

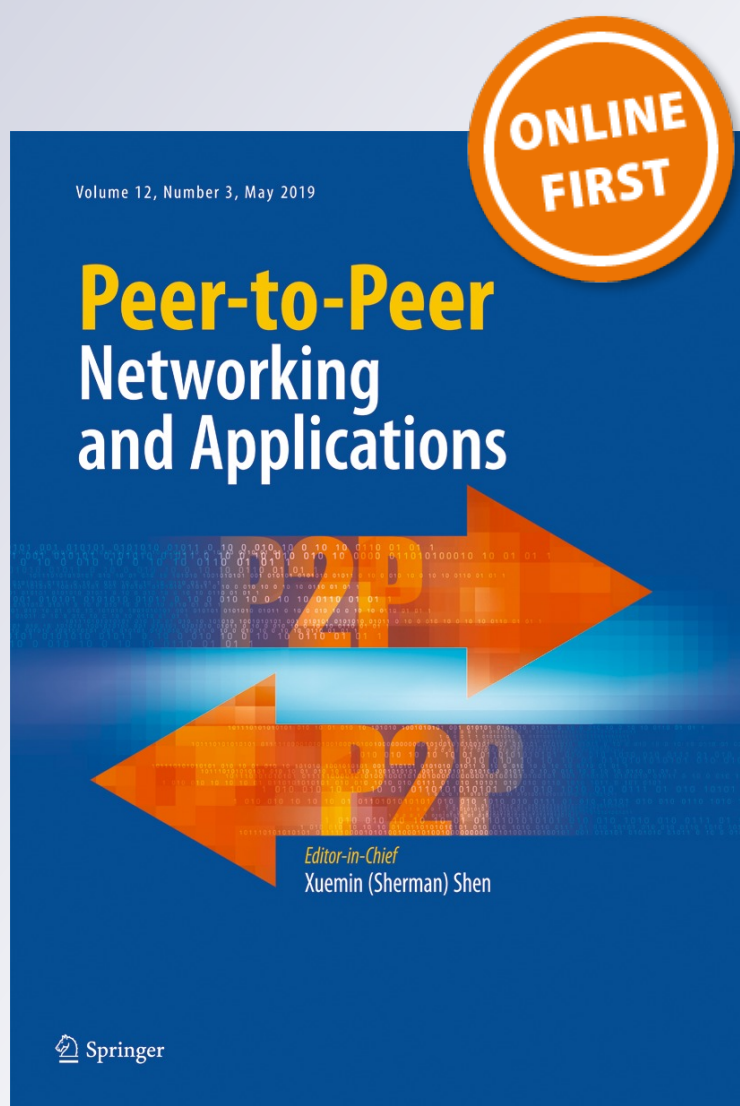
# *Internet of things system technologies based on quality of experience*

**Dongsik Son, Seungpyo Huh, Taejin Lee  
& Jin Kwak**

**Peer-to-Peer Networking and  
Applications**

ISSN 1936-6442

Peer-to-Peer Netw. Appl.  
DOI 10.1007/s12083-019-00727-1



**Your article is protected by copyright and all rights are held exclusively by Springer Science+Business Media, LLC, part of Springer Nature. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at [link.springer.com](http://link.springer.com)".**



# Internet of things system technologies based on quality of experience

Dongsik Son<sup>1</sup> · Seungpyo Huh<sup>2</sup> · Taejin Lee<sup>3</sup> · Jin Kwak<sup>4</sup>

Received: 11 October 2018 / Accepted: 31 January 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

With rapid development of IOT technology, QOE is considered in design process to ensure that end users are satisfied using IOT applications and System. However, as if to mock this new paradigm, hackers in the cyber world are attacking the subjective acceptability of an IOT application perceived by the end user through intelligent cyber threats such as DDoS attacks, wiretapping, remote control, ID/Password hijacking using IOT devices. Due to this, the number of SIEM introduction is increasing in order to detect threat patterns in a short period of time with a large amount of structured/unstructured data from IOT devices, to precisely diagnose crisis to threats, and to provide an accurate alarm to an administrator by correlating collected information. In this paper, we design and propose an automated profiling model that leverages the analysis results of the IOT event profiling in SIEM to improve attack detection, efficiency and speed against potential threats.

**Keywords** IOT security profiling · IOT big data analysis · IOT correlation

## 1 Introduction

According to IDC's latest research report, the global big data and analytics market is expected to grow by 12.4%

---

This article is part of the Topical Collection: *Special Issue on IoT System Technologies based on Quality of Experience*  
Guest Editors: Cho Jaeik, Naveen Chilamkurti, and SJ Wang

---

✉ Jin Kwak  
jkwak.security@gmail.com

Dongsik Son  
over0033@gmail.com

Seungpyo Huh  
huhspunk@gmail.com

Taejin Lee  
kinjecs0@gmail.com

<sup>1</sup> ISAA Lab., Department of Computer Engineering, Ajou University Suwon, Suwon, South Korea

<sup>2</sup> Gyeonggi University, Suwon, South Korea

<sup>3</sup> Department of Computer Engineering, Hoseo University Cheonan, Asan, South Korea

<sup>4</sup> Department of Cyber Security, Ajou University Suwon, Suwon, South Korea

year-on-year to reach \$150.8 billion, and investment in analytical solutions for collecting events from numerous IOTs are getting increasing [1, 4, 18]. Today, security threats can cause enormous financial damage with new attack techniques. Therefore, each security companies have been actively researching and developing SIEM(Security Information & Event Management) which is big data security solution. Also, the number of companies introducing SIEM solutions is continuously increasing [2, 19, 21]. The real-time detection field, which is the core of security monitoring, requires rapid detection of attack sites that are continuously attacking in various ways and correlation analysis in various source logs [20]. However, existing SIEMs have difficulty in preemptive response and rapid analysis due to complex processes and lack of awareness of functions. Therefore, it is necessary to set up an effective function in the managerial aspect so that administrator can easily create and profile rules in various security event logs. This paper is to propose profiling auto-generation model with correlation analysis based on big data log. The proposed model analyzes the rank by attack site and target IP in the existing profile analysis result using graph analysis and correlates these result with Security Intelligence system to identify the most important IP and, finally recommend the profile generation model to the administrator. [22, 23] The composition of this paper is as follows. Chapter 2 analyzes the existing analysis model of integrated security

monitoring. Section 3 proposes a profiling auto-generation model with correlation analysis based on big data Log. Section 4 evaluates the experimental results of the proposed model and last Section 5 concludes.

## 2 Review of related literature

In the case of the Big Data Security Analysis Model, there are a number of analytical methods to detect potential attacks for security threats. Andrey F is a widely used predictive analysis that uses some linear relationship between two variables in various fields. In the security field, It detects abnormal symptoms from heterogeneous security event logs and associates them with each other in gathered logs. Finally, it deduces and finds an event that can be suspected of an attack. [3] However, as the number of data increases, the correlation coefficient value becomes larger and reliability may be lowered [5]. SIEM with rule-based analysis studied by In-Seok J and Idoia A filters threat events with conditions such as AND, OR between various events. The filter condition has the advantage of alerting to the administrator in the form of alarm such as Dashboard Pop-up, SMS, E-Mail if various conditions such as attack IP, destination IP, signature, and port are satisfied. But, there are disadvantages in that threats cannot be detected when rules do not exist if the detection rate is good within the defined condition range. So there is always a need for a lot of human resources to maintain the latest rules. [6, 7] Alistair S and Manuel E can execute effective detection interworking with SIEM based on highly expert-based scenarios with years of know-how and technology, such as hacking threats response, malware and ransomware analysis. However, there is a disadvantage in that it is necessary to have a specialist group capable of analyzing difficult infringement incidents and that human resources and physical resources should be fully supported. [8, 9] Alvaro A and Matthias is an analytical technique that can easily be applied in various fields, from traditional statistical analysis to statistical analysis using a parallel framework. Recently, it is equipped as a function in the security solution. Especially, in the field of security monitoring service, there is active researchers on threat analysis using machine learning and artificial intelligence technology, and blocking and prevention analysis of abnormal patterns in advance. [10, 11] Kuan-Yu C and Gerardo C create a pattern or rule based on a specific time with time series analysis and generate an alarm to the administrator when it is determined that the pattern is abnormal. It is mainly used for short-term prediction because it has an

advantage that it can analyze the minimum data quickly. However, there is a difficulty in long-term prediction and it is possible to perform mid-term prediction using a lot of information, but a more complicated process is required [12, 13]. In addition, Yunhong et al. Claim that there are Collaborative Filtering model and Contents-based Filtering model in introducing profile recommendation technology [14], and Collaborative Filtering model is divided into User-based and Item-based again. Collaborative filtering has the advantage of intelligent analysis that it can generate another item is similar to recommendation by measuring similarity, However, there is a disadvantage that it is difficult to recommend without existing data referred to as “cold start.” [15, 16] The Content-based filtering model analyzes contents itself rather than analyzing the behavior of users and the properties of items so that it solves the cold start problems arising from collaborative filtering. [17] Therefore, in order to solve the problems of the above analysis, in Chapter 3, we propose a model of automatic generation recommendation method that enables cross-reference analysis to minimize security administrator's decision.

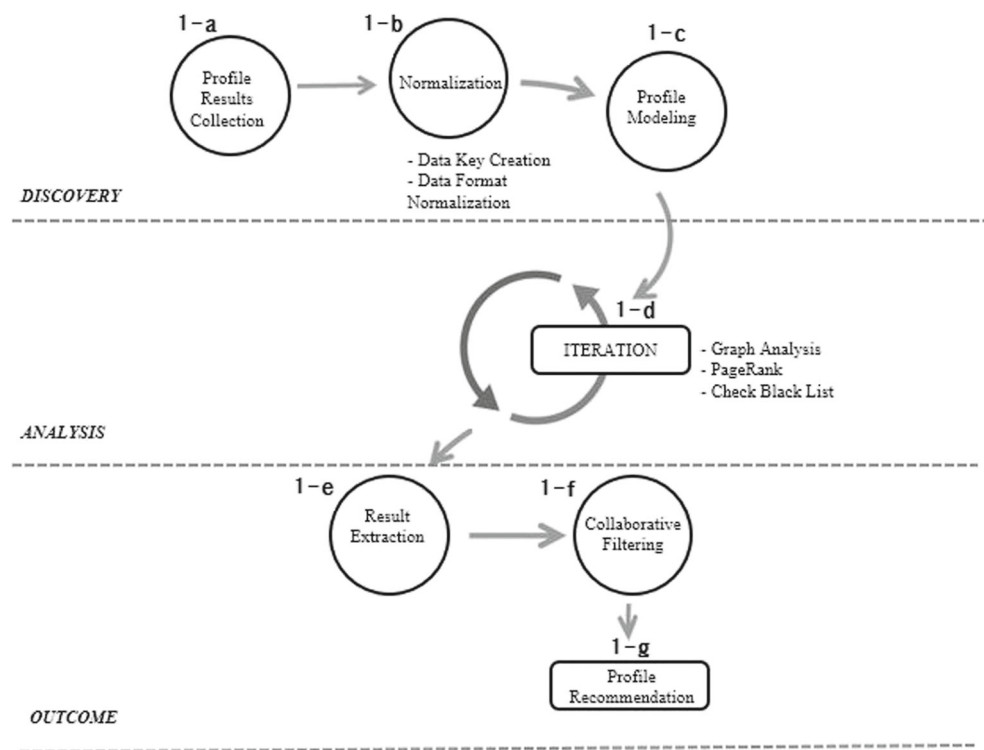
## 3 Proposed scheme

### 3.1 System overview

SniperBD1 which is a big data security solution for collecting events from numerous IOTs has a profile function that stores the results detected by different conditions in the heterogeneous security log. Accordingly, a formalization work is required in order to analyze the stored profile results according to the proposed environment. As shown in Fig. 1, 1-a the result of the profile is collected, 1-b after the formalization work, 1-c it is constructing and modeling dataset according to the graph analysis and the page rank analysis, which is the core analysis model. 1-d At this time, after interworking with Security Intelligence information system and sort IP which has malicious activity, 1-e the result is extracted, 1-f recommend profile auto-generation based on collaborative filtering and 1-g finally perform an alert function to the administrator.

### 3.2 Profile result collecting - data acquisition

In the proposed system, if a profile meets various conditions based on heterogeneous security log, it is considered abnormal behavior and is saved as a profiling result. The profiles have various conditional expressions as shown in Table 1. If it fits the conditions, the profile result is stored

**Fig. 1** System overview

in the chronological order in which it is detected and has a filename such as a date and profile ID (Fig. 2).

Each item of the profile result is configured as result information detected by profile such as Row key, device IP, device name, start time, end time, source IP, destination IP, source port and destination port as shown in Fig. 3. Therefore, the necessary information for modeling the profile is source IP and destination IP in Fig. 3 and Profile ID in Fig. 2.

### 3.3 Profile modeling – generation process

The source IP and the destination IP among the profile result information are formalized and expressed as a graph theory composed of two elements, namely a node (vertex) and an edge. Graph theory is a mathematical structure used to model relationships between objects and consists of connecting nodes and edges. It is divided into a non-directional graph and directional graph according

**Table 1** Stored profile

|           | Seq. | Operator | Field     | Operator | Value               | Type | Regex |
|-----------|------|----------|-----------|----------|---------------------|------|-------|
| Diffusion | 1    |          | External  | =        | Intelligence IP     | List | Off   |
| Type 1    | 2    | And      | Signature | =        | Win32Bagle.worm     | Text | Off   |
|           | 3    | Or       | Signature | =        | Win32Netsky.worm    | Text | Off   |
|           | 4    | Or       | Signature | =        | Win32Mytob.worm     | Text | Off   |
| APT       | 1    |          | FW log    | =        | Firewall Accept IPs | List | Off   |
| Type 1    | 2    | And      | FW log    | =        | Hidden Server IPs   | List | Off   |
|           | 3    | Or       | FW log    | =        | Block Ports         | List | Off   |
|           | 4    | Or       | Signature | =        | Terminal Service    | Text | Off   |

```

20190210022514_7672.26 20190210045349_9074.26 20190210072250_480.26 20190210094538_1885.26 20190210120644_3292.26 20190210143147_4697.26 20
20190210022524_7673.26 20190210045349_9074.26 20190210072259_481.26 20190210094540_1886.26 20190210120645_3293.26 20190210143204_4698.26
20190210022526_7674.26 20190210045345_9075.26 20190210072302_482.26 20190210094546_1887.26 20190210120655_3294.26 20190210143215_4699.103
20190210022536_7675.26 20190210045346_9076.26 20190210072314_483.26 20190210094550_1888.26 20190210120704_3295.26 20190210143219_4700.26
20190210022538_7676.26 20190210045355_9076.39 20190210072314_484.26 20190210094555_1889.26 20190210120711_3296.26 20190210143215_4701.26
20190210022546_7677.26 20190210045355_9076.39 20190210072315_485.26 20190210094617_1890.26 20190210120714_3297.26 20190210143227_4702.26
20190210022546_7678.26 20190210045357_9080.26 20190210072317_486.114 20190210094617_1891.26 20190210120722_3298.26 20190210143237_4703.26
20190210022547_7679.26 20190210045404_9881.26 20190210072322_487.26 20190210094629_1892.26 20190210120723_3299.26 20190210143242_4704.26
20190210022557_7680.26 20190210045405_9882.26 20190210072327_488.26 20190210094641_1893.26 20190210120732_3300.39 20190210143247_4705.26
20190210022558_7681.26 20190210045415_9883.26 20190210072333_489.26 20190210094649_1894.26 20190210120732_3301.26 20190210143256_4706.26
20190210022611_7682.26 20190210045416_9884.26 20190210072341_490.26 20190210094655_1895.26 20190210120735_3302.26 20190210143258_4707.26
20190210022613_7683.26 20190210045418_9885.26 20190210072346_491.26 20190210094707_1896.26 20190210120747_3303.26 20190210143307_4708.26
20190210022614_7684.26 20190210045434_9886.26 20190210072353_492.26 20190210094707_1897.103 20190210120750_3304.26 20190210143312_4709.26
20190210022629_7685.26 20190210045447_9887.26 20190210072355_493.30 20190210094714_1898.26 20190210120759_3305.26 20190210143316_4710.26
20190210022638_7686.103 20190210045448_9888.26 20190210072355_494.26 20190210094717_1899.26 20190210120804_3306.26 20190210143323_4711.26
20190210022644_7687.26 20190210045505_9889.26 20190210072356_495.26 20190210094719_1900.26 20190210120804_3307.103 20190210143324_4712.26
20190210022646_7688.26 20190210045515_9900.26 20190210072400_496.88 20190210094725_1901.26 20190210120812_3308.26 20190210143336_4713.26
20190210022656_7689.26 20190210045517_9901.26 20190210072405_497.26 20190210094740_1902.26 20190210120813_3309.26 20190210143344_4714.26
20190210022710_7690.26 20190210045536_9902.26 20190210072412_498.37 20190210094746_1903.26 20190210120814_3310.39 20190210143348_4715.26
20190210022712_7691.26 20190210045537_9903.103 20190210072415_499.26 20190210094754_1904.26 20190210120814_3311.26 20190210143348_4716.26
20190210022725_7692.26 20190210045547_9904.26 20190210072424_500.26 20190210094759_1905.26 20190210120814_3312.26 20190210143356_4717.26
20190210022727_7693.26 20190210045547_9905.26 20190210072436_501.26 20190210094804_1906.26 20190210120815_3313.26 20190210143356_4718.30
20190210022746_7694.26 20190210045602_9906.26 20190210072437_502.26 20190210094807_1907.26 20190210120828_3314.26 20190210143356_4719.26
20190210022746_7695.26 20190210045615_9907.26 20190210072438_503.26 20190210094815_1908.26 20190210120846_3315.26 20190210143358_4720.26
[root@sniperbd1 20190210]#

```

**Fig. 2** Profile results screen

to the direction of an edge. As shown in Fig. 4, the proposed graph network configuration step allows the symmetric edges to have multiple profile relationships between the same IPs and to have the multi-edge in parallel. In other words, one IP is connected to several profiles, and any IP is structured to allow various connections. Table 2 shows the configuration profile items in Fig. 4.

### 3.4 Page rank reference model

PageRank is aimed at getting the rank of each document based on how many documents are linking to it in a web page. This algorithm is used to get the rank of web pages, but it is also used to evaluate the importance of scientific papers or to find influential SNS users. In the proposed method, assuming that each IP is a web page and edge is a profile, it calculates rank ratio by applying the

PageRank algorithm. In other words, if the IP referenced in each IP has many edges, the rank is high and it means that this IP is important from another IP. So it is necessary to look closely at security monitoring. Before referring to this model, nodes are represented by IP and links are represented by profiles as shown in Fig. 4. Also, as shown in Fig. 5, a profile is shown between Ips. Table 3 shows the numerical information about the node, and a long value is sequentially given to generate a unique ID for IP.

In the link scheme of Table 4, the Source ID and the Destination ID represent the unique ID value sequentially given in Table 2, and the profile ID is defined as Property accordingly. For example, the first row of Table 4 is the profile log directing the source 192.168.1.100 to the destination 192.168.1.101 in profile 31. According to these schemes, you can configure Vertex (node) and Edge (link) by providing a graph analysis library in Apache Spark to

[illegible]

**Fig. 3** Profile results contents information screen

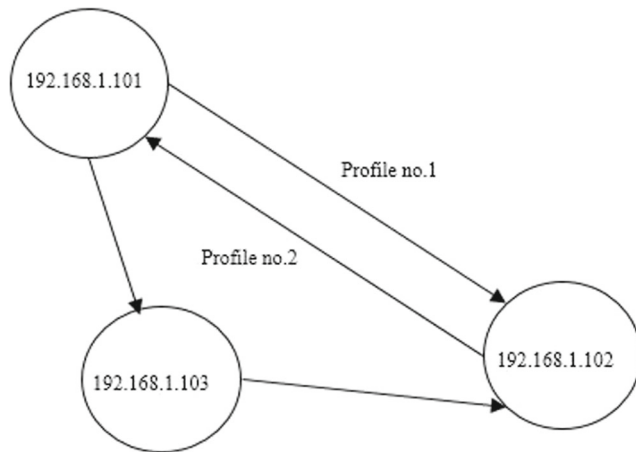


**Fig. 4** Profile graph network configuration

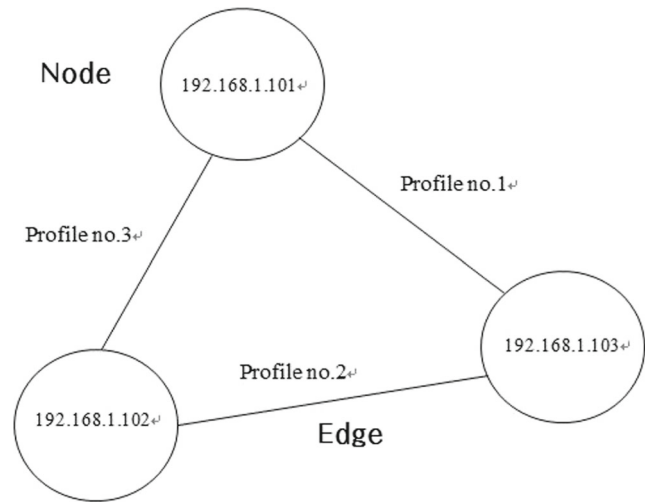


**Fig. 5** Graph model showing node and link relationship

create nodes and edges. The pseudo-code that configures the vertex and edge is as follow as Fig. 6. Figure 6 is the pseudo-code for creating Edge and Vertex. To import into Apache Spark's RDD type, after importing profile information from the database, it converts the IP value into Long type and creates Edge with source IP, destination ID and profile ID value. Also, it converts all of profile ID value, source IP, destination IP to Long value except for duplicate IP, then create the vertex and is terminated.

Figure 7 shows the pseudo-code that performed graph-based PageRank analysis with Edge RDD and Vertex RDD values created in Fig. 6. The representation of the code is simple, but it performs core function for finding the most important IP. When we look at the individual IPs as vertices, they are considered to interact with each other and give importance to all IP (source, destination) existing in the whole graph. This is a method of recommending an important IP to administrators as a given result. It is not simply to give a score because of the high frequency but to give a score to the vertices that seem to be important on the graph. Equation 1 shows a formula for assigning a score to a vertex and updates the rank of each IP with the weighted sum of neighbor ranks for the directional network

G with the neighbor matrix A. In this, 'a' is a Damping Factor with a value between 0 and 1, 'N' is the total number of nodes, and 'd<sub>out</sub>(v)' is a degree toward the link of v. The K-means clustering algorithm starts with a training data set and a given number of clusters K. The samples in the training data set are assigned to a cluster based on a similarity measurement. Euclidean distance is generally used to measure the similarity. The K-means algorithm tries to find an optimal solution by minimizing the square error:

$$PR(u) = \frac{1-a}{N} + a \sum_v A_{vu} PR(v) / d_{out}(v) \quad (1)$$

Figure 7 shows pseudo-code that finds the edge of the source IP and destination IP in node using Graph.triplets.filter, and extract black IP list and each array interworking security Intelligence information system with the found source and destination IP.

### 3.5 Page rank reference model

The proposed automatic creation recommendation model extracts the accuracy based on collaborative filtering and

**Table 2** Profile collection contents

| IP            | DST IP        | PROFILE ID |
|---------------|---------------|------------|
| 192.168.1.101 | 192.168.1.102 | 1          |
| 192.168.1.102 | 192.168.1.103 | 2          |
| 192.168.1.103 | 192.168.1.105 | 3          |

**Table 3** Node scheme

| IP | Property      |
|----|---------------|
| 1  | 192.168.1.100 |
| 2  | 192.168.1.101 |
| 3  | 192.168.1.102 |

**Table 4** Link scheme

| Source ID | Destination ID | Property |
|-----------|----------------|----------|
| 1         | 2              | 31       |
| 2         | 3              | 22       |
| 3         | 2              | 12       |

recommends a new profile between similar sets of IP. Collaborative filtering can be learned without knowing the attributes of profile ID and IP, and the data set before learning is shown in Table 5. Table 5 shows the dataset based on the source IP, destination IP, profile ID, and rank ratio created in Fig. 8 through the PageRank reference model. Also, the datasheet is made by dividing into source IP and the destination IP as shown in Table 5. The created dataset does not include profile ID and IP information. In Table 5, the first field Profile ID is the information stored in the actual database, the second field Item is IP information digitized as an integer. The third field is the rank ratio indicating which IP is most important by measuring which IP is most connected between IPs through the PageRank reference model. Figure 8 shows the pseudo-code for recommending the auto-created profile IP. It creates an ALS recommendation model to abstract the profile IP, profile ID, and rank data. It extracts profile\_ip and profile\_id from ratings RDD and performs prediction for the pair of profile\_ip - item(profile\_id) using model.predict. 'profile\_ip - item(profile\_id)' is used as key and predicted rating is used as a value. It creates a new RDD by combining two RDDs with the same type of key and the actually predicted ratings

for the pair of profile\_ip – item(profile\_id). Finally, it sums the squared errors by using 'reduce' and calculate the mean square error(MSE) by dividing by 'rateandpreds.count'.

## 4 Experimental result

### 4.1 Experimental environment

The experimental environment of the proposed system is shown in Table 6, and the security log is collected in more than 300 security solution. Therefore, we did an experiment in the environment where more than 6000 are detected during 1 month in October among the results detected by the profile function described above.

Figure 9 shows about 80 profile screens defined in the proposed system. Each profile was defined with different conditions and the experiment was performed under the condition that more than 6000 cases were detected per day on the profile.

### 4.2 Experimental Analysis

Figure 10 shows the visualization result corresponding to the main part of this study. On the left side is the profile recommendation screen for the source and the right is the profile recommendation screen for the destination. In Fig. 10, each IP is a node, and the profile is a link. The larger the size of a circle is displayed, the higher the risk cross-reference rate is, and the smaller the value is displayed, the lower the risk cross-reference rate is. It means that

**Fig. 6** Graph model showing node and link relationship

| Function 1 – Vertex and Edge Generation Function <sup>o</sup>                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Description. This function import profile information from the database, creates an Edge. It import IP value, creates a Vertex (node) <sup>o</sup> |
| <i>K</i> : Database information <sup>o</sup>                                                                                                       |
| <i>edge</i> : edge buffer <sup>o</sup>                                                                                                             |
| <i>vertex</i> : vertex buffer <sup>o</sup>                                                                                                         |
| 1. <b>while</b> k = 1 to <i>K</i> <b>do</b> <sup>o</sup>                                                                                           |
| 2. <b>for</b> i, j = 1 to DB( <i>key</i> , <i>val</i> ; <i>ue</i> ) <sup>o</sup>                                                                   |
| 3.         getID <- i <sup>o</sup>                                                                                                                 |
| 4.         src, dst <- parse(j) <sup>o</sup>                                                                                                       |
| 5.         result <- ipToLong(src, dst) <sup>o</sup>                                                                                               |
| 6.         edge <- result, getID <sup>o</sup>                                                                                                      |
| 7.         vertex <- i, result <sup>o</sup>                                                                                                        |
| 8. <b>end</b> <sup>o</sup>                                                                                                                         |
| 9. <b>end</b> <sup>o</sup>                                                                                                                         |

**Fig. 7** Performs page rank algorithm after edge and vertex RDD generation

| Function 2 – Page Rank Execution Function <sup>o</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Description. It create RDD with vertex and edge buffer information and it perform page rank algorithm after constructing graph.<sup>o</sup></p> <p><sup>o</sup></p> <p><i>graph</i> : Graph<sup>o</sup></p> <p><i>eRDD</i> : edge RDD<sup>o</sup></p> <p><i>vRDD</i> : vertex RDD<sup>o</sup></p> <p><i>srcRank</i> : Profile source IP Rank Buffer<sup>o</sup></p> <p><i>dstRank</i> : Profile destination IP Rank Buffer<sup>o</sup></p> <p><sup>o</sup></p> <ol style="list-style-type: none"> <li>1. <i>graph</i> &lt;- Graph(<i>vRDD</i>, <i>eRDD</i>)<sup>o</sup></li> <li>2. <i>ranks</i> &lt;- <i>graph</i>.pagerank.vertices<sup>o</sup></li> <li>3. <i>result</i> &lt;- <i>ranks</i>.join(<i>vRDD</i>)<sup>o</sup></li> <li>4. <b>while</b> <i>k</i> = 1 to <i>result</i> <b>do</b><sup>o</sup></li> <li>5.     <i>src</i> &lt;- <i>graph</i>.triplets.filter(<i>t</i> =&gt; <i>t</i>.srcAttr.equals(<i>k</i>))<sup>o</sup></li> <li>6.     <i>dst</i> &lt;- <i>graph</i>.triplets.filter(<i>t</i> =&gt; <i>t</i>.dstAttr.equals(<i>k</i>))<sup>o</sup></li> <li>7.     <i>result</i> &lt;- CheckBlackList(<i>src</i>, <i>dst</i>)<sup>o</sup></li> <li>8.     <i>srcRank</i> &lt;- <i>k</i>, <i>src</i>, <i>result</i><sup>o</sup></li> <li>9.     <i>dstRank</i> &lt;- <i>k</i>, <i>dst</i>, <i>result</i><sup>o</sup></li> <li>10. <b>end</b><sup>o</sup></li> </ol> |

It is considered as more important IP if the risk cross-reference rate has the higher value and, It is considered as relatively less important IP if the risk cross-reference rate has the lower value. In the circle, the threat level indicator is composed of C (critical), H (high), M (medium) and L (low). The Security Intelligence Black IP list is marked with a C rating. Others are indicated by the threat level of the general event log.

The reliability of Fig. 10 can be a reliable measure as it is the mean square error value in the auto-generation recommendation model of the proposed method. Accordingly, It can be determined whether the IP is recommended or not according to the reliability. The new

profile creation screen is shown in Fig. 11, and a new profile is created in the following order.

1. When creating a new profile, if you click the IP that can be recommended among the source IP and destination IP, The screen of 11-a in Fig. 11 appears. You can choose which type to detect in the recommended profile.
2. If the type is determined, You can select whether to detect in real time or background as shown in 11-b screen
3. In 11-c screen for repeat condition setting, the condition for time can be selected in background detection, and the condition for time/number can be selected in real time detection.
4. 11-d is a screen for creating a new profile title. The recommended profile keyword provides a keyword that can be easily set by the administrator. This keyword is found as a word by the tokenizer() function in the title of the correlation profile and is designated as a central noun by extracting a word root with high detection frequency.

If you create a profile in the above process, a recommended profile is created in the existing profile definition screen, as shown the red box in Fig. 12. First, the default setting is changed from 'unused' to

**Table 5** Graph analysis result data set structure

| Profile IP | Profile ID | Rate   |
|------------|------------|--------|
| 120        | 32         | 18.231 |
| 120        | 10         | 15.231 |
| 53         | 12         | 13.123 |
| 53         | 10         | 11.421 |
| 2          | 12         | 10.333 |
| 43         | 32         | 7.123  |

**Fig. 8** Pseudocode to auto generation recommendation model

| Function 3 – Auto Generation Recommendation Function <sup>a</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Description. This function is modeled to perform recommend system, the page rank value is evaluated and the profile is recommended measuring average error squared value based on the page rank value. <sup>a</sup><br><sup>a</sup><br>1. <i>Get profile_IP, profile_ID</i> <sup>a</sup><br>2. <i>ratings &lt;- data.map(case Array(profile_IP, profile_ID, rate))</i> <sup>a</sup><br>3. <i>model &lt;- ALS.train(ratings, rank, numIteration)</i> <sup>a</sup><br>4. <i>item &lt;- ratings.map(case Rating(profile_ip, profile_id, rate)) =&gt; (profile_ip, profile_id)</i> <sup>a</sup><br>5. <i>prediction &lt;- model.predict(item).map(case Rating(profile_ip, profile_id, rate)) =&gt; ((profile_ip, profile_id), rate)</i> <sup>a</sup><br>6. <i>rateandpreds &lt;- ratings.map(case Rating(profile_ip, profile_id, rate)) =&gt; ((profile_ip, profile_id), rate).join(predictions)</i> <sup>a</sup><br>7. <i>MSE &lt;- rateandpreds.map(case ((profile_ip, profile_id), (r1, r2)) =&gt; math.pow((r1 - r2), 2)).reduce(_ + _) / rateandpreds.count</i> <sup>a</sup> |

'enabled', The log detected by the new recommendation profile can be checked on the existing log inquiry screen.

### 4.3 Experimental result

Performance evaluation was performed by using the samples of 45 well-known attack and 50 kinds of unknown attack in order to proceed in two types. The samples of well-known attacks are used by the attack related to 2017 OWASP TOP 5 as shown in Fig. 13 and the samples of unknown attack are used by the attack that was not detected on 'virustotal.com' among the APT attack diagnosed through actual reversing analysis.

In the first experiment, samples were applied to the A-system as a rule-based analysis system (SniperTMS, WINS), the B-system as a correlation-based system, and the proposed system. (AlienVault OSSIM, AT&T) For convenience, it is decided to call the system A and B respectively. However, since the analysis methods and results of the comparative systems are different from each other, they are regarded as the same if they result in

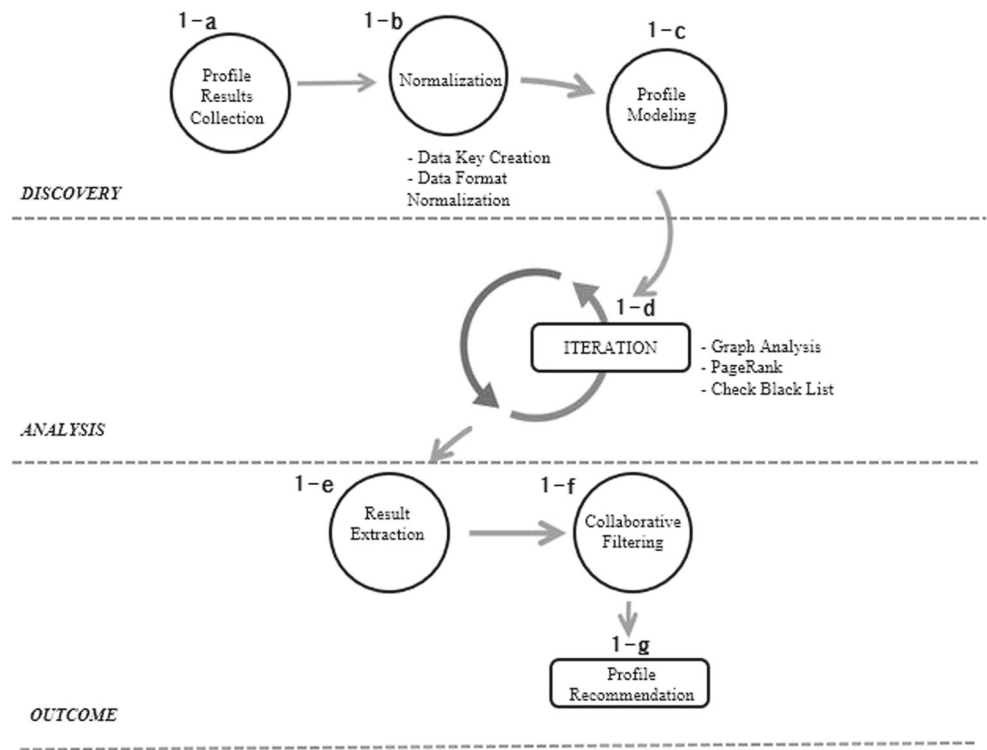
correlated or cross-referenced derivatives. Before detecting potential threats, Table 7 shows the results of preliminary testing of 45 well-known attacks in Fig. 13 for the comparison and proposed systems for one month in October. Table 7 shows the number of generation correlated or referenced by well-known attacks during the month in October. On the average, both systems A and B seem to have a large number of correlated generations for well-known attacks, however, the average number of cross-references in the proposed system is significantly higher than that of the comparison system. Also, in the results verified to determine the accuracy of whether it is an actual well-known attack or not in the proposed system, 150 cases, 83% of the average number of cases, were actually detected as well-known attacks. This means that the newly created profile can detect well-known attacks.

In second experiment, we re-register the profile or rule of each result derived from the first experiment in each system and attack 50 kinds of Unknown attack. Unknown attacks are samples which is classified into five categories such as Scanning, Network, DoS, Web, and System, and which is self-developed to do real malicious behavior. Unknown attack detection is considered as successful detection if the results detected by the condition are extracted or there are cross-referenced generation after unknown attack. As a result of the detection test from July to October as shown in Table 8, The number of the extraction result and reference generation detected by the condition for all attacks were 1, but the proposed system showed more than 2 cases per month. In detail, each one detected in systems A and B was blocked in a whitelist scheme that blocked all values except the first allowed list. Whitelist security can be more secure, but it is not recommended because it is difficult

**Table 6** Proposed system configuration

| Categorization          | Version                   |
|-------------------------|---------------------------|
| OS                      | CentOS 6.6                |
| Store                   | Hadoop 2.x                |
| Analysis                | Apache Spark 1.6.x Graphx |
| Database                | MariaDB 5.1.x             |
| Collaborative Filtering | Apache Spark 1.6.x MLlib  |

**Fig. 9** Proposed system profile set screen



to operate. However, the proposed system detected a large number of unknown attacks, and the number of detections in the proposed system was two in July and an average

of eight in October as shown in Fig. 14. This means that the generated profile condition is to detect the variant of the unknown attack effectively by increasing the detection

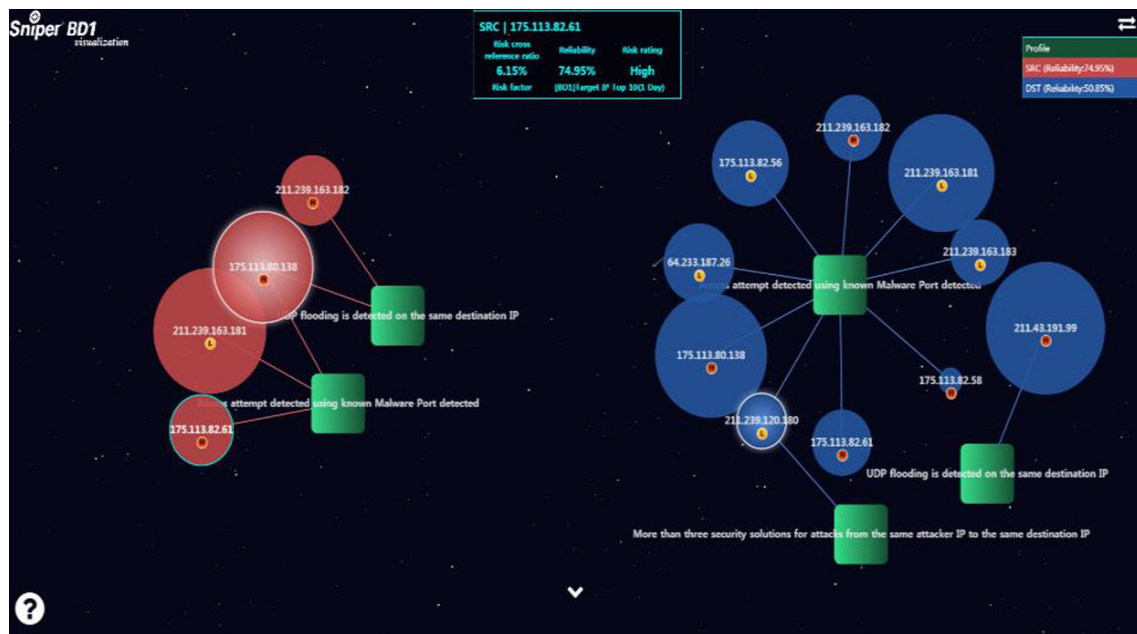


**Fig. 10** Profile auto generation main screen

**11-a**

**1. Select Event type.**

Event

All

**11-b**

**2. Realtime / Background**

Background

Realtime

**11-c**

**3. Select Repeat condition value.**

Every Hour 1 hour ago

**11-d**

**Give a name for a new profile.**

Keyword: Known management port detection, If a new type of attack that has not been detected for a month is detected

| Field                 | Value                |
|-----------------------|----------------------|
| Event Type            | Event                |
| Realtime / Background | Background           |
| Repeat Condition      | Every Day 1 hour ago |

Fig. 11 New profile registration process screen

| ACTIVE OPTION | REG. DATE           | REGISTER DIVISION | SCHEDULE   | PROFILE NAME                                                                                         | CONDITIONS                                      |
|---------------|---------------------|-------------------|------------|------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| Inactive      | 2018-12-18 09:08:50 | Automation        | Background | Known management port detection If a new type of attack that has not been detected for a month is de | Every Day 1 hour ago                            |
| Inactive      | 2018-06-26 16:55:19 | Automation        | Realtime   | UDP Flooding DDoS IoT Botnet Detection                                                               | 5 for 60 seconds[DST IP]                        |
| Inactive      | 2018-06-26 16:51:47 | Automation        | Realtime   | Watering Hole Suspicious Web Script Detection                                                        | 1 for 4 seconds[]                               |
| Inactive      | 2018-06-26 16:48:37 | Automation        | Realtime   | IoT Bot suspected detected                                                                           | 2 for 10 seconds[SRC IP]                        |
| Inactive      | 2018-06-26 14:42:17 | Automation        | Realtime   | If a new type of attack that has not been detected for a month is detected month Web Shell attack de | 1 for 10 seconds[]                              |
| Inactive      | 2017-12-18 16:23:45 | Automation        | Background | Malware Port Access Detection Detection Same Destination IP UDP Flooding                             | Every Hour 1 hour ago                           |
| Inactive      | 2017-12-06 16:20:41 | Manual            | Realtime   | Repeat Attack - FW                                                                                   | 15 for 60 seconds[SRC IP]                       |
| Active        | 2017-11-27 15:52:19 | Manual            | Realtime   | Web Shell attack detected[Test]                                                                      | 1 for 10 seconds[]                              |
| Active        | 2017-11-16 16:52:54 | Manual            | Realtime   | include injection - test                                                                             | 1 for 1 second[]                                |
| Active        | 2017-10-20 16:30:52 | Manual            | Realtime   | UDP flooding is detected on the same destination IP                                                  | 5 for 60 seconds[DST IP]                        |
| Active        | 2017-09-19 16:42:31 | Manual            | Realtime   | Apache Struts Attack detected                                                                        | 1 for 4 seconds[]                               |
| Active        | 2017-09-07 16:41:21 | Manual            | Realtime   | MalwareIP Detected                                                                                   | 1 for 5 seconds[]                               |
| Inactive      | 2017-08-25 11:35:06 | Manual            | Background | Background Test-2                                                                                    | from 2017-09-06 16:00:00 to 2017-09-07 15:59:59 |
| Inactive      | 2017-08-17 10:39:18 | Manual            | Realtime   | Multiple attack detected for specific attacker IP[Test]                                              | 2 for 30 seconds[SRC IP]                        |
| Inactive      | 2016-11-10 05:00:19 | Manual            | Realtime   | [Actual use]Looking for undetected country for a week                                                | 1 for 1 second[SRC country]                     |

Fig. 12 New profile registration screen

Fig. 13 2017 OWASP TOP 5

| Risk                                         | OWASP 10 | Related Signature                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Feature                                                                                |
|----------------------------------------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Injection                                    | A1       | <ul style="list-style-type: none"> <li>- GNU bash Environment Variable Command Injection</li> <li>- Wordpress Wpdb_prepare SQL Injection</li> <li>- Trend Micro Control Manager SQL Injection</li> <li>- Schneider Electric U.motion Builder SQL Injection.B</li> <li>- HPE IMC wmiConfigContent Expression Language Injection</li> <li>- HPE IMC userSelectPagingContent EL Injection</li> <li>- GNU bash Environment Variable Command Injection</li> <li>- dotCMS categoriesServlet Blind SQL Injection</li> <li>- Sun Java Deployment Toolkit Argument Injection Vul</li> <li>- PHP-Nuke SQL Injection vulnerability</li> </ul>                                                                                      | Injection attack prevention                                                            |
| Broken Authentication and Session Management | A2       | <ul style="list-style-type: none"> <li>- PHPMailer mailSend() Remote Code Execution.C</li> <li>- Apache Struts2 DefaultActionMapper Remote Command Exe</li> <li>- WordPress User Photo Component Remote File Upload Vulnerability</li> <li>- Sasser Worm ftpd Remote Buffer Overflow Exploit (TCP-5554)</li> <li>- Oracle Java Applet Rhino Script Engine Remote Exe</li> <li>- Oracle Endeca Server createDataStore Remote Command Execution</li> <li>- Nagios Remote Plugin Executor Arbitrary Command Execution</li> <li>- MS XML Core Services Remote Code Execution Vul</li> <li>- MS Vector Markup Language Remote Code Execution</li> <li>- MS Outlook Express and Windows Mail Remote Code Execution</li> </ul> | Cookie Tampering protection, Cookie Proxying, Cookie Encryption, CSRF tagging, Use SSL |
| Cross-Site Scripting (XSS)                   | A3       | <ul style="list-style-type: none"> <li>- EPSON TMNet WebConfig XSS</li> <li>- D-Link PHP ActionTag XSS</li> <li>- Cacti spikekill.php XSS</li> <li>- Apache Struts2 XWork WebWork XSS.A</li> <li>- Apache Struts2 XWork WebWork XSS</li> <li>- Apache Struts2 Dynamic Method Invocation XSS</li> <li>- Apache Struts showConfig.action XSS</li> <li>- Apache Struts actionNames.action XSS</li> <li>- MS System Center Operations Manager Web Console XSS</li> <li>- MS SharePoint Server Callback Function XSS</li> </ul>                                                                                                                                                                                              | XSS Attack zPrevention                                                                 |
| Broken Access Control                        | A4       | <ul style="list-style-type: none"> <li>- Apache Struts2 ParametersInterceptor ClassLoader Sec Bypass.C</li> <li>- Apache Struts2 ParametersInterceptor ClassLoader Sec Bypass.B</li> <li>- Apache Struts2 ParametersInterceptor ClassLoader Sec Bypass.A</li> <li>- Apache Struts2 ParametersInterceptor ClassLoader Sec Bypass</li> <li>- Apache Struts Parameters Interceptor security bypass</li> <li>- Apache Struts 2 ParameterInterceptor Class OGNL Command Exe</li> <li>- Apache Struts CookieInterceptor Security Bypass(8080)</li> <li>- Apache Struts CookieInterceptor Security Bypass</li> </ul>                                                                                                           | Apache Struts security bypass vulnerability                                            |

conditions to several combinations. In the results verified to determine the accuracy of whether it is an actual well-known attack or not in the proposed system, over 85%

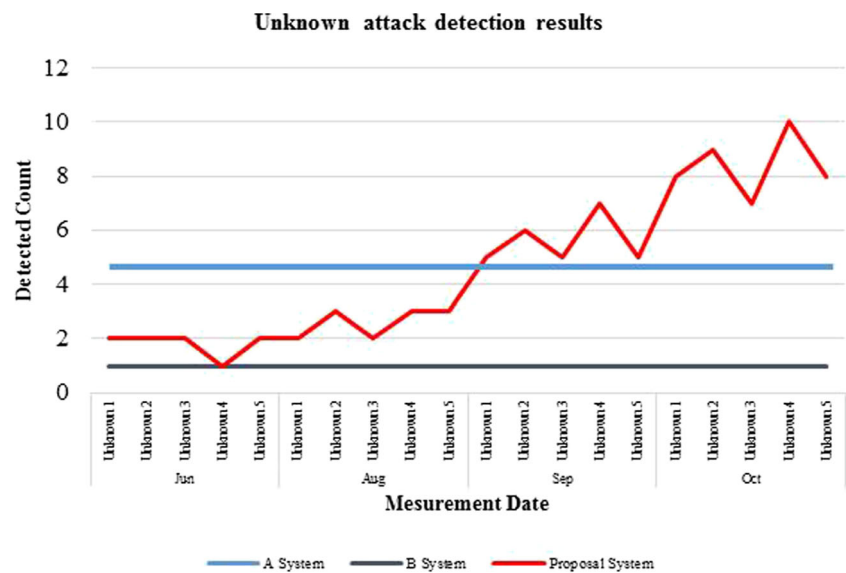
of the average number of cases in all category except unknown 3, were actually detected as unknown attacks. This means that the newly created profile can detect well-known

**Table 7** Comparison of existing system and proposed system after well known attack

| System   | A1 | A2 | A3 | A4 | A5 | Avg | TP  | FP | TPR |
|----------|----|----|----|----|----|-----|-----|----|-----|
| A        | 4  | 17 | 8  | 4  | 9  | 41  | 20  | 19 | 49% |
| B        | 12 | 7  | 3  | 6  | 15 | 43  | 22  | 21 | 51% |
| Proposal | 53 | 14 | 34 | 45 | 33 | 179 | 150 | 39 | 83% |

**Table 8** Comparison of existing system and proposed system after unknown attack

| Category  | System   | Jun | Aug | Sep | Oct | AVG | TP | FP | TPR |
|-----------|----------|-----|-----|-----|-----|-----|----|----|-----|
| Unknown 1 | A        | 1   | 1   | 1   | 1   | 4   | 0  | 4  | 0%  |
|           | B        | 1   | 1   | 1   | 1   | 4   | 0  | 4  | 0%  |
|           | Proposal | 2   | 2   | 5   | 8   | 17  | 15 | 2  | 88% |
| Unknown 2 | A        | 1   | 1   | 1   | 1   | 4   | 0  | 0  | 0%  |
|           | B        | 1   | 1   | 1   | 1   | 4   | 0  | 0  | 0%  |
|           | Proposal | 2   | 2   | 6   | 9   | 20  | 17 | 3  | 85% |
| Unknown 3 | A        | 1   | 1   | 1   | 1   | 4   | 0  | 4  | 0%  |
|           | B        | 1   | 1   | 1   | 1   | 4   | 0  | 88 | 0%  |
|           | Proposal | 2   | 2   | 5   | 7   | 16  | 12 | 4  | 75% |
| Unknown 4 | A        | 1   | 1   | 1   | 1   | 4   | 0  | 4  | 0%  |
|           | B        | 1   | 1   | 1   | 1   | 4   | 0  | 4  | 0%  |
|           | Proposal | 1   | 3   | 7   | 10  | 42  | 37 | 5  | 88% |
| Unknown 5 | A        | 1   | 1   | 1   | 1   | 4   | 0  | 4  | 0%  |
|           | B        | 1   | 1   | 1   | 1   | 4   | 0  | 4  | 0%  |
|           | Proposal | 2   | 3   | 5   | 8   | 18  | 16 | 2  | 88% |

**Fig. 14** Detection result after unknown attack

attacks. As a result, we could evaluate the performance of the proposed system with the automatically generated profile results through the second experiment. Also, we can evaluate whether the proposed auto generation profile can detect potential threats or effectively cope with zero-day attacks through experimental evaluation.

## 5 Conclusions

This paper dramatically improved the detection rate of unknown attacks by an average of four times over the analysis of existing profiling by extracting accuracy with collaborative filtering and recommending new profiles among similar sets of IPs after reinterpreting the profiling results. In order to verify the validity of this study, we used 100 kinds of malicious code traffic to improve the accuracy of the demonstration.

This research contributes to two major areas of business continuity planning (BCP) for the survival of enterprises and organizations. First, the profile is to detect and recognize the event log if it is detected by the conditions set by the administrator. On the contrary, there is a disadvantage that it is difficult to generate a profile and the response speed is slow unless the expert technique is fostered. The proposed model of this study does not need such technical expertise and can reduce the speed of response as much as possible. Also, it is possible to create a profile in only a few steps. Therefore, it proposes risk management measures for proactive response against unknown external hacking attacks, thereby presenting threat indexes for various types of unknown attacks to management, thereby supporting risk management for enterprise survival.

Second, it is possible to analyze the scenario of hacking threat by enabling cross-reference analysis between multiple profiles, and it is possible to reduce the analysis time in the case of infringement analysis and to be able to respond quickly. Therefore, management decision-making time is reduced to ensure business continuity of enterprises or organizations.

Currently, SIEM is designed as a semi-automatic setting so that the recommended profile conditions in real business can be changed by the administrator, but it will be automated to minimize administrator decision as reducing profile conditions.

**Acknowledgements** This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2017R1E1A1A01075110).

## References

1. <http://www.kr.idc.asia/press/pressreleasearticle.aspx?prid=483>

2. Butler B (2014) Gartner: Internet of Things has reached hype peak, 13:8. <http://www.networkworld.com/article/2464007/cloud-computing/gartner-internet-of-things-has-reached-hype-peak.html>
3. Andrey F, Igor K, Didier E (2017) Correlation of security events based on the analysis of structures of event types. *Intelligent Data Acquisition and Advanced Computing Systems*, IEEE
4. Gartner Magic Quadrant (2016) Magic Quadrant for Security Information and Event Management, 3
5. Reza S, Ali G (2006) Alert correlation survey: framework and techniques. In: *Proceedings of the 2006 international conference on privacy, security and trust*. Article no. 37
6. Jeon I-s, Han K-h, Kim D-w, Choi J-y (2015) Using the SIEM Software vulnerability detection model proposed. *J Korea Inst Inf Secur Cryptology*, pp 962–965
7. Idoia A, Sergio A (2012) Improving the automation of security information management: a collaborative approach. *IEEE Secur Privacy* 10(1):55–59
8. Alistair S (1998) Scenario-based requirements analysis. *Requirements Eng* 3(1):48–65
9. Manuel E, Theodoor S, Engin K, Christopher K (2012) A survey on automated dynamic malware-analysis techniques and tools. *J ACM Comput Survey*, pp 8–9
10. Alvaro A, Pratyusa K, Sreeranga P (2013) Big data analytics for security. *IEEE Secur Privacy*, pp 11–12
11. Matthias G, Michael F, Basel K, Adrian T, Ruth B, Alessandro M Anomaly Detection in the Cloud: Detecting Security Incidents via Machine Learning. *International Workshop on Eternal Systems: Eternals 2012: Trustworthy Eternal Systems via Evolving Software Data and Knowledge*. 103–116
12. Chen K-Y, Luesak L, Chou S-T (2007) Hot topic extraction based on timeline analysis and multidimensional sentence modeling. *IEEE Trans Knowl Data Eng*, pp 1023–1024
13. Canfora G, Ceccarelli M, Cerulo L, Di Penta M (2010) Using multivariate time series and association rules to detect logical change coupling: An empirical study. In: *IEEE international conference on software maintenance (ICSM)*, vol 2010
14. Yunhong Z, Robert W, Pan S (2008) Large-scale parallel collaborative filtering for the netflix prize. In: *International conference on algorithmic applications in management*, pp 337–348
15. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: *Proceeding WWW '01 Proceedings of the 10th international conference on World Wide Web*. ACM, New York, pp 285–295
16. Linden G, Smith B, York J (2003) [Amazon.com](http://www.amazon.com) recommendations: item-to-item collaborative filtering. *IEEE Internet Comput*, pp 78–79
17. Lops P, de Gemmis M, Semeraro G (2010) Content-based recommender systems: state of the art and trends, recommender systems handbook, pp 73–105
18. Fang S, Xu LD, Zhu Y, Ahati J, Pei H, Yan J, Liu Z (2014) An integrated system for regional environmental monitoring and management based on internet of things. *IEEE Trans Ind Inf* 10(2):1596
19. Cisco (2013) The Internet of Everything for Cities: Connecting People, Process, Data, and Things To Improve the 'Livability' of Cities and Communities, Cisco Point of View Report. [http://docbox.etsi.org/Workshop/2013/201306\\_SMARTCITIES/S03\\_SmartintheCity/CISCO\\_andriantsiferana.pdf](http://docbox.etsi.org/Workshop/2013/201306_SMARTCITIES/S03_SmartintheCity/CISCO_andriantsiferana.pdf)
20. Zhu J, Chan DS, Prabhu MS, Natarajan P, Hu H, Bonomi F (2013) Improving Web Sites Performance Using Edge Servers in Fog Computing Architecture. In *Service Oriented System Engineering (SOSE)*
21. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Ioe (IoE): a vision architectural elements and future directions. *Futur Gener Comput Syst* 29(7):1645–1660

22. (2016) Common vulnerability scoring system 3.0, Recommendation ITU-T X.1521, 03
23. IEEE-Standards (2016) Cherry Tom, security AND IOT IN IEEE standards

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Dongsik son** is a researcher at ISAA Lab., Department of Computer Engineering, Ajou University and vice president of GTEsecu corp, South Korea. He received the M.A. from Kunkuk University. His research interests include APT attacks, malware & exploit analysis, network security, incident handling, network forensics and applied security mechanisms for Cloud and Big Data system and so on.



**Seungpyo Huh** is a freedom researcher. He received the M.A. from Kyunggi University. His research interests include APT attacks, IOT analysis using AI and Bigdata.



**Taejin Lee** is a professor at Computer Engineering at Hoseo University. He received the Ph.D. from Ajou University. His research interests include malware analysis, network security and endpoint detection response.



**Jin Kwak** is a professor at Dept. Of Cyber Security at Ajou University, Korea. He received the Ph.D. degree from SKKU, Korea. His research interests include Cryptographic protocols, Applied security mechanisms for Cloud and Big Data system and so on.