the context of intrusion detection events. To identify context of intrusion detection events, sequential association rule is used. Sequential association rule is a data mining algorithm that reflects the order of intrusion detection events in the existing continuous association rule mining algorithm (CARMA) and creates sequential rules by applying attack thread and attack session results to sequential association rule.

(1) Predicting Time of Intrusion Attempt Event

As statistical distribution characteristics of the differences in the time of occurrence between events, 90 % or more is 0 s in case of thread unit sequential rules and a distribution by max. 1 day or more (86,400 s) is displayed. In case of session unit sequential rules, a distribution by min. 0 s and max. 2–3 days is displayed. As shown in (Fig. 2), a standard deviation of time differences between

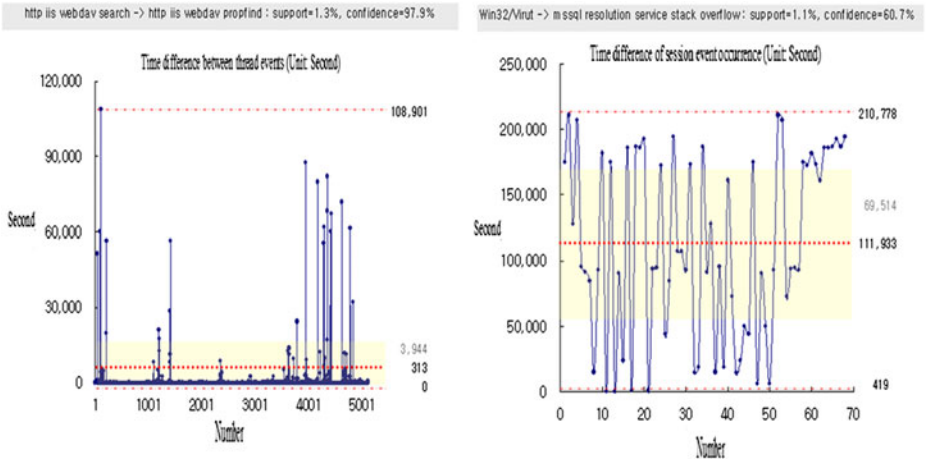


**Fig. 2** Statistical distribution of differences in the time of occurrence between events

events is large and therefore, it is difficult to predict the time of event occurrence using average values.

As for prediction of intrusion attempt event time, there is a method to predict an event occurrence within Tmax time based on the max. time (Tmax).

- $$T_{prediction} \leq T_{\max}$$

Another method is to predict an event occurrence with average time (Tavg) and to suggest the min. value (Tmin), max. value (Tmax) and standard deviation (Tstdev) at the same time.

- $$T_{prediction} = T_{avg}(T_{\min}, T_{\max}, T_{stdev})$$

Intrusion detection events occurring in government and public organizations are collected into the source DB of enterprizes every five minutes. Therefore, only the events of intrusion attempt of which time difference between the intrusion detection events is 5 min or longer can be predicted even if the time of prediction is ignored.

### 3.7 Predicting intrusion

Intrusion predictor searches events on an attack scenario when an intrusion detection event takes place and then predicts events to occur afterwards. When a single intrusion detection event occurs, one or more events can take place afterwards. To select events among a number of possible events to present as the results of prediction. the selection criteria are necessary.

(1) Solutions to Collision of Sequential Rules

Sequential rules extracted by using sequential association rule present prediction results according to the following criteria.

- Select an event of high confidence

– *confidence* ($E_n$)>*confidence* ($E_k$): Select $E_n$
– *confidence* ($E_n$)>*confidence* ($E_k$): Select $E_k$

- Select an event of high support

– *support*($E_n$)>*support*($E_k$) : Select $E_n$
– *support*($E_n$)<*support*($E_k$) : Select $E_k$

- Select an event of high priority: Priority is decided according to frequency of repetitive detection of sequential rules.

– *priority* ($A \rightarrow E_n$)>*priority* ($A \rightarrow E_k$): Select $E_n$
– *priority* ($A \rightarrow E_n$)<*priority* ($A \rightarrow E_k$): Select $E_k$

If it is impossible to select a single event under the conditions above, both $E_n$ (intrusion detection event $n$) and $E_k$ (intrusion detection event $k$) are presented as prediction results. However, it must be implemented so that application status of the conditions above, critical values of comparison and application priorities can be changed by user.

(2) Allocating Priorities to Sequential Rules

Allocation of priorities according to frequency of repetitive detection of sequential rules is carried out as of the following.

$P_{init}$: Initial priority (> 0)

$LP$: Learning period

$P_{dec}$:Amount of priority decrease in case sequential rule is not detected $P_{inc}$: Amount of priority increase in case sequential rule is repetitively detected

$P(A \rightarrow E_n)$: Priority of sequential rules through which $E_n$ event occurs after $A$ event

Initial sequential rule creation: $P(A \rightarrow E_n) = P_{init}$

The same sequential rule not detected at learning: $P(A \rightarrow E_n) = P(A \rightarrow E_n) - P_{dec}$

The same sequential rule detected at learning: $P(A \rightarrow E_n) = P(A \rightarrow E_n) + P_{inc}$

If $P(A \rightarrow E_n) = 0$: Deleting the sequential rule concerned $P(A \rightarrow E_n)$

※ Priority increases when the same sequential rule is detected. Priority decreases when the same sequential rule is not detected

## 4 Test of prediction based on intrusion detection event

### 4.1 Test data

The intrusion attempt prediction performance was tested by dividing intrusion detection event data into the categories of learning, prediction and verification. Learning was carried out for 3 days (Apr. 25, 2011 - Apr. 27, 2011) using intrusion detection events and prediction was conducted using intrusion detection events for 1 day following the learning period (Apr. 28, 2011). Then, verification was carried out using intrusion detection events for 4 days after learning period (Apr. 28, 2011 - May 1, 2011) including the day of prediction (Apr. 28, 2011). As a result of eliminating repetitive events from all intrusion detection events, an average elimination rate of 83 % was found. In addition, for 47 % of the intrusion detection events, it was not possible to recognize which organizations were the source and target areas.

Table 1 lists the number of events and number of repetitive events of data used in the test of prediction based on intrusion detection event as well as the rate of inability to recognize organizations in which the events occurred and uses of the events.

### 4.2 Test procedures and methods

A test of prediction based on intrusion detection event consists of the three stages of learning, prediction and verification. Procedures of each stage are as outlined below.

**Table 1** Intrusion detection data for intrusion attempt prediction performance verification

| Date | Number of intrusion detection events | Number of events after eliminating repetition (elimination rate %) | Inability to recognize organizations in which events occurred | Remarks | |
|---|---|---|---|---|---|
| Apr. 25, 2011 | 3,804,936 | 588,405(84 %) | 45 % | Learning | |
| Apr. 26, 2011 | 1,667,822 | 284,234(82 %) | 41 % | | |
| Apr. 27, 2011 | 2,129,665 | 403,359(81 %) | 41 % | | |
| Apr. 28, 2011 | 2,017,170 | 421,489(79 %) | 43 % | Prediction | Verification |
| Apr. 29, 2011 | 2,587,751 | 488,549(81 %) | 49 % | – | |
| Apr. 30, 2011 | 4,554,718 | 760,872(83 %) | 56 % | – | |
| May 1, 2011 | 3,243,031 | 540,094(83 %) | 53 % | – | |

(1) Learning Procedures

- Collecting and pre-treating (eliminating repetition) learning data (Apr. 25, 2011 - Apr. 27, 2011)
- Extracting attack threads (collection of uni-directional events)
- Extracting attack sessions (collection of bi-directional events)
- Learning sequential rules in the unit of attack thread
- Calculating time difference distribution in relation to sequential rules in the unit of attack thread
- Learning sequential rules in the unit of attack session
- Calculating time difference distribution in relation to sequential rules in the unit of attack session

(2) Prediction Procedures

- Collecting and pre-treating (eliminating repetition) prediction data (Apr. 28, 2011)
- Administering prediction by event (prediction based on max. time)

(3) Evaluation Procedures

- Collecting and pre-treating (eliminating repetition) evaluation data (Apr. 28, 2011 - May 1, 2011)
- Evaluating if the corresponding events occur within the predicted time in relation to each prediction case

### 4.3 Test results

Table 2 summarizes results of a test carried out in the previously described methods using event data for prediction based on intrusion detection event.

Table 3 shows sequential rules based on attack thread. As for rule_id 55, when intrusion detection events take place by 187 times (antecedent), the reliability of intrusion detection events occurring by 197 times (consequent) is 100 % and the frequency of occurrence is 0.005. Although the frequency is low, it can be interpreted that, when intrusion detection events occur by 187 times on the data learned, intrusion detection events will definitely occur by 197 times. In other words, when 187 intrusion detection events take place, it can be predicted that 198 intrusion detection events will occur. However, when 187 intrusion detection events occur, the average time for occurrence of 198 intrusion detection events (timediff_avg) is 2 min and 7 s

**Table 2** Procedures and results of learning using intrusion detection events

| Order | Details | Time required | Results | Remarks |
|---|---|---|---|---|
| 1 | Collecting and pre-treating learning data | 601 s | No. of events for learning: 1,275,998 | Rate of elimination: 83 % |
| 2 | Extracting attack threads | 10 s | No. of threads detected: 419,333 | 3.04 (events/thread) |
| 3 | Extracting attack sessions | 4 s | No. of sessions detected: 6,009 | 25.38 (events/session) |
| 4 | Learning sequential rules in the unit of attack thread | 95 s | No. of sequential rules detected: 547 | min_support: 0.001<br>min_confidence: 0.001 |
| 5 | Calculating time difference distribution in relation to sequential rules in the unit of attack thread | 2 h | Distribution of time difference between events by sequential rule | |
| 6 | Learning sequential rules in the unit of attack session | 3 s | No. of sequential rules detected: 488 | min_support: 0.001<br>min_confidence: 0.001 |
| 7 | Calculating time difference distribution in relation to sequential rules in the unit of attack session | 6 h | Distribution of time difference between events by sequential rule | |

(162 s). This indicates that a time difference of 5 min or less is not significant in an environment where the period of collecting intrusion detection events is 5 min.

Table 4 shows sequential rules based on attack session. The sequential rules show similar characteristics to (Table 3). However attack session extractor finds sequential rules from a log to extract bi-directional intrusion attempt events between a source area and a target area.

### 4.4 Issues in prediction based on intrusion detection event

(1) Time required in prediction and verification of intrusion detection event

The results of a test on the recently collected intrusion detections (for 7 days from Apr. 25, 2011 to May 1, 2011) indicated that the daily average count of intrusion detection event occurrence was 498,143 (after eliminating repetitive events) and that the time required in learning, prediction and verification of events occurring in 1 day was 2.73 h, 27.8 h and 168 h (7 days) respectively.

Therefore, a process to minimize test data through pre-treatment, such as to eliminate unnecessary intrusion detection events or those with low relevance, is necessary. In addition, for pre-treatment of the data, information about the environment of monitoring network, such as information about network system and service of each organization as well as the existing vulnerabilities, is required. Moreover, it is necessary to extract successful attack events by time unit, attack type and organization and therefore to distinguish them from all intrusion detection events.

(2) Validity of prediction due to narrow gap in occurrence of intrusion detection events

The test results indicated that the time difference in occurrence of intrusion detection events verified of their correlation was mostly within several seconds. Intrusion detection events occurring in government and public organizations are collected into the enterprizes monitoring system DB every 5 min. Therefore, prediction is disabled if the time difference in occurrence of intrusion detection events is 5 min or less. Since response to the predicted events is mostly impossible, the usefulness of prediction is lowered.

**Table 3** Sequential rules based on attack thread

| rule_id | antecedent | consequent | support | confidence | timediff_cnt | timediff_min | timediff_max | timedifff_avg | timedifff_dev |
|---|---|---|---|---|---|---|---|---|---|
| 55 | 187 | 197 | 0.005 | 100 | 24 | 0 | 2580 | 162 | 519.2821 |
| 56 | 149 | 149 | 0.004 | 100 | 2811 | 46 | 65519 | 153 | 1781.08774 |
| 57 | 109 | 109 | 0.001 | 100 | 26 | 1 | 25741 | 2469 | 5982.966 |
| 58 | 35 | 35 | 0.002 | 100 | 32 | 59 | 52442 | 4933 | 10115.877 |
| 59 | 170 | 164 | 0.001 | 100 | 60 | 0 | 962 | 143 | 208.053238 |
| 60 | 457 | 455 | 0.007 | 96.667 | 674 | 0 | 1739 | 4 | 75.95362 |
| 61 | 188 | 197 | 0.006 | 95.833 | 25 | 0 | 2580 | 163 | 507.2031 |
| 62 | 457 | 457 | 0.007 | 93.333 | 329 | 1 | 152279 | 3547 | 13612.5889 |
| 63 | 436 | 436 | 0.139 | 88.832 | 10361 | 58 | 234120 | 5101 | 10184.1885 |
| 64 | 249 | 249 | 0.002 | 87.5 | 79 | 58 | 15420 | 1266 | 2771.24658 |
| 65 | 237 | 238 | 0.009 | 86.842 | 89 | 0 | 90360 | 1624 | 9983.627 |
| 66 | 473 | 473 | 0.002 | 85.714 | 30 | 60 | 15420 | 2766 | 4110.52441 |
| 67 | 40 | 40 | 0.002 | 85.714 | 12 | 59 | 3121 | 354 | 874.919 |
| 68 | 625 | 625 | 0.01 | 85 | 4366 | 58 | 156240 | 434 | 5068.8623 |
| 69 | 239 | 238 | 0.023 | 83.505 | 333 | 0 | 153479 | 820 | 8796.692 |
| 70 | 361 | 361 | 0.004 | 83.333 | 1149 | 58 | 50639 | 1161 | 2146.67 |
| 71 | 170 | 170 | 0.001 | 83.333 | 66 | 58 | 1860 | 461 | 535.187256 |
| 72 | 234 | 238 | 0.007 | 82.143 | 36 | 0 | 3240 | 135 | 563.854858 |
| 73 | 430 | 430 | 0.006 | 81.481 | 12266 | 1 | 70559 | 149 | 715.376648 |
| 74 | 95 | 95 | 0.005 | 80.952 | 131 | 58 | 31440 | 683 | 3311.87671 |
| 75 | 233 | 238 | 0.004 | 80 | 29 | 0 | 0 | 0 | 0 |
| 76 | 76 | 334 | 0.0002 | 80 | 18 | 0 | 0 | 0 | 0 |
| 77 | 330 | 330 | 0.01 | 78.751 | 68 | 58 | 221520 | 5759 | 28072.9785 |
| 78 | 551 | 551 | 0.124 | 76.686 | 2685 | 58 | 185221 | 4403 | 18665.4531 |

**Table 4** Sequential rules based on attack session

| rule_id | antecedent | consequent | support | confidence | timediff_cnt | timediff_min | timediff_max | timediff_avg | timediff_dev |
|---|---|---|---|---|---|---|---|---|---|
| 977 | 30 | 30 | 0.033 | 100 | 2 | 0 | 0 | 0 | 0 |
| 978 | 169 | 113 | 0.033 | 100 | 2 | 779 | 779 | 779 | 0 |
| 979 | 454 | 454 | 0.033 | 100 | 2 | 0 | 0 | 0 | 0 |
| 980 | 534 | 455 | 0.033 | 100 | 2 | 150481 | 150481 | 150481 | 0 |
| 981 | 328 | 328 | 0.033 | 100 | 2 | 180 | 180 | 180 | 0 |
| 982 | 433 | 328 | 0.033 | 100 | 2 | 1080 | 1080 | 1080 | 0 |
| 983 | 457 | 457 | 0.366 | 100 | 22 | 59 | 219600 | 66703 | 93654.1953 |
| 984 | 36 | 36 | 0.266 | 100 | 16 | 0 | 200101 | 106447 | 83675.33 |
| 985 | 168 | 168 | 0.033 | 100 | 2 | 539 | 539 | 539 | 0 |
| 986 | 144 | 64 | 0.033 | 100 | 2 | 19319 | 19319 | 19319 | 0 |
| 987 | 297 | 297 | 0.033 | 100 | 2 | 3599 | 3599 | 3599 | 0 |
| 988 | 678 | 678 | 0.133 | 100 | 8 | 0 | 106980 | 58875 | 44490.22 |
| 989 | 89 | 293 | 0.067 | 100 | 4 | 0 | 780 | 390 | 450.333221 |
| 990 | 5 | 5 | 0.033 | 100 | 2 | 0 | 0 | 0 | 0 |
| 991 | 91 | 85 | 0.033 | 100 | 2 | 961 | 961 | 961 | 0 |
| 992 | 91 | 303 | 0.033 | 100 | 2 | 961 | 961 | 961 | 0 |
| 993 | 168 | 113 | 0.033 | 100 | 2 | 719 | 719 | 719 | 0 |
| 994 | 11 | 11 | 0.067 | 100 | 4 | 661 | 16920 | 8790 | 9387.138 |
| 995 | 14 | 14 | 0.233 | 100 | 14 | 0 | 8880 | 1268 | 3224.65234 |
| 996 | 355 | 452 | 0.033 | 100 | 2 | 18839 | 18839 | 18839 | 0 |
| 997 | 331 | 548 | 0.033 | 100 | 2 | 95580 | 95580 | 95580 | 0 |
| 998 | 481 | 481 | 0.067 | 100 | 4 | 0 | 60 | 30 | 34.6410179 |
| 999 | 17 | 17 | 0.2 | 100 | 12 | 0 | 2761 | 820 | 1139.27893 |

(3) Intrusion detection data and intrusion detection rules

Intrusion detection rules applied to TAS are frequently added, modified and deleted. Even time intrusion detection rules are changed, the sequential rules must be learned and the rules must be applied to an independent prediction model based on intrusion detection event. This process must be carried out while an independent prediction system based on intrusion detection event is operated up to the present. For this, employees in full charge of the system must be appointed.

Detection rules inserted by administrator are mostly created using specific character strings generated after an attack succeeds in the stage of attack. In other words, intrusion detection events relating to an attack occur only when the attack succeeds and no intrusion detection events are generated in relation to an attempt of an attack before succeeding in the attack. It is difficult to analyze the correlation of events before and after an attack using a log of intrusion detection events as such. In fact, out of 2878 TAS detection rules, the number of TAS rules provided by information security service companies is 1459, the number of intrusion detection rules inserted by administrator is 1419 and the percentage of detection rules occurring after a successful attack is approx. 80 %.

(4) Stability and accuracy of intrusion detection system

TAS used in monitoring system does not guarantee stability and accuracy to detect the continuously changing methods and technologies of attack by type. TAS is an intrusion detection system for detection of misuses and it only detects attacks for which the detection rules exist. In other words, TAS cannot make predictions for attacks it does not detect.

## 5 Conclusion

The prediction model based on intrusion detection event analytically expressed the results of prediction in order to improve on the weakness of a time-series prediction model. In addition, this prediction model shows a possibility of threat prediction by analyzing correlation of intrusion detection events.

As a result of a test on an independent prediction model based on intrusion detection event, not only the problems of time required in prediction and verification of intrusion detection events and of prediction effectiveness due to a narrow gap in the occurrence of intrusion detection events, but also the fundamentally intrinsic problems of cyber threat prediction, such as problems concerning intrusion detection data and intrusion detection rules, stability and accuracy of intrusion detection system and also requirements of the automated prediction and handling systems in terms of monitoring have been raised. The problems are difficult to solve under the current national environments unlike the problems raised under ideal environments proposed by the related studies. It is considered that prediction based on intrusion detection event is possible only in environments where technical problems for prediction do not exist, usable data are available and the operating system is completely equipped. The results of this study will be sufficiently useful once the three requirements listed above for prediction based on intrusion detection event are satisfied in the future.

## References

1. Ammann DWP and Kaushik S (2002) Scalable, graphbased network vulnerability analysis. In: Proceedings of 9th ACM Conference on Computer and Communications Security, Washington, DC, November 2002
2. 'Attack graphs' predict computer security, http://www.eetimes.com/showArticle.jhtml?articleID=209601075

3. Cuppens F and Miege A (2002) Alert correlation in a cooperative intrusion detection framework. Proceedings 2002 IEEE Symposium on Security and Privacy, page 202, Berkeley, CA, USA, IEEE Comput. Soc.
4. Lee W, Qin X (2003) Statistical causality analysis of infosec alert data. In: Proceedings of the International Symposium on the Recent Advances in Intrusion Detection (RAID 2003), pages 73–94. Springer-Verlag, 2003
5. Ning P, Cui Y, Reeves DS, Xu D (2004) Techniques and tools for analyzing intrusion alerts. ACM Trans Inf Syst Secur 7(2):274
6. Oleg Sheyner SJRL, Haines J, Wing JM (2002) Automated generation and analysis of attack graphs. In: Proceedings of the 2002 IEEE Symposium on Security and Privacy (SP 2002), pages 273–284, 2002
7. Phillips C, Swiler LP A graph-based system for network-vulnerability analysis. In: Proceedings of the 1998 workshop on New security paradigms. ACM Press
8. Thurimella JJT, Ramakrishna (2006) A framework for the application of association rule mining in large intrusion detection infrastructure. In: Proceedings of the International Symposium on the Recent Advances in Intrusion Detection (RAID 2006), pages 1–18, 2006
9. Wenke Lee SS (1998) Data mining approaches for intrusion detection. In: Proceedings of the 7th USENIX Security Symposium, pages 79–94, 1998
10. Xinming Ou WFB and McQueen MA (2006) A scalable approach to attack graph generation. In: ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA

**Yong-Ho Kim** received his Ph.D. degree in Department of Information Security from the Kyonggi University, South Korea, in 2008. Now, hi is adjust professor in Department of Information and Communication Engineering, SungKyunKwan University, Republic of Korea. Dr. Kim research interests Digital Forensics and Network Forensics.

**Won Hyung Park** received his Ph.D. degree in Department of Information Security from the Kyonggi University, South Korea, in 2009. Now, hi is Professor in Department of Information Management, Far East University, South Korea. he co-authored more than 40 technical papers in the area of informaion security. Also, hi has been reviewer for International Journal(Computer-Journal) of Oxford Univ. press and IEEE International Conference(ICISA 2012, ICISA 2011).