We present an automated verification methodology of the security events, that is, IDS alerts, based on heuristic analysis. The proposed verification methodology aims to automatically identify real cyberattacks from the security events and filter out false positive, so that the security analyst is able to conduct security monitoring and response more effectively. For the proposed verification methodology, we used the 1,528,730,667 security events that were obtained from Science and Technology Security Center (S&T-SEC). We then extracted the core security events that were caused by the real cyberattacks. Among the core security events, we selected the top 20 types of the security events in the number of the real attacks that they raised. By analyzing the top 20 types of the security events, we discovered essential elements and optional elements for using in the automated verification of the security events. The evaluation results showed that the proposed verification methodology could contribute to the reduction (about 67%) of the meaningless security events. Furthermore, we demonstrated that the proposed verification methodology contributed to the detection of 140 true negatives that were not identified by the security analyst and the total accuracy of the proposed verification methodology was 96.1%.

1. Introduction

Intrusion Detection System (IDS) is one of the most powerful security appliances to respond to cyber threats on the Internet [1]. The main purpose of the IDS is to detect cyberattacks using predefined signatures that were made by security experts. Although IDS has been an important role to improve the security level of organizations, it has two main drawbacks. Firstly, it is unable to detect unforeseen cyberattacks, that is, 0-day attack, whose signatures were not made by the security experts. Secondly, the IDS raises not only a tremendous amount of alerts, but also most of them are false positives, i.e., misdetected attacks. Because of this, security analyst who carries out the security monitoring and response suffers from analyzing all the IDS alerts and finding out meaningful alerts related to real cyberattacks from them.

In order to cope with the problems, there have been many efforts on effectively removing meaningless IDS alerts [2–7], extracting attack scenarios [8–11], identifying unknown attacks [12], and visualizing IDS alerts [13, 14]. Although these approaches can contribute to improving performance of the IDS and helping security analyst for efficient analysis of IDS alerts, there is a limitation in that they do not focus on identifying whether the IDS alerts are caused by real cyberattacks or not. Because of this, the security analyst has to carry out additional analysis for the IDS alerts in order to identify real cyberattacks.

In this paper, we present an automated verification methodology of the security events, that is, IDS alerts, based on heuristic analysis. This research is an expansion of our pervious approach [15]. The proposed verification methodology aims to automatically identify real cyberattacks from
the security events and filter out false positive, so that the
security analyst is able to conduct security monitoring and
response more effectively. For the proposed verification
methodology, we used the 1,528,730,667 security events that
were obtained from Science and Technology Security Center
(S&T-SEC) [16]. It is providing security monitoring and
response service for the Korean government research insti-
tutes and has deployed Threat Management System (TMS)
into the boundary network of each organization. TMS is
similar to the IDS. The S&T-SEC collects all of the security
events, that is, TMS events, obtained from TMSes of Korean
government research institutes and security analyst analyzes
them to identify which security events are associated with real
attacks or false positives.

Among the 1,528,730,667 security events, we extracted the
core security events that were caused by the real cyberattacks.
Among the core security events, we selected the top 20 types
of the security events in the number of the real attacks that
they raised. By analyzing the top 20 types of the security
events, we discovered essential elements and optional ele-
ments for using in the automated verification of the security
events.

In order to evaluate effectiveness of the proposed verifica-
tion methodology, we used 1,528,730,667 security events. The
evaluation results showed that the number of security events
which should be analyzed by the security analyst was reduced by
33% of the entire security events if we apply the pro-
posed verification methodology to only the top 20 security
events. This means that the proposed verification method-
ology contributes to improving the rapidity of the security
monitoring and response. Also, we recognized that the
proposed verification methodology contributed to detection of
140 true negatives that were not identified by the security
analyst. Finally, the total accuracy of the proposed verification
methodology was 96.1% with respect to the top 20 types of the
security events.

The rest of this paper is organized as follows. In Section 2,
we give a brief description for the existing approaches based
on IDS alerts. In Section 3, we describe the proposed verifica-
tion methodology in detail. In Section 4, we provide experi-
mental results of the proposed verification methodology.
Finally, we present concluding remarks and suggestions for
future work in Section 5.

2. Related Work

Many approaches have been proposed for managing and
analyzing IDS alerts effectively. They can be categorized into
four groups by their purpose. The first group is that they focus
on the reduction of IDS alerts that are false positives [2–7].
In order to reduce the meaningless IDS alerts, they mainly
use machine learning and data mining techniques, especially
clustering algorithms such as K-NN, based on comparison
of similarity among IDS alerts. For example, Giacinto et al.
proposed a clustering method of IDS alerts so as to make uni-
fied description of attacks and attain a high-level description
of cyber threats. Treinen and Thurimella made meta-alarms
to identify known attack patterns in IDS alerts and used a data
mining technique, that is, association rule. The main purpose
of the second group is to build attack scenarios from IDS
alerts and to find out real cyberattacks that were matched to
the predefined attack scenarios [8–11]. Thus, there is a limita-
tion in that they only deal with the previously known
cyberattacks. The third group focuses on the detection of
unknown attacks from IDS alerts [12]. To this end, it extracts
seven statistical features from IDS alerts and finds out 0-
day attacks whose statistical values are different from the
normal IDS alerts. The last group is based on visualization of
IDS alerts to intuitively detect remarkable cyber threats from
usual IDS alerts [13, 14].

3. Proposed Methodology

3.1. Overall Architecture and Procedure. Figure 1 shows the
overall architecture and the procedure of the proposed
methodology for automated verification of security events
that were detected by TMS. The proposed methodology is
composed of three main phases: selecting data types, feature
extraction, and automated verification process. During the
phase of selecting data types, it first collects security events
recorded by TMS and classifies them into two types. One is
the type of core security events related to real cyberattacks
while the other type is false positives that are not related to
real cyberattacks. By investigating the core security events, it
extracts 7 data types that are used for the second phase. See
Section 3.2 for more detail.

The feature extraction phase starts with classifying top 20
security events from the core security events. It then extracts
features for using the automated verification with respect to
each type of top 20 security events. Since TMS records the
security events using two types of detection mechanism (i.e.,
signature and threshold based detection mechanisms), the
features are also divided into two types. See Section 3.3 for
more detail.

Finally, the automated verification process makes the verifica-
tion algorithm using the features that were extracted
from the second extraction phase. It then categorizes top 20
security events into 4 groups automatically. See Section 3.4
for more detail.

3.2. Selecting Data Types. Figure 2 shows the overall proce-
dure of the first phase, that is, the selecting data types, which
consists of three main steps: Collection, Analysis, and Extrac-
tion. The main purpose of this phase is to determine data
types that are useful for the verification of the security events.
For this phase, we used the security events of 7 years and
3 months (March 10th, 2006–May 31st, 2013) that were
detected by TMSs of S&T-SEC. Among the security events,
we observed that there were 9,036 real attacks and collected
the core security events that were related to them. Also, the
core security events have 799 types of the security events.
Note that we sanitized the name of the security events due
to the security problem. We analyzed the core security events
based on the know-how of the analyst in S&T-SEC, the past
accident history, related papers, and cooperation with the
related external security institutes. As a result, we extracted 7
significant data types for using the verification of the security events as follows.

(i) **SRC/DST IP.** Source and destination IP addresses that were used for the session.

(ii) **SRC/DST PORT.** Source and destination port numbers that were used for the session.

(iii) **NETWORK.** Type of networked systems such as WEB server, SMTP server, and DNS server.

(iv) **PROTOCLOL.** Protocol used for the communication such as TCP, UDP, and ICMP.

(v) **Number of SECURITY EVENTS.** Number of the security events with the same source IP address.

(vi) **PACKET SIZE.** Size of the packet attached in the security event.

(vii) **PAYLOAD.** Payload of the packet attached in the security event.

3.3. Feature Extraction

3.3.1. Overall Procedure. Figure 3 shows the overall procedure of the second phase, that is, the feature extraction, which consists of three main steps: Classification, Extraction, and Verification. The main purpose of this phase is to extract features that are used for the verification of the security events. For this phase, we used the security events of 8 months (May 1st, 2012–December 31st, 2012) and 7 months (January 1st, 2013–July 31st, 2013) that were detected by TMSs of S&T-SEC.
Table 1: Top 20 types of the security events and their statistics.

| Rank | Event name         | Number of security events | Number of real attacks | Type       |
|------|--------------------|---------------------------|------------------------|------------|
| 1    | D**5               | 507                       | 170                    |            |
| 2    | F****w             | 142                       | 98                     |            |
| 3    | Su**** us-net****504 | 1101                      | 55                     |            |
| 4    | R***ng             | 999                       | 47                     |            |
| 5    | In**-co***ct       | 324,247                   | 63                     |            |
| 6    | G****1-Sub***e     | 6,740                     | 26                     | Signature  |
| 7    | cn(g***t)          | 41,153                    | 102                    |            |
| 8    | Do***bot           | 212                       | 58                     |            |
| 9    | H**me              | 12,244                    | 78                     |            |
| 10   | Su**** us-Net****501| 1,101                     | 19                     |            |
| 11   | Net*** 2010(th)    | 11,422                    | 53                     |            |
| 12   | tcp syn flooding   | 9,474,044                 | 15                     | Threshold  |
| 13   | Mybi**** co***ct   | 65                        | 17                     | Signature  |
| 14   | Rat(****.dll)      | 726                       | 28                     | Signature  |
| 15   | udp-fragmented     | 12,009,382                | 8                      | Threshold  |
| 16   | Da****et,5,3,1     | 97                        | 37                     | Signature  |
| 17   | udp-same ip        | 870,139,035               | 7                      | Threshold  |
| 18   | tcp service scan   | 125,037,129               | 7                      |            |
| 19   | RAT(G**** t3.75)   | 2,498                     | 24                     | Signature  |
| 20   | Ho***i or L***K   | 613,286                   | 48                     |            |
| Total|                   | 1,017,676,130             | 960                    |            |

The total number of the security events was 1,528,730,667. Among the entire security events, we selected the top 20 types of the security events in the number of real attacks that they raised as shown in Table 1. Note that we sanitized the name of the security events due to the security problem. The total number of the real attacks caused by the top 20 types of the security events was 960. Also, 923 real attacks were related to the 16 types of the security events detected by signature based detection mechanism while the rest of real attacks were related to the 4 types of the security events detected by threshold based detection mechanism. We then analyzed all of the security events related to the top 20 security events based on the 7 data types extracted in the first phase. From the analysis, we extracted three types of features (i.e., essential elements, optional elements, and parameters) with respect to each type of the top 20 security events. Finally, we verified the security events using the three types of features, so that we are able to divide them into four groups: true positive group (i.e., real attacks), false positive group, unverified group, and verified group. In case of unverified and verified group, security analyst needs to carry out additional analysis for the security events in order to determine whether they are related to the real attacks or not, while the security events that belong to the rest of two groups do not require additional analysis by the security analyst.

3.3.2. Signature Based Security Events. In case of the 16 types of the signature based security events, we focused on only the
security events which are directly related to the real attacks and analyzed them based on the 7 data types of the first phase in Section 3.2. During the analysis, we checked which data types are useful for determining the real attacks with respect to each type of the signature based security events. As a result, we observed that only 2 data types, “SRC/DST IP” and “PAYLOAD,” are the most significant while the 5 data types (i.e., “SRC/DST PORT,” “NETWORK,” “PROTOCOL,” “NUMBER of SECURITY EVENTS,” and “PACKET SIZE”) do not contribute to determination of real attacks almost. In addition, in case of the data type of “PAYLOAD,” we extracted common strings that are included in the security events of the real attacks and recognized that there are two types of common strings. The first type is that they appeared in all the security events of the real attacks. The second type is that they appeared in the subset of the security events of the real attacks. Because of this, we divided the common string into essential elements (i.e., the first type) and optional elements (i.e., the second type). Also, the essential and optional elements have their parameters as unixed strings or values. In case of the data type of “SRC/DST IP,” it represents only source and destination IP addresses of the security events. Thus, they do not have common strings (including parameters) like the data type of “PAYLOAD.” So, we only checked whether the location (i.e., inside or outside of the organization) of the IP addresses in the security events is important or not for determining real attacks. With respect to their significance, they are also divided into essential elements or optional elements. Table 2 shows an example of features that were extracted from the 507 security events of D∗∗5.

### Table 2: Example of features of D∗∗F security event.

| Item       | Element                                | Meaning                          | Parameter        | Ratio | Data type       |
|------------|----------------------------------------|-----------------------------------|------------------|-------|-----------------|
| Source IP  | IP address of target (internal) institute | N/A                               | SRC/DST IP       | 100%  |
| Destination IP | IP address of external hosts            | N/A                               | asp?             | 100%  |
| asp        | String                                 | 32-digit hexadecimal              | 98%              |
| MAC        | Media access control address           | 12-digit hexadecimal with “.” mark | 1%               |
|            |                                        | 12-digit hexadecimal without “.” mark | 1%               |
| Essential elements |                                |                                   |                  |
| OS         | Operating system                       | HTTP/1.1                          | 90%              |
|            |                                        | win7                              | 2%               |
|            |                                        | Windows%20Xp                       | 2%               |
|            |                                        | Windows%207                        | 2%               |
|            |                                        | winxp%20 Professional              | 2%               |
|            |                                        | unknow                            | 2%               |
|            |                                        | unknow                            | 77%              |
| avs        | Anti-virus vaccine                     | (⋯⋯)                             | 20%              |
|            |                                        | (V3)                              | 3%               |
|            |                                        | winxp% Professional               | 57%              |
|            |                                        | Windows%207                       | 40%              |
|            |                                        | unknow                            | 1%               |
|            |                                        | III0star                          | 1%               |
|            |                                        | l05war                            | 1%               |
|            |                                        | Google.page                       | 88%              |
|            |                                        | MyApp                             | 4%               |
|            |                                        | Mozilla/3.0                       | 4%               |
|            |                                        | Mozilla/4.0                       | 2%               |
|            |                                        | Mozilla/5.0                       | 2%               |
|            |                                        | IP Address                        | 48%              |
|            |                                        | Domain name                       | 52%              |
| Cache-control | String                           | no-cache                          | 100%             |

#### 3.3.3. Threshold Based Security Events.
In case of the 4 types of the threshold based security events, the corresponding packets do not have payload data in most cases. Even though they contain some data, it represents a form of garbage, that is, meaningless strings like “AAAAA.” Because the main purpose of the cyber threats related to the threshold based security events is to send a lot of packets to the target hosts or networks in a short time (e.g., one second), the victims cannot provide their service or work normally anymore. Thus, it is not able to extract meaningful strings (i.e., essential and optional elements) from payload data of the security events. Instead of the common strings, therefore, we adopted the
characteristics of the threshold based security events for verifying them in an automated manner.

Due to the detection mechanism of the threshold based security events, they have the following two main characteristics. Firstly, there are lots of the security events that were caused by the same source IP address if they are caused by a real attack, because attackers have to send a tremendous amount of packets to the victim(s) in order to succeed in their attack. In this situation, it is obvious that the number of the security events caused by the specific source IP address is very large. Therefore, we use the number of the threshold based security events as an essential element for verifying them automatically. See Section 3.4 for more detail.

Secondly, the location of the destination IP addresses in the security events is distributed on the wide range. Because attackers tend to send their attack packets to the victims at random when they aim to search and probe their targets, we use the destination IP addresses of the threshold based security events as the second essential element for automated verification for them. See Section 3.4 for more detail.

3.4. Automated Verification Process. Figure 4 shows the overall verification process of the signature based security events and the threshold based security events. In case of the former, the verification process is as follows.

(i) Classification. In general, new types of the security events are provided by the security vendors periodically. Thus, it is unable to verify all types of the security events in practice. This classification step aims to check whether the security event fed into the verification process is verifiable or not. If the corresponding security event can be verified, then it is sent to the essential comparison step. Otherwise, it is sent to the group of unverified events which should be analyzed by the security analyst.

(ii) Essential Comparison. With respect to each security event sent from the classification step, we compare the corresponding essential elements with its real data. If it satisfies all the essential elements, then it is sent to the optional comparison step. Otherwise, it is regarded as false positive which should not be analyzed by the security analyst anymore.

(iii) Optional Comparison. With respect to each security event sent from the essential comparison step, we compare the corresponding optional elements with its real data. If it satisfies one of all the optional elements at least, then it is regarded as true positive, that is, real attack. Otherwise, it is sent to the group of verified events. The security analyst needs to check whether it is caused by a real attack or not using the verification results obtained from the essential and optional elements. This means that the security analyst does not have to analyze it anymore.

In case of the threshold based security events, the verification process is as follows.

(i) Classification. This classification step aims to check whether the security event fed into the verification process is verifiable or not. If the corresponding security event can be verified, then it is sent to the Extracting step. Otherwise, it is sent to the group of unverified events which should be analyzed by the security analyst.

(ii) Extracting. It extracts source IP address from the original security event sent from the classification step. The source IP address is sent to the history comparison step.

(iii) History Comparison. With respect to the source IP address sent from the extracting step, we compare it with that of the other security events in which the time
interval is one day. If a security event exists in that its source IP address is the same with the original security event, then the original security event is regarded as true positive, that is, real attack. Otherwise, it is regarded as false positive which should not be analyzed by the security analyst anymore.

(iv) **Darknet Comparison.** With respect to the source IP address sent from the extracting step, we compare it with a set of unused IP addresses (i.e., darknet). If the source IP address matches to the darknet, then it is regarded as true positive, that is, real attack. Otherwise, it is regarded as false positive which should not be analyzed by the security analyst anymore.

### 4. Performance Evaluation

In order to evaluate effectiveness of the proposed verification methodology based on heuristic analysis, we first estimated the reduction rate of the security events by the proposed verification methodology. For this evaluation, we compared the number of the entire security events with that of the top 20 security events which can be applied to the proposed methodology. Figure 5 shows the number of the entire security events and the top 20 security events and their ratio. From Figure 5, we can easily see that the entire security events numbered 1,528,730,667 and the number of the top 20 security events was 1,017,676,130 which are responsible about 67% of the entire security events used in our experiment. This result means that the number of security events which should be analyzed by the security analyst was reduced by 33% of the entire security events if we apply the proposed verification methodology to the top 20 security events. Therefore, it could be said that the proposed verification methodology contributes to improving the rapidity of the security monitoring and response.

Secondly, we estimated the accuracy of the proposed verification methodology. For this evaluation, we further investigated the security events that belong to the true positive (real attacks) by the proposed verification methodology. As described in Section 3.3, the total number of the real attacks that were identified by the security analyst of S&T-SEC was 960. However, when we applied the verification methodology to the entire security events, we observed that there are 140 additional candidates for real attacks as shown in Table 3. In other words, 131 signature based security events were matched to the essential and the optional elements and they are grouped into the true positive. Also, 9 threshold based security events were satisfied with one of the two conditions (i.e., event history and darknet) and, consequently, they were identified as real attacks. In our further analysis, we recognized that the 140 candidates are related to real attacks. This result means that the proposed verification methodology can contribute to detection of true negatives that were not identified by the security analyst. Some examples of the 140 true negatives can be referred from our previous work [17].

Finally, we estimated the accuracy of the proposed verification methodology using new set of the security events (May 1st, 2012∼December 31st, 2012 and January 1st, 2013∼December 31st, 2013). Table 4 shows the evaluation results.
### Table 3: Number of the real attacks unidentified by the security analyst.

| Event name       | Number of total security events | Number of true positives | Number of true negatives | Period             |
|------------------|--------------------------------|--------------------------|--------------------------|--------------------|
| 1 D**5           | 507                            | 170                      | 21                       | 2012.05.01–09.30   |
| 2 F****w         | 142                            | 98                       | 31                       | 2012.05.01–09.30   |
| 3 Su******us-net***504 | 1101                           | 55                       | 6                        | 2012.05.01–12.31   |
| 4 R***ng         | 999                            | 47                       | 4                        | 2012.05.01–12.31   |
| 5 In***-co***ct  | 324,247                        | 63                       | 9                        | 2012.05.01–12.31   |
| 6 G****t-Sub***e | 6,740                          | 26                       | 4                        | 2012.05.01–09.30   |
| 7 cn(g***t)      | 41,153                         | 102                      | 12                       | 2012.05.01–12.31   |
| 8 Do***bot       | 212                            | 58                       | 5                        | 2012.05.01–12.31   |
| 9 H**me          | 12,244                         | 78                       | 7                        | 2012.05.01–12.31   |
| 10 Su******us-Net***501 | 1,101                       | 19                       | 6                        | 2012.05.01–12.31   |
| 11 Net***2010(h***th) | 11,422                        | 53                       | 10                       | 2012.05.01–12.31   |
| 12 tcp syn flooding | 9,474,044                     | 15                       | 4                        | 2012.05.01–12.31   |
| 13 Mybi_***-co***ct | 65                            | 17                       | 0                        | 2012.05.01–12.31   |
| 14 Rat(***.dll)  | 726                            | 28                       | 10                       | 2013.01.01–07.31   |
| 15 udp-fragmented | 12,009,382                    | 8                        | 2                        | 2012.05.01–12.31   |
| 16 Da*****et_5.3.1 | 97                            | 37                       | 0                        | 2013.01.01–07.31   |
| 17 udp-same ip   | 870,139,035                    | 7                        | 1                        | 2013.01.01–07.31   |
| 18 tcp service scan | 125,037,129                    | 7                        | 2                        | 2012.05.01–12.31   |
| 19 RAT(G***t.75) | 2,498                          | 24                       | 5                        | 2013.01.01–07.31   |
| 20 Ho***i or L**K | 613,286                       | 48                       | 1                        | 2013.01.01–07.31   |
| Total            | 1,017,676,130                  | 960                      | 140                      |                    |

### Table 4: Accuracy of the proposed verification methodology.

| Event name       | Number of true positives | Period               | Accuracy | Type       |
|------------------|--------------------------|----------------------|----------|------------|
| 1 D**5           | 136                      | 2013.05.01–05.31     | 100%     |            |
| 2 F****w         | 71                       | 2013.01.01–07.31     | 100%     |            |
| 3 Su******us-net***504 | 38                       | 2013.01.01–06.30     | 100%     |            |
| 4 R***ng         | 8                        | 2013.01.01–06.30     | 100%     |            |
| 5 In***-co***ct  | 5                        | 2013.01.01–06.30     | 100%     |            |
| 6 G****t-Sub***e | 17                       | 2013.01.01–03.31     | 100%     | Signature  |
| 7 cn(g***t)      | 89                       | 2013.01.01–07.31     | 87%      |            |
| 8 Do***bot       | 34                       | 2013.01.01–07.31     | 100%     |            |
| 9 H**me          | 2                        | 2013.01.01–06.30     | 100%     |            |
| 10 Su******us-Net***501 | 3                       | 2013.01.01–06.30     | 96%      |            |
| 11 Net***2010(h***th) | 60                       | 2013.01.01–07.31     | 100%     |            |
| 12 tcp syn flooding | 15                       | 2012.05.01–12.31     | 80%      | Threshold  |
| 13 Mybi_***-co***ct | 48                       | 2013.01.01–07.31     | 93%      | Signature  |
| 14 Rat(***.dll)  | 8                        | 2013.08.01–09.30     | 100%     |            |
| 15 udp-fragmented | 8                        | 2012.05.01–12.31     | 100%     | Threshold  |
| 16 Da*****et_5.3.1 | 11                       | 2013.08.01–09.30     | 100%     | Signature  |
| 17 udp-same ip   | 7                        | 2012.05.01–12.31     | 100%     | Threshold  |
| 18 tcp service scan | 6                        | 2013.01.01–07.31     | 66%      |            |
| 19 RAT(G***t.75) | 8                        | 2013.08.01–09.30     | 100%     | Signature  |
| 20 Ho***i or L**K | 14                       | 2013.08.01–09.30     | 100%     |            |
| Total            | 588                      | 96.1%                |          |            |
Note that the evaluation period of each type of the security events is different because the corresponding real attacks were caused by different periods. We applied the proposed verification methodology to the top 20 security events and estimated the verification accuracy by identifying whether the true positives by the verification methodology are real attacks or not. From Table 4, we can easily see that the total accuracy of the proposed verification methodology is 96.1%. In addition, only 5 types of the security events among the top 20 security events have less than 100% of the accuracy and the other 15 types of the security events have 100% of the accuracy. In order to improve the accuracy of the 5 types of the security events, it is needed to verify them by using additional dynamic features, not only static features, that is, the essential and the optional elements, proposed in this paper.

5. Conclusion

In this paper, we have proposed the automated verification methodology of the security events based on heuristic analysis. The proposed verification methodology aims to automatically identify real cyberattacks from the security events and filter out false positives. During the heuristic analysis, we used the 1,528,730,667 security events that were obtained from Science and Technology Security Center (S&T-SEC) [14]. Among the 1,528,730,667 security events, we extracted the core security events that were caused by the real cyberattacks and selected the top 20 types of the security events in the number of the real attacks that they raised. By analyzing the top 20 types of the security events, we discovered essential elements and optional elements for using in the automated verification of the security events.

The evaluation results showed that the number of security events which should be analyzed by the security analyst was reduced by 33% of the entire security events if we apply the proposed verification methodology to only the top 20 security events. This means that the proposed verification methodology contributes to improving the rapidity of the security monitoring and response. Also, we recognized that the proposed verification methodology contributed to detection of 140 true negatives that were not identified by the security analyst. Finally, the total accuracy of the proposed verification methodology was 96.1% with respect to the top 20 types of the security events. Furthermore, it is very important that the collection terms of the security events used in this paper never affect the verification quality of the proposed methodology. Because each security event is raised when a specific cyberattack happened, it always corresponds to the same cyberattack. Therefore, if we apply the proposed methodology into the top 20 types of the security events, its verification quality is always the same even if the collection terms are different.

In the future work, we first consider analyzing the cyberattacks in which they use encrypted payload. Because the signature based security devices, for example, IDSs, cannot detect such cyberattacks, the proposed methodology deals with only the security events detected by them. In addition, it is needed to estimate the verification quality of the proposed methodology when we apply it into the security events that were collected in the different terms, because the top 20 types of the security events will be changed for each term according to the alteration of the attack fashion.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by “Construction and Operation of Science and Technology CyberSecurity Center” Program funded by the Ministry of Science, ICT, and Future Planning (G-14-GM-IR02) and “A Study on Advanced Autoverification Technology of Large-scale Security Events” Program funded by Korea Institute of Science and Technology Information (K-14-L06-C15-S01).

References

[1] D. E. Denning, “An intrusion-detection model,” IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222–232, 1987.
[2] D. Yu and D. Frincke, “A novel framework for alert correlation and understanding,” in Applied Cryptography and Network Security, vol. 3089 of Lecture Notes in Computer Science, pp. 452–466, Springer, Berlin, Germany, 2004.
[3] H. Debar and A. Wespí, “Aggregation and correlation of intrusion-detection alerts,” in Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection (RAID ’01), pp. 85–103, Springer, 2001.
[4] G. Giacinto, R. Perdisci, and F. Roli, “Alarm clustering for intrusion detection systems,” in Proceedings of the International Conference on Machine Learning and Data Mining (MLDM ’05), vol. 3587 of Lecture Notes in Artificial Intelligence, pp. 184–193, 2005.
[5] P. Ning, Y. Cui, and D. S. Reeves, “Analyzing intensive intrusion alerts via correlation,” in Proceedings of the 5th International Symposium on Recent Advances in Intrusion Detection (RAID ’02), pp. 74–94, Springer, Zurich, Switzerland, October 2002.
[6] J. J. Treinen and R. Thurimella, “A framework for the application of association rule mining in large intrusion detection infrastructures,” in Recent Advances in Intrusion Detection, vol. 4219 of Lecture Notes in Computer Science, pp. 1–18, 2006.
[7] F. Xiao, S. Jin, and X. Li, “A novel data mining-based method for alert reduction and analysis,” Journal of Networks, vol. 5, no. 1, pp. 88–97, 2010.
[8] P. Ning, Y. Cui, and D. S. Reeves, “Constructing attack scenarios through correlation of intrusion alerts,” in Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 245–254, New York, NY, USA, November 2002.
[9] P. Ning, D. Xu, C. G. Healey, and R. S. Amant, “Building attack scenarios through integration of complementary alert correlation methods,” in Proceedings of the 11th Annual Network and Distributed System Security (NDSS ‘04), pp. 97–111, 2004.
[10] P. Ning, Y. Cui, D. S. Reeves, and D. Xu, “Techniques and tools for analyzing intrusion alerts,” ACM Transactions on Information and System Security, vol. 7, no. 2, pp. 274–318, 2004.
[11] S. O. Al-Mamory and H. Zhang, "IDs alerts correlation using grammar-based approach," *Journal in Computer Virology*, vol. 5, no. 4, pp. 271–282, 2009.

[12] J. Song, H. Takakura, and Y. Kwon, "A generalized feature extraction scheme to detect 0-day attacks via IDS alerts," in *Proceedings of the International Symposium on Applications and the Internet (SAINT '08)*, pp. 51–56, IEEE Computer Society Press, 2008.

[13] Y. Livnat, J. Agutter, S. Moon, and S. Foresti, "Visual correlation for situational awareness," in *Proceedings of the IEEE Symposium on Information Visualization (InfoVis '05)*, pp. 95–102, October 2005.

[14] K. Lakkaraju, W. Yurcik, and A. J. Lee, "NVisionIP: netflow visualizations of system state for security situational awareness," in *Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC '04)*, pp. 65–72, ACM, Washington, DC, USA, October 2004.

[15] J. S. Song, S. S. Choi, K. I. Kim, and H. S. Park, "Toward a lightweight security monitoring and response scheme based on automated verification of IDS alerts," in *Proceedings of the International Conference on Intelligence Fusion (ICIF '13)*, pp. 19–20, 2013.

[16] "Science&Technology Security Center," http://www.sntsec.or.kr/.

[17] S. S. Choi, J. S. Song, S. H. Kim, and S. K. Kim, "A model of analyzing cyber threats trend and tracing potential attackers based on darknet traffic," *Security and Communication Networks*, vol. 7, no. 10, pp. 1612–1621, 2014.
Submit your manuscripts at http://www.hindawi.com