Article

Big data log-based correlation analysis profiling auto generation model

Dongsik Sohn[1], Seungpyo Huh[2] , Taejin Lee[3], Jin Kwak[4]

[1] ISAA Lab., Department of Computer Engineering, Ajou University, Suwon, South Korea; over0033@gmail.com
[2] SOC, Wins., Seongnam, South Korea; hsp083@wins21.co.kr
[3] Department of Computer engineering, Hoseo University, Korea; kinjecs0@gmail.com
[4] Department of Cyber Security, Ajou University, Suwon, South Korea; jkwak.security@gmail.com
* Correspondence: jkwak.security@gmail.com; Tel.: +82-10-6773-9484

Abstract: 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, to precisely diagnose crisis to threats, and to provide an accurate alarm to an administrator by correlating collected information. However, it is difficult to quickly recognize and handle with various attack situations using a solution equipped with complicated functions during security monitoring. In order to overcome this situation, new detection analysis process has been required, and there is an effort to increase response speed during security monitoring and to expand accurate linkage analysis technology. In this paper, reflecting these requirements, we design and propose profiling auto-generation model that can improve the efficiency and speed of attack detection for potential threats requirements. we design and propose profiling auto-generation model that can improve the efficiency and speed of attack detection for potential threats.

Keywords: Big Data; SIEM; Correlation Analysis; Cyber Crime Profiling

1. Introduction

According to IDC's latest research report, the global big data and analytics market is expected to grow by 12.4% year-on-year to reach $ 150.8 billion, and investment in analytical solutions is getting increasing [1]. 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].

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. 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.

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.

 doi:10.20944/preprints201804.0144.v1

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 in the past. 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. However, as the number of data increases, the correlation coefficient value becomes larger and reliability may be lowered[3].

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 [4, 5].

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[6, 7].

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[8, 9].

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 [10, 11].

In addition, Yunhong et al. Claim that there are Collaborative Filtering model and Contents-based Filtering model in introducing profile recommendation technology [12], 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." [13][14] 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. [15] 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 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 Figure 1, ① the result of the profile is collected, ② after the formalization work, ③ it is constructing and modeling dataset according to the graph analysis and the page rank analysis, which is the core analysis model. ④ At this time, after interworking with

Security Intelligence information system and sort IP which has malicious activity, ⑤ the result is extracted, ⑥ recommend profile auto-generation based on collaborative filtering and ⑦finally perform an alert function to the administrator.

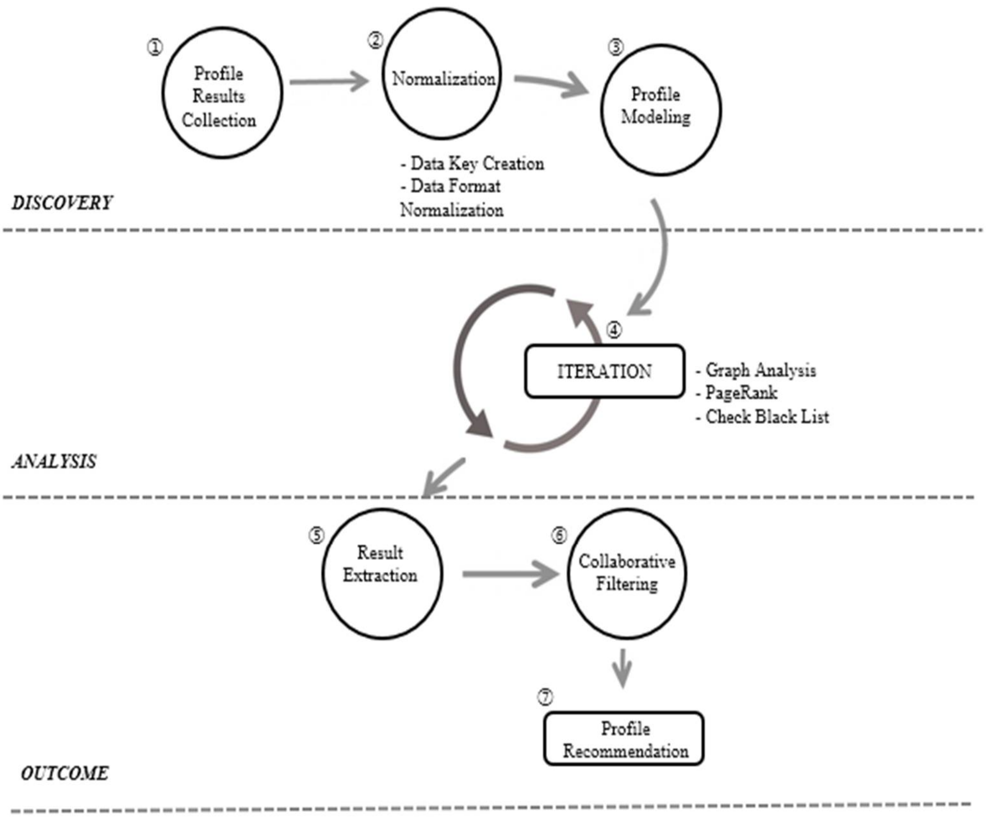
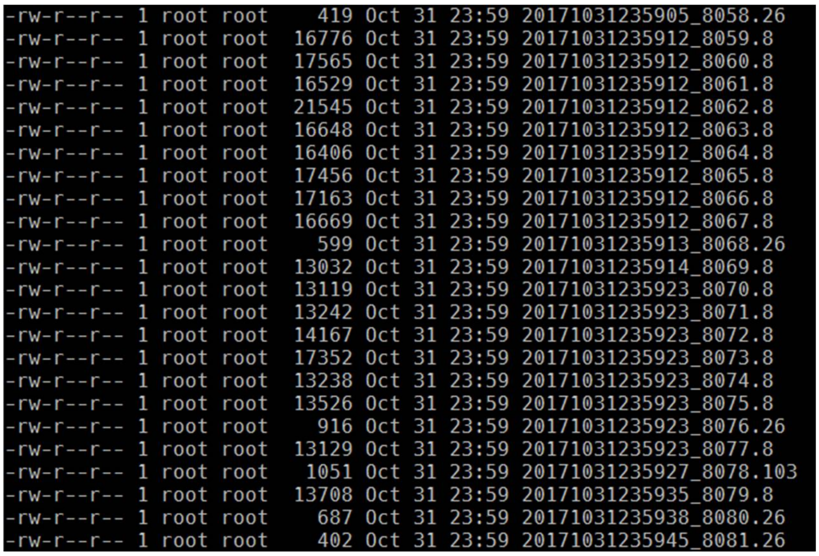


Figure 1. System Overview.

3.2. Profile Result Collecting - Data Acquisition

In the proposed system, if a profile meets various conditions based on heterogeneous security log, it is regarded as abnormal behavior and stored as a profile. The profile information is stored in the order of the detected time and has a filename such as a date and profile IP. 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 Figure 3. Therefore, the necessary information for modeling the profile is source IP and destination IP in Figure 3 and Profile ID in Figure 2.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 April 2018 doi:10.20944/preprints201804.0144.v1



```
-rw-r--r-- 1 root root   419 Oct 31 23:59 20171031235905_8058.26
-rw-r--r-- 1 root root 16776 Oct 31 23:59 20171031235912_8059.8
-rw-r--r-- 1 root root 17565 Oct 31 23:59 20171031235912_8060.8
-rw-r--r-- 1 root root 16529 Oct 31 23:59 20171031235912_8061.8
-rw-r--r-- 1 root root 21545 Oct 31 23:59 20171031235912_8062.8
-rw-r--r-- 1 root root 16648 Oct 31 23:59 20171031235912_8063.8
-rw-r--r-- 1 root root 16406 Oct 31 23:59 20171031235912_8064.8
-rw-r--r-- 1 root root 17456 Oct 31 23:59 20171031235912_8065.8
-rw-r--r-- 1 root root 17163 Oct 31 23:59 20171031235912_8066.8
-rw-r--r-- 1 root root 16669 Oct 31 23:59 20171031235912_8067.8
-rw-r--r-- 1 root root   599 Oct 31 23:59 20171031235913_8068.26
-rw-r--r-- 1 root root 13032 Oct 31 23:59 20171031235914_8069.8
-rw-r--r-- 1 root root 13119 Oct 31 23:59 20171031235923_8070.8
-rw-r--r-- 1 root root 13242 Oct 31 23:59 20171031235923_8071.8
-rw-r--r-- 1 root root 14167 Oct 31 23:59 20171031235923_8072.8
-rw-r--r-- 1 root root 17352 Oct 31 23:59 20171031235923_8073.8
-rw-r--r-- 1 root root 13238 Oct 31 23:59 20171031235923_8074.8
-rw-r--r-- 1 root root 13526 Oct 31 23:59 20171031235923_8075.8
-rw-r--r-- 1 root root   916 Oct 31 23:59 20171031235923_8076.26
-rw-r--r-- 1 root root 13129 Oct 31 23:59 20171031235923_8077.8
-rw-r--r-- 1 root root  1051 Oct 31 23:59 20171031235927_8078.103
-rw-r--r-- 1 root root 13708 Oct 31 23:59 20171031235935_8079.8
-rw-r--r-- 1 root root   687 Oct 31 23:59 20171031235938_8080.26
-rw-r--r-- 1 root root   402 Oct 31 23:59 20171031235945_8081.26
```

Figure 2. Profile Results Screen.

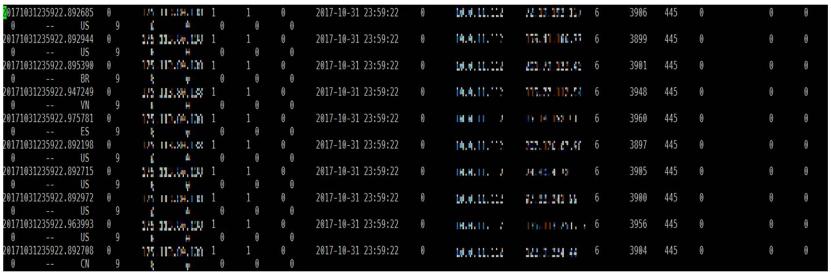


Figure 3. Profile Results Contents Information Screen

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 to the direction of an edge. As shown in Figure 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 1 shows the configuration profile items in Figure 4.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 April 2018 doi:10.20944/preprints201804.0144.v1

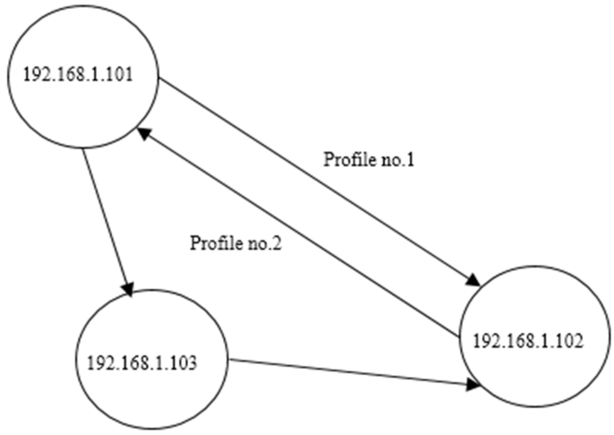


Figure 4. Profile Graph Network Configuration.

Table 1. Profile Collection Contents.

SRC 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

3.4. Graph Analysis – 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 Figure 4. Also, as shown in Figure 5, a profile is shown between IPs

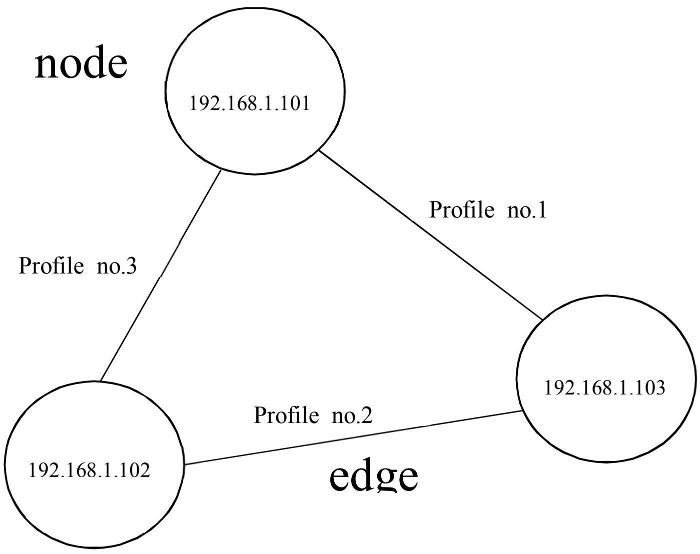


Figure 5. Graph model showing node and link relationship.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 April 2018 doi:10.20944/preprints201804.0144.v1

Table 2 shows the numerical information about the node, and a long value is sequentially given to generate a unique ID for IP.

Table 2. Node Scheme.

ID	Property
1	192.168.1.100
2	192.168.1.101
3	192.168.1.102

Table 3. Link Scheme.

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

In the link scheme of Table 3, 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 3 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 create nodes and edges. The pseudo-code that configures the vertex and edge is as follow as Figure 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.

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

Figure 6. Pseudocode to Generate Vertices and Edges.

Figure 7 shows the pseudo-code that performed graph-based PageRank analysis with Edge RDD and Vertex RDD values created in Figure 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_out(v)' is a degree toward the link of v.

$$\mathrm{PR(u)} = \frac{1-a}{N} + a\sum_{v} A_{vu}\, PR(v)/d_{out}(v) \tag{1}$$

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.

Function 2 – Page Rank Execution Function
Description. It create RDD with vertex and edge buffer information and it perform page rank algorithm after constructing graph. *graph : Graph* *eRDD : edge RDD* *vRDD : vertex RDD* *srcRank : Profile source IP Rank Buffer* *dstRank : Profile destination IP Rank Buffer* 1. graph <- Graph(vRDD, eRDD) 2. ranks <- graph.pagerank.vertices 3. result <- ranks.join(vRDD) 4. **while** k = 1 to result **do** 5. src <- graph.triplets.filter(t => t.srcAttr.equals(k)) 6. dst <- graph.triplets.filter(t => t.dstAttr.equals(k)) 7. result <- CheckBlackList(src, dst) 8. srcRank <- k, src, result 9. dstRank <- k, dst, result 10. **end**

Figure 7. Performs Page Rank Algorithm after Edge and Vertex RDD Generation.

3.5. Collaborative Filtering - Automatic Creation Recommendation

The proposed automatic creation recommendation model extracts the accuracy based on collaborative filtering and 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 4. Table 4 shows the dataset based on the source IP, destination IP, profile ID, and rank ratio created in Figure 8 through the PageRank reference model. Also, the datasheet is made by dividing into source IP and the destination IP as shown in table 4. The created dataset does not include profile ID and IP information. In Table 4, 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 April 2018 doi:10.20944/preprints201804.0144.v1

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'.

Table 4. 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

Function 3 – Auto Generation Recommendation Function

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.

1. *Get profile_IP, profile_ID*
2. ratings <- data.map(case Array(profile_IP, profile_ID, rate))
3. model <- ALS.train(rantings, rank, numIteration)
4. item <- ratings.map(case Rating(profile_ip, profile_id, rate)) => (profile_ip, profile_id)
5. prediction <- model.predict(item).map(case Rating(profile_ip, profile_id, rate)) => ((profile_ip, profile_id), rate)
6. rateandpreds <- ratings.map(case Rating(profile_ip, profile_id, rate)) => ((profile_ip, profile_id), rate).join(predictions)
7. MSE <- rateandpreds.map(case ((profile_ip, profile_id), (r1, r2)) => math.pow((r1 – r2), 2)).reduce(_+_) / rateandpreds.count

Figure 8. Pseudocode to Auto Generation Recommendation Model

4. Experimental Result

4.1. Experimental Environment

The experimental environment of the proposed system is shown in Table 5, 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 April 2018 doi:10.20944/preprints201804.0144.v1

Table 5. 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

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.

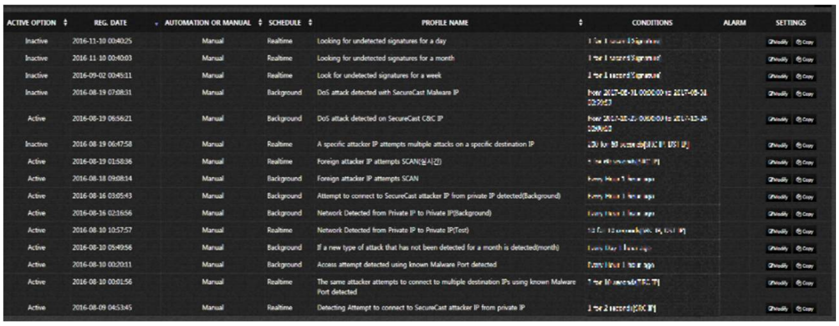


Figure 9. Proposed System Profile Set Screen.

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 Figure 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 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 April 2018 doi:10.20944/preprints201804.0144.v1

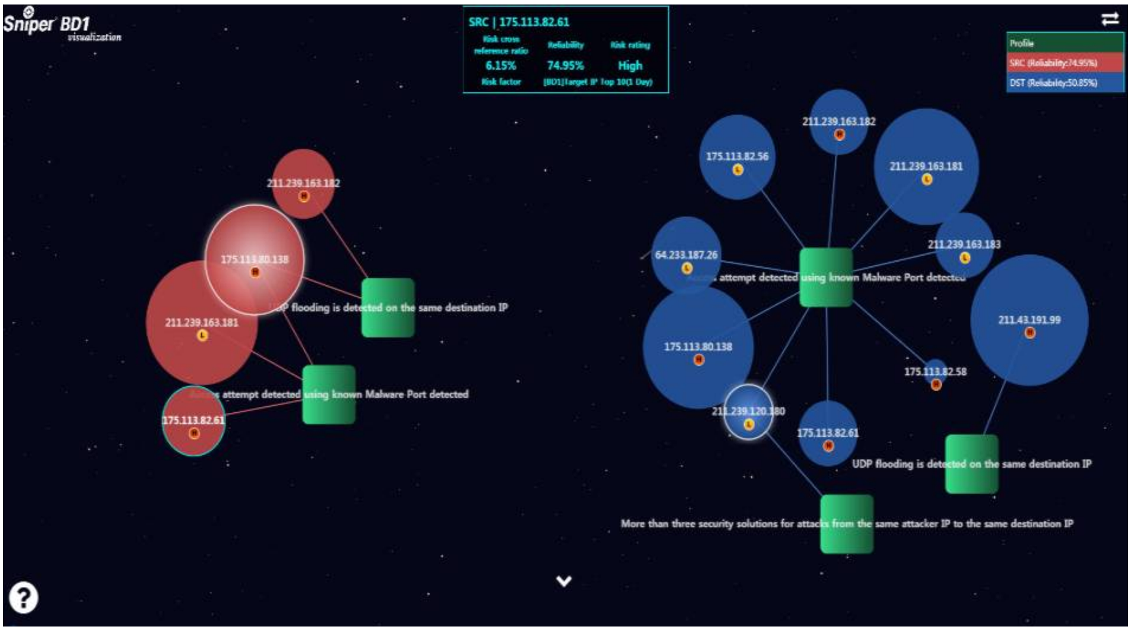


Figure 10. Profile Auto Generation Main Screen.

The reliability of Figure 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 Figure. 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 ① in Figure.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 ② screen
3. In ③ 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. ④ 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 Figure 12. First, the default setting is changed from 'unused' to 'enabled', The log detected by the new recommendation profile can be checked on the existing log inquiry screen.

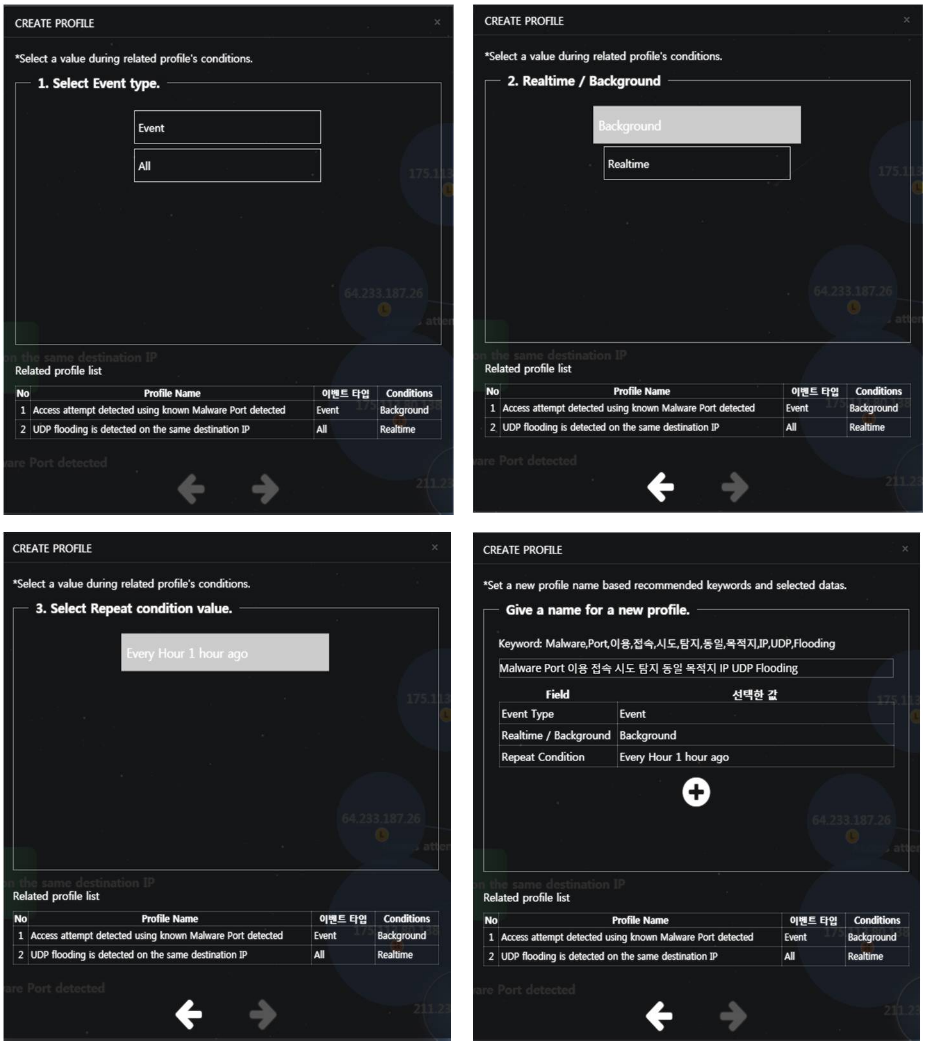


Figure 11. New Profile Registration Process Screen.

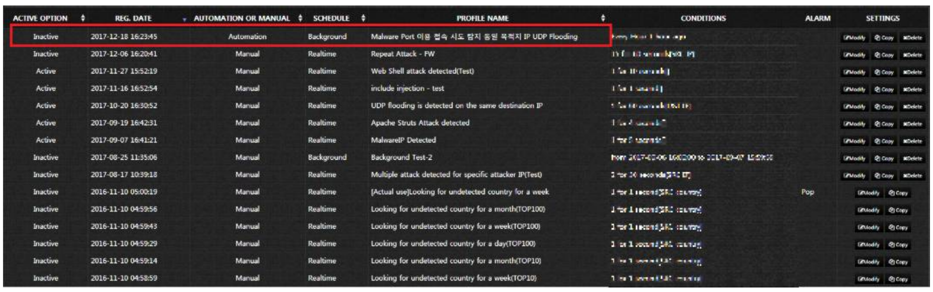


Figure 12. New Profile Registration Screen.

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 Table 6 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.

Table 6. 2017 OWASP TOP 5.

Risk	OWASP 10	Related Signature	Feature
Injection	A1	- GNU bash Environment Variable Command Injection - Wordpress Wpdb_prepare SQL Injection - Trend Micro Control Manager SQL Injection - Schneider Electric U.motion Builder SQL Injection.B - HPE IMC wmiConfigContent Expression Language Injection - HPE IMC userSelectPagingContent EL Injection - GNU bash Environment Variable Command Injection - dotCMS categoriesServlet Blind SQL Injection - Sun Java Deployment Toolkit Argument Injection Vul - PHP-Nuke SQL Injection vulnerability	Injection attack prevention
Broken Authentication and Session Management	A2	- PHPMailer mailSend() Remote Code Execution.C - Apache Struts2 DefaultActionMapper Remote Command Exe - WordPress User Photo Component Remote File Upload Vulnerability - Sasser Worm ftpd Remote Buffer Overflow Exploit (TCP-5554) - Oracle Java Applet Rhino Script Engine Remote Exe - Oracle Endeca Server createDataStore Remote Command Execution - Nagios Remote Plugin Executor Arbitrary Command Execution - MS XML Core Services Remote Code Execution Vul - MS Vector Markup Language Remote Code Execution - MS Outlook Express and Windows Mail Remote Code Execution	Cookie Tampering protection, Cookie Proxying, Cookie Encryption, CSRF tagging, Use SSL

Cross-Site Scripting (XSS)	A3	- EPSON TMNet WebConfig XSS - D-Link PHP ActionTag XSS - Cacti spikekill.php XSS - Apache Struts2 XWork WebWork XSS.A - Apache Struts2 XWork WebWork XSS - Apache Struts2 Dynamic Method Invocation XSS - Apache Struts showConfig.action XSS - Apache Struts actionNames.action XSS - MS System Center Operations Manager Web Console XSS - MS SharePoint Server Callback Function XSS	XSS Attack zPrevention
Broken Access Control	A4	- Apache Struts2 ParametersInterceptor ClassLoader Sec Bypass.C - Apache Struts2 ParametersInterceptor ClassLoader Sec Bypass.B - Apache Struts2 ParametersInterceptor ClassLoader Sec Bypass.A - Apache Struts2 ParametersInterceptor ClassLoader Sec Bypass - Apache Struts Parameters Interceptor security bypass - Apache Struts 2 ParameterInterceptor Class OGNL Command Exe - Apache Struts CookieInterceptor Security Bypass(8080) - Apache Struts CookieInterceptor Security Bypass	Apache Struts security bypass vulnerability
Security Misconfiguration	A5	- OpenSSL X.509 IPAddressFamily Extension Parsing Error DoS.A - OpenSSL X.509 IPAddressFamily Extension Parsing Error DoS - OpenSSL TLS Heartbeat Extension Memory Disclosure - OpenSSL TLS Heartbeat Extension Memory Disclosure - Linux ftpd SSL Buffer Overflow (TCP-21)	PCI reports, SSL features

In the first experiment, samples were applied to the A-system as a rule-based analysis system, the B-system as a correlation-based system, and the proposed system. 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 Table 6 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 April 2018 doi:10.20944/preprints201804.0144.v1

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.

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%

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 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 Figure 13. This means that the generated profile condition is to detect the variant of the unknown attack effectively by increasing the detection 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 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.

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	3	6	9	20	17	3	85%
Unknown 3	A	1	1	1	1	4	0	4	0%
	B	1	1	1	1	4	0	4	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%

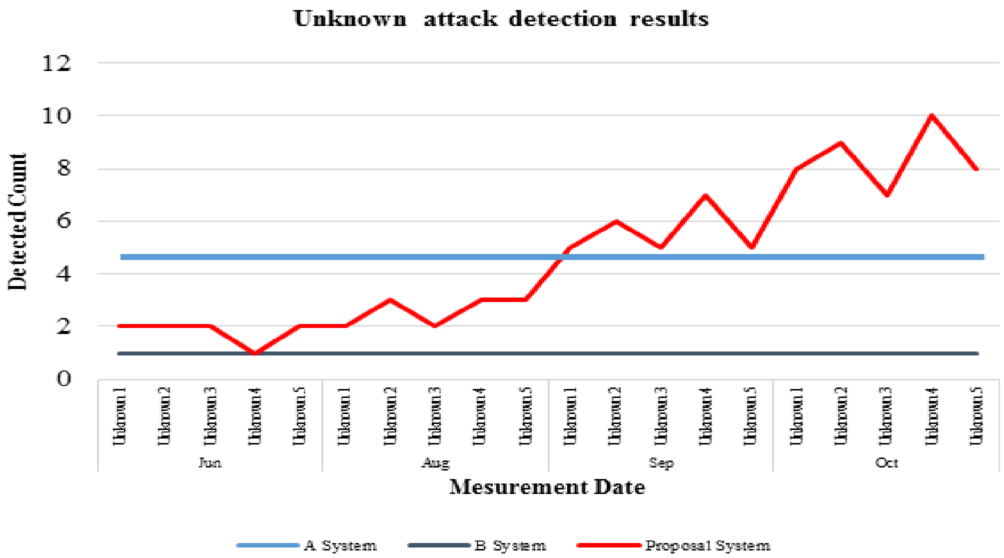


Figure 13. Detection Result after Unknown Attack

5. Conclusions

The profile function of the SIEM solution is to detect and recognize the event log detected by the conditions set by the administrator. On the contrary, there is a disadvantage that it is difficult to generate a profile and response speed is slow unless the expert technology is cultivated. 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. In addition, it is possible to do scenario analysis in hacking threats through cross-reference analysis between several profiles and, it reduces analysis and response time in terms of infringement accident analysis. 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.

6. Acknowledgment

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

Conflicts of Interest: The authors declare no conflict of interest.

References

1. http://www.kr.idc.asia/press/pressreleasearticle.aspx?prid=483
2. Gartner Magic Quadrant. Magic Quadrant for Security Information and Event Management. 2016, 3.
3. Reza, S.; Ali, G. Alert Correlation Survey: Framework and Techniques. *Proceedings of the 2006 International Conference on Privacy, Security and Trust*. Article No, 37, Oct, 2006.
4. Andrey, F.; Igor, K.; Didier, E. Correlation of security events based on the analysis of structures of event types, Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), *2017 9th IEEE International Conference on: Bucharest*, Romania.

 doi:10.20944/preprints201804.0144.v1

5. Jeon, I.-S.; Han, K.-H.; Kim, D.-W; Choi, J.-Y. Using the SIEM Software vulnerability detection model proposed. *Journal of The Korea Institute of Information Security & Cryptology*. VOL.25, NO.4, Aug, 2015
6. Idoia, A.; Sergio, A. Improving the Automation of Security Information Management: A Collaborative Approach. *IEEE Security & Privacy*, Vol. 10, issue 1, pp, 55-59, Jan,-Feb, 2012
7. Alistair, S. Scenario-based requirements analysis. Requirements Engineering. Vol 3, issue 1, pp 48–65, March 1998.
8. Manuel, E.; Theodoor, S.; Engin, K.; Christopher, K. A survey on automated dynamic malware-analysis techniques and tools. *Journal ACM Computing Surveys*, Vol 44, issue 2, Feb, 2012.
9. Alvaro, A.; Pratyusa, K.; Sreeranga, P. Big Data Analytics for Security. *IEEE Security & Privacy*, Vol. 11, issue 6, Nov-Dec, 2013.
10. Matthias, G.; Michael, Felderer.; 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*. pp 103-116.
11. Chen, K.-Y.; Luesak, L.; Chou, S.-T. Hot Topic Extraction Based on Timeline Analysis and Multidimensional Sentence Modeling. *IEEE Transactions on Knowledge and Data Engineering*. Vol. 19, issue 8, Aug, 2007.
12. 12. Gerardo Canfora, Michele Ceccarelli, Luigi Cerulo, Massimiliano Di Penta, "Using multivariate time series and association rules to detect logical change coupling: An empirical study", *Software Maintenance (ICSM), 2010 IEEE International Conference on*, Sep, 2010
13. 13. Yunhong, ZhouDennis, WilkinsonRobert, SchreiberRong Pan, "Large-Scale Parallel Collaborative Filtering for the Netflix Prize", *International Conference on Algorithmic Applications in Management*, pp. 337-348. 2008
14. 14. Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl, "Item-based collaborative filtering recommendation algorithms", *Proceeding WWW '01 Proceedings of the 10th international conference on World Wide Web*, pp. 285-295. ACM New York, NY, USA ©2001
15. G. Linden, B. Smith, J. York, "Amazon.com recommendations: item-to-item collaborative filtering", *IEEE Internet Computing*, Vol. 7, issue 1, Jan.-Feb. 2003