SECURITY INFORMATION EVENT MANAGEMENT WITH LATENT SEMANTIC ANALYSIS TECHNIQUE FOR THREAT IDENTIFICATION

PAVARIT DAIRINRAM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE (COMPUTER SCIENCE) FACULTY OF GRADUATE STUDIES MAHIDOL UNIVERSITY 2014

Thesis
entitled

SECURITY INFORMATION EVENT MANAGEMENT WITH LATENT SEMANTIC ANALYSIS TECHNIQUE FOR THREAT IDENTIFICATION

……..……..…..……
Mr. Pavarit Dairinram
Candidate

……..……..…..……...…
Assoc. Prof. Damras Wongsawang,
Ph.D. (Information Engineering)
Major advisor

……..……..…..……
Asst. Prof. Vasaka Visoottiviseth,
Ph.D. (Computer Engineering)
Co-advisor

……..……..…..……...…
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

……..……..…..……
Asst. Prof. Sudsanguan Ngamsuriyaroj,
Ph.D.
Program Director
Master of Science Program in
Computer Science
Faculty of Information and
Communication Technology,
Mahidol University

Thesis
entitled

SECURITY INFORMATION EVENT MANAGEMENT WITH LATENT SEMANTIC ANALYSIS TECHNIQUE FOR THREAT IDENTIFICATION

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Science (Computer Science)
on
July 18, 2014

……
Mr. Pavarit Dairinram
Candidate

……
Lect. Panomporn Suvannapattana,
Ph.D. (Mobile Communication)
Chair

……
Assoc. Prof. Damras Wongsawang,
Ph.D. (Information Engineering)
Member

……
Asst. Prof. Vasaka Visoottiviseth,
Ph.D. (Computer Engineering)
Member

……
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

……
Assoc. Prof. Jarernsri L. Mitrpanont, Ph.D.
Dean
Faculty of Information and
Communication Technology
Mahidol University

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to Assoc. Prof. Damras Wongsawang, my research advisor, for his patient guidance, invaluable support, and enthusiastic encouragement throughout this study. I am also particularly thankful to my advisory committee Asst. Prof. Vasaka Visoottiviseth for the useful critiques of this research work. My sincere thanks go to Lect. Panomporn Suvannapattana, for being the chairman and external examiner committee for my thesis defense examination.

My appreciation also goes to the undergraduate student in Computer Science track who supported for the data set for this research. Special thanks are extended to my lecturers, especially Asst. Prof. Dr. Charnyote Pluempitiwiriyawej and faculty members of ICT who support the facility and my research activities.

Finally, I would like to extend my deepest gratitude to my family for their love, understanding, and encouraging throughout my study and also to my lecturer, Lect. Pagaporn Pengsart who gave me the opportunity to pursue my dreams, and thank you for her support.

Pavarit Dairinram

SECURITY INFORMATION EVENT MANAGEMENT WITH LATENT SEMANTIC ANALYSIS TECHNIQUE FOR THREAT IDENTIFICATION

PAVARIT DAIRINRAM 5237679 ITCS/M

M.Sc. (COMPUTER SCIENCE)

THESIS ADVISORY COMMITTEE: DAMRAS WONGSAWANG, Ph.D., VASAKA VISOOTTIVISETH, Ph.D.

ABSTRACT

Security in a heterogeneous and complex network is one of the most significant challenges for administrators. A lot of devices are needed to handle and perform the protection and prevention in order to secure the network resources and assets from the threats, which are growing rapidly. The Security Information and Event Management (SIEM) is the major tool that helps administrators attend to the current situation. It is deployed to manage and identify the threats. Besides these, it is able to initiate the actions for protection and prevention of the network and also generate a report, which is conforms to the security standard. On the other hand, the amount of data from devices is significantly large, and the variation of threats is also a major concern for identifying them. To mitigate these problems, Latent Semantic Analysis (LSA) was proposed in this research. LSA is one of the most powerful tools that can provide efficiency in the exact matching, commonly used in information retrieval. LSA improves its performance by reducing the amount of unnecessary data generated from network devices. Additionally, it can be used to identify a similar threat pattern from the similarity between events and threats. The experiments showed that the LSA approach could help improve the threat identifying process by eliminating an amount of unnecessary data without a degradation in accuracy.

KEY WORDS: SECURITY INFORMATION EVENT MANAGEMENT / LATENT SEMANTIC ANALYSIS / THREAT IDENTIFICATION / NETWORK SECURITY

89 pages

การจัดการสารสนเทศและเหตุการณ์ของความมั่นคงด้วยเทคนิคการวิเคราะห์ความหมายแฝง สำหรับการระบุภาวะคุกคาม

SECURITY INFORMATION EVENT MANAGEMENT WITH LATENT SEMANTIC ANALYSIS TECHNIQUE FOR THREAT IDENTIFICATION

ปวริศ ด้ายรินรัมย์ 5237679 ITCS/M

วท.ม. (วิทยาการคอมพิวเตอร์)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์ : ดำรัส วงศ์สว่าง, Ph.D., วัสกา วิสุทธิวิเศษ, Ph.D.

บทคัดย่อ

ระบบความมั่นคงในระบบเครือข่ายที่มีความหลากหลาย และ สลับซับซ้อนถือเป็นอุปสรรค์สำหรับ ผู้ดูแลระบบในการรักษาความมั่นคงต่อ ระบบเครือข่ายและ สินทรัพย์ภายในระบบเครือข่าย ผู้ดูแลระบบต้องทำการบริหารจัดการ อุปกรณ์ภายในเครือข่ายให้มีความสามารถป้องกันและป้องปรามจากภาวะคุกคามต่าง ๆ ที่มีจำนวนมากได้ อุปกรณ์จัดการสารสนเทศและเหตุการณ์ของความมั่นคง หรือ SIEM เป็นอีกอุปกรณ์หนึ่ง ที่อำนวยความสะดวกให้ผู้ดูแลระบบสามารถแก้ไขปัญหา และบริหารจัดการ อีกทั้งช่วยระบุภัยคุกคามที่กำลังโจมตี และเสนอแนะแนวทางการปฏิบัติ พร้อมการดำเนินการป้องกันและป้องปราม หลังจากที่ทำการระบุภัยคุกคามได้อย่างถูกต้อง อีกทั้งสร้างรายงาน ที่อ้างอิงตามมาตรฐานความมั่นคง ต่อผู้ดูแลระบบได้ ทั้งนี้ข้อมูลที่ได้จากอุปกรณ์ต่างๆ ในเครือข่ายนั้นอาจมีขนาดใหญ่ รวมไปถึงการผันแปรของภัยคุกคาม ประเด็นเหล่านี้อาจส่งผลต่อเวลาและความถูกต้องในการระบุภัยคุกคามได้ ดังนั้น การใช้เทคนิคการวิเคราะห์ความหมายแฝง หรือ LSA ได้ถูกนำเสนอในการวิจัยนี้ เพื่อบรรเทาปัญหาข้างต้นลง โดยเทคนิคนี้ช่วยให้ลด จำนวนของข้อมูลที่ไม่จำเป็นออก เพื่อให้การวิเคราะห์นั้นมีประสิทธิภาพที่ดีขึ้น อีกทั้งเทคนิคนี้สามารถวิเคราะห์คล้ายคลึงของภัยคุกคามที่มีความเป็นไปได้จากข้อมูลของภัยคุกคามและข้อมูลเหตุการณ์ที่มีอยู่ในฐานข้อมูล โดยจากการทำลองในการวิจัยนี้พบว่า การใช้เทคนิค LSA ในอุปกรณ์ SIEM นั้นช่วยในการปรับปรุงขั้นตอนการระบุภัยคุกคามโดยการลดจำนวนของข้อมูลที่ไม่จำเป็นออกไป โดยที่การระบุยังคงความแม่นยำ เช่นเดียวกับก่อนการลดจำนวนของข้อมูล

89 หน้า

CONTENTS

CONTENTS (cont.)

CONTENTS (cont.)

LIST OF TABLES

LIST OF TABLES (cont.)

LIST OF TABLES (cont.)

LIST OF FIGURES

LIST OF FIGURES (cont.)

CHAPTER I
INTRODUCTION

Currently many businesses have implemented the ICT infrastructure. They align their network environment as distributed and heterogeneous, which are secured and always available. Also all assets in their network, such as information, application, devices or other components that support their business, the network administrator needs to ensure only the authorized users are able to access to their assets.

The networks are attacked by threats everyday and can harm the business. There are many types of threats that can harm their assets, For example, malicious code, hacker and cracker, etc. The solutions of this issue, is installing the network security tools and network security devices in order to prevent and protect their assets. Other the network security tools that help the network administrator such as Network Management Software (NMS), Security Information and Event Management software (SIEM), etc., can be deployed for monitoring the network infrastructure. Next, the network security devices are used for monitoring the real-time traffic and send the alert when they observe anomaly types by inspecting traffic flow, for instance, Intrusion Detection System (IDS), Intrusion Protection System (IPS), Proxy Server, etc.

The network administrator needs to collect all data from both network security tools and network security devices in order to identify the threat while network is being attacked by some threats. After they identified it, the network will be protected with the right action. However, the most networks are very complex. They need to implement a lot of network security tools and network devices inside the network for threat identifying. Therefore, they may generate huge of data for identification.

The SIEM tool employs many techniques for threat identifying. There are three techniques are applied such as AI techniques, Model Traversing technique, and Fault Propagation Models. The most common technique used in SIEM tool is the

rule-based technique. By this technique, the network administrator will create the patterns of the threats or signatures of threats as the rules and keep them inside the SIEM repository. After that, SIEM will collect the alerts, reports, logs, etc. from many devices and compare with the known threats' patterns. However, not all of them can be identified correctly. There are some it may identify the wrong threats (False Positive). There is various research works supporting our research. The aim of this research is to reduce the time for threat identification and get more an accurate result.

1.1 Motivation

The several of threats are attacking the network every single minute of the day. Also the time for identification and accurate results are the most significant key for protecting and preventing business assets that may harm by the threat. The research needs to approve the identification technique more accurate and faster.

1.2 Problem statement

The problem issue in this research is how we can identify the threat in a network environment with the large size of information, which collected from devices in network. We have to study more in details to find out an appropriate technique one that performs more accurate and faster for identification.

1.3 Objectives

1. To study and implement the Latent Semantic Analysis technique in the SIEM tool.

2. To compare and evaluate other techniques in SIEM tool.

1.4 Scopes

1. Study and Implement only Correlation function in SIEM

2. Implement on dataset that collected from closed network and simulated dataset.

1.5 Organization of the Thesis

This thesis is organized into 6 chapters as follows.

Chapter 1: Introduction

This chapter gives a brief of SIEM, Threat identification, and LSA. And it presents the motivation, the objectives, the scope and the organization of the thesis.

Chapter 2: Background

This chapter describes the background of SIEM, Threat identification Latent Semantic Analysis, and Vector Space Model.

Chapter 3: Literature Review

This chapter describes the related work of Threat Identification and existing technique that are using in a Network Security tool.

Chapter 4: Methodology

This chapter describes methodology of this research, data, and the analysis of SIEM and LSA features in details.

Chapter 5: Experimental Results

This chapter shows the experimental results of our research.

Chapter 6: Conclusion

This chapter analyzes and compares the results of an experiments to get a conclusion.

CHAPTER II
BACKGROUND

This chapter reviews current of threat identification, SIEM, Latent Semantic Analysis, and Vector Space Model.

2.1 Threat Identification

There are four main categories of threat identification, such as AI Techniques, Model Traversing Techniques, Fault Propagation Models, and the Information Retrieval technique. However, only three of them are commonly used. Figure 2.1 shows taxonomy of threat identification.

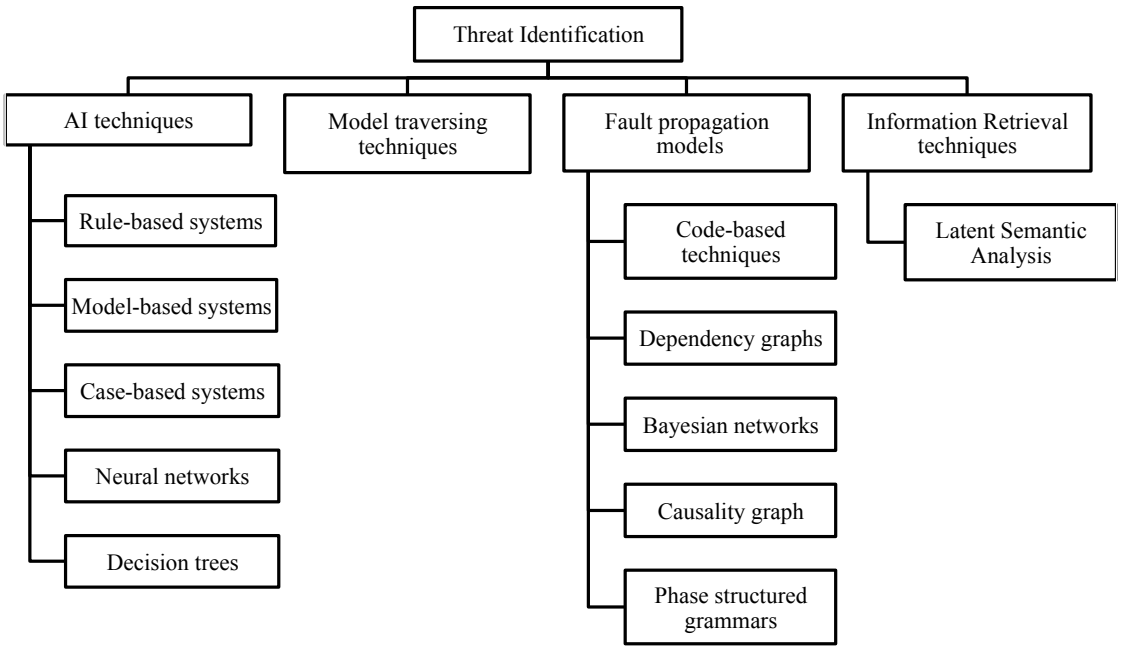

Figure 2.1: Taxonomy of Threat Identification Techniques

2.1.1 AI Techniques

It is the most widely used for this type is an Expert System. The Expert system needs to input the knowledge from human experts into the system. The knowledge is represented in the expert system as Rules, Logics, Semantics Nets, etc., Moreover; the knowledge may come from the experience of the human experts. Inside the expert system, there is the part that uses AI terminology for determined the result, which is an Inference Engine. There are five types of an inference engine that uses inside the expert system. The first technique is Rules-based system. It derives rules from the knowledge. After querying, it will search for the matched rule and gives the answer back. The second, Model-based systems, the rule-based may not the deep analysis in some environment that structured and functioned. It needs to perform all of analysis in order to find out the cause of the problem that may vary from application to application, for example, Industrial, Networks, and Aerospace, etc. This technique will map the model from the real world into the abstract model. This includes the communication, processing, and signal that are associated. After users query it will give the answer by finding the associated path that passed through any element in a model. Then it finds the root element that causes the problem. The third technique is case-based systems; this technique will keep the previous causes and symptoms that occurred. After user query, it will find the cause that is matched symptoms to a user. The forth, the neural network, this technique performs calculations of traditional neural network method to find the answer. The fifth technique is decision trees. It is represented as graphs that guide the users to observe symptoms. While observing, there is some weights or cost in each node used to guide the users reaching to the root cause or goal node.

2.1.2 Model Traversing Technique

It represents the network topology and defines a relationship of the network devices. All connectivity and operations are also defined in the topology. All deep information as well defines in this topology, for instance, the failure condition that related to other components or which components that generated the alarms/events. After users query, it will find all possible components that have the dependency from the initiated alarm/event of components.

2.1.3 Fault Propagation Models

This technique relies on a graphical model of the system. The connections of nodes in a graph represent the failure condition or alarm/event from a component is related to other components that failed with another condition or made another alarm/event. Many graphical approaches may apply in this technique. There are five types of graphical techniques that widely use. The first technique is the Code-base system. It represents the relationship of the problem-symptom as a codebook. Then it finds which problem is the root cause by eliminating some problems and symptoms that are not related to the root cause. Next, the dependency graph, it is a directed graph. It represents the components and weights the edges that describe cause-effect relationship. Third technique is Bayesian networks. This technique is represented as directed graph. However, the traversing is based-on the probability. Next technique is a causality graph; it is directed and an acyclic graph. It performs analysis of causal relations between problems. After users query, it finds the nodes relating to symptoms, and maps these symptoms corresponding to problems in a graph. The last technique is the Phrase structured grammars. It represents a hierarchical graph, which borrows the concept of structure natural language to determine the cause of problem.

2.1.4 Information Retrieval technique

This technique uses the concept of the Information Retrieval or IR domain. The IR domain performs a searching technique inside the data or corpus. The identification technique will be applied with the similar to the method used to search or identify the threat. The common technique that use in IR is the vector space model that is perform the matching by using the similarity value of words and query word. The Latent Semantic Analysis is the technique of this research applying the Vector Space model technique and the Single Value Decomposition. The more information of these two approaches is later described in this chapter.

2.2 Security Information and Event Management

The Security Information and Event Management (SIEM) is the tools for collecting, threat correlating, managing an incident, and generating the report. It is working with another two systems, which are Security Information Management (SIM) and Security Event Management (SEM). The SIM is the tool that concerns the interested information from historical data in any assets, for example, the log of operating system, log from network elements, log of application, etc. However, SEM tool focuses on the real-time activities in a network such as the alerts from the network devices.

There are three layers in the system: Asset Layer, Data Processing Layer, and Identification Layer. All three layers are described below.

1) Asset Layer is the layer that contains the elements in the system, which is sensitive and effect to the business, for example, Radius Server, Anti-Virus, Firewall, IDS, NMS (Network Management System), Workstation, Servers, and Network devices.

2) Data Processing Layer is the layer that consists of SIM and SEM. The collector will collect the data from the elements in the Asset Layer. It collects information from many different protocols and format such as SNMP, CIM, or Web Service. After that, the collector will divide the data into SIM and SEM. The Security Threat Manager uses this layer to find and keep the threat patterns in this layer.

3) Identification Layer is the layer for SIEM. After identifying the threat, it will generate some actions such as batch file and report which is compliance to the security standard, for instance ISO 27000 family, HIPPA, and FISMA, etc. Figure 2.2 shows the overview of SIEM system.

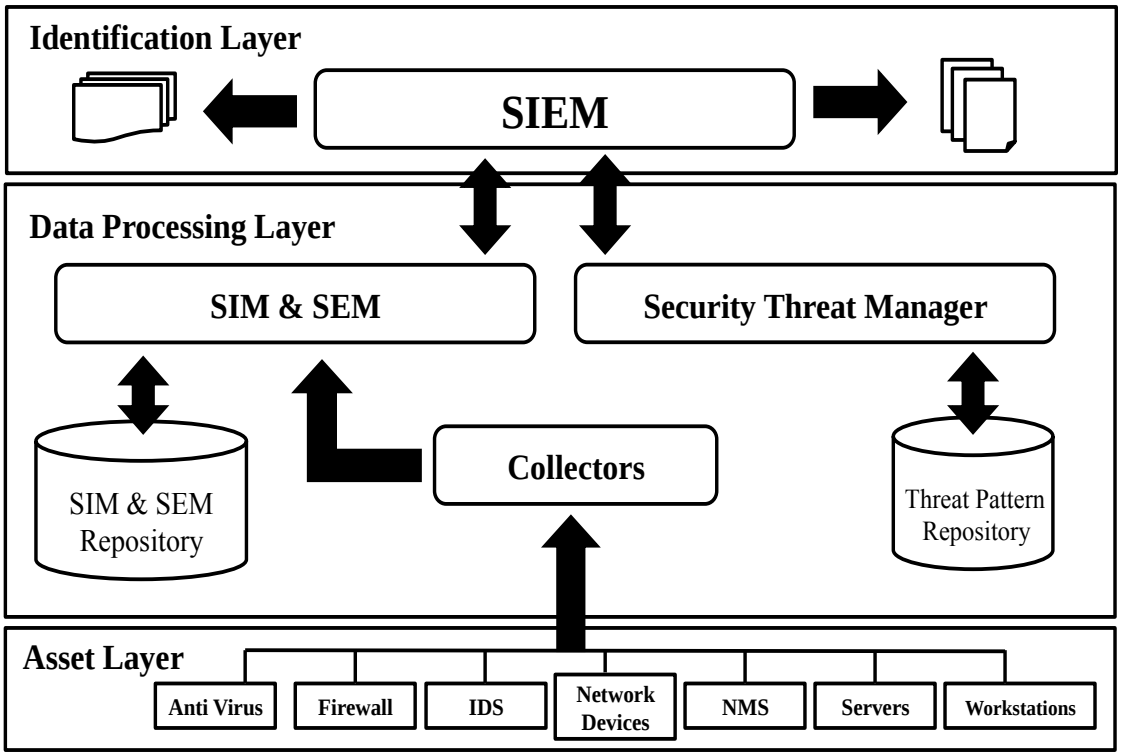

Figure 2.2: SIEM Overview

Inside SIEM, there are four functions, which are Log Consolidation, Threat Correlation, Incident Management, and Reporting.

1) Log Consolidation – Gather the raw data from SIM and SEM and normalize the data into the equivalent format. This process of this function is to find the same attribute of all data. After that they will be defined as the common attributes. Then any data corresponding to the common attributes will be changed the current attributes into common attributes.

2) Threat Correlation – Collect the normalized data from collector and find the matching threat that has the same behavior. All threat pattern needs to have a same common attribute like the normalized data.

3) Incident Management – Create actions after identified the threat such as notification, create a trouble ticket, or an automate response (e.g. batch file) in order to prevent and protect the assets.

4) Reporting – Generate the report which is compliance to the security standard to the administrator. Figure 2.3 shows the SIEM process structure.

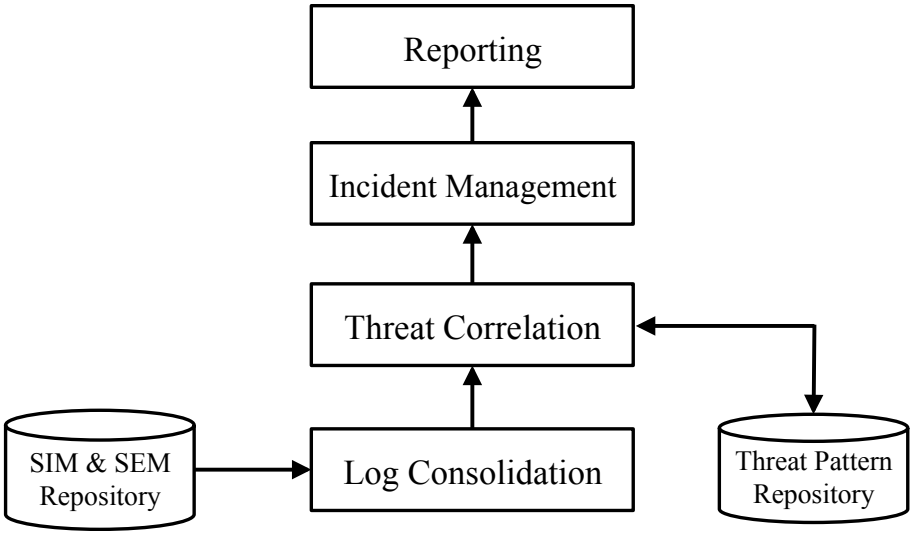

Figure 2.3: SIEM Structure

2.3 Latent Semantic Analysis

The Latent Semantic Analysis (LSA) or Latent Semantic Indexing (LSI) is developed for the Information Retrieval area. It uses the vectorial semantics to analyze the relationships between documents and terms. It is used for comparing texts in the relevant documents and query entered by users. The Singular Value Decomposition (SVD) is used for calculating the relevant.

The details of common steps are explained below:

1) Creating a co-occurrence matrix of Document and Terms. This matrix represents the occurrence value by counting the number of times Term 'Y' occurs in Document 'X'. In general, the matrix contains a lot of '0' values. It is Sparse Matrix. The matrix contains an unwanted data and needs to be eliminated in order to reduce the number of dimensions of a matrix.

2) Weighting the values. The sparse matrix needs to find the significant terms. This step uses some techniques to assign a weight to the terms such as Term Frequency – Inverse Document Frequency (TF-IDF). It uses to assign the weight for the most significant terms higher than the common terms.

3) Calculating the SVD. This step will reduce the number of dimension for Documents and Terms. This step takes be taking an advantage for reducing the memory and time while implementing the LSA.

4) Ranking the result. In this step, user will get the terms and documents relationship that relevant to the query from users. It is shown as a rank from the highest value to the lowest value.

2.3.1 Single Value Decomposition

The singular value decomposition of a matrix is the factorization of matrix into the product of three matrices as described by the equation below.

$$A = U\Sigma V^T$$

where A is the t x d matrix, U is the t x m orthonormal matrix, Σ is m x m diagonal matrix, and V is the m x d m orthogonal matrix.

After applied the Equation, some rows of U, Σ, and V are removed for eliminating the unnecessary data. The number of columns defines as value of k. It uses for reduce the number of dimension in the matrix. It may be defined as one dimension at least. Figure 2.4 shows the dimension reduction.

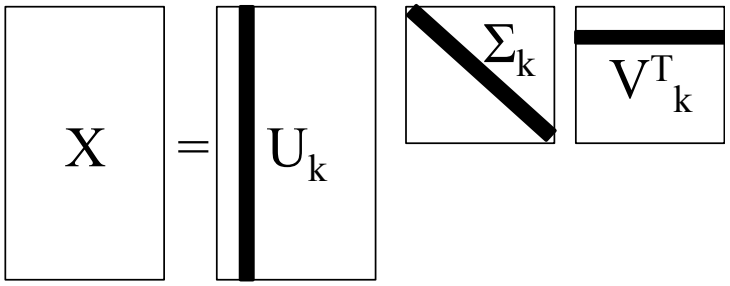

Figure 2.4: Three Matrices after SVD Calculation

The matrix size before compressing by using SVD actually is A= mxn. There are some noise or unwanted data inside the matrix and they need to be eliminated out.

After compressing the matrix by SVD, the matrix size will be smaller than A = m x n. The SVD use only at least 1 or 2 dimension. For instance, the data set kept

in matrix with size 100,000 x 100,000. After compression with SVD, the minimum size of matrix may be 1 x 100,000 or 2 x 100,000.

However, the matrix, after compression with SVD, always represent the values in different from the original matrix. Some data may be lost after compression. Thus, before applying use SVD, it is needed to consider how many dimensions that will not affect to the accuracy of data. Figure 2.5 depicts the reduction process.

Figure 2.5: The Compression of Matrix by SVD

2.4 Vector Space Model

It is a technique in Information Retrieval domain for comparing the similarity between documents and words in corpus. Figure 2.6 below describes the terms which appears inside many documents. Also document may have the same terms inside.

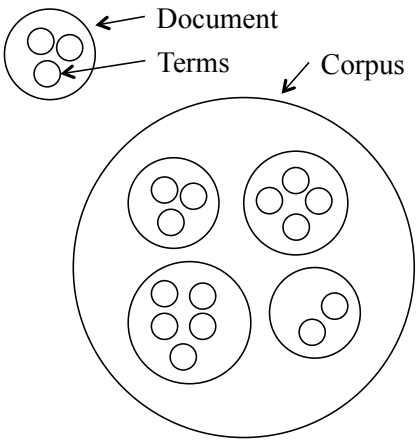

Figure 2.6: Vector Space Model Representation

The Vector Space Model will use the query from the users and convert into the vector. Then all terms in the corpus will also convert into the vector. The conversions are shown in the equations below.

$$\overrightarrow{d_i} = (W_{i,1} + W_{i,2} + \ldots + W_{i,t})$$

$$\vec{q} = (W_{q,1} + W_{q,2} + \ldots + W_{q,n})$$

where d is the document vector, and q is the query vector. The $w_{i,j}$ is the weight for term j in document i, and $w_{q,n}$ is the weight for term n in query q. Figure 2.7 depicts the document-vector conversion.

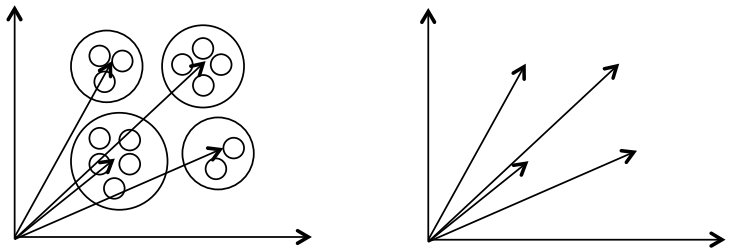

Figure 2.7: The Conversion of Documents and Words into Vector

The next step is finding the similarity value between the query vector and the terms' vector. The nearest vector, the most similar is the most relevant vector. Suppose the document and query vectors are very close that means extremely similar the vector should appear truly near or having the same direction. That means if they got a very small degree the value of similarity is close to 1 but if not it close to 0.

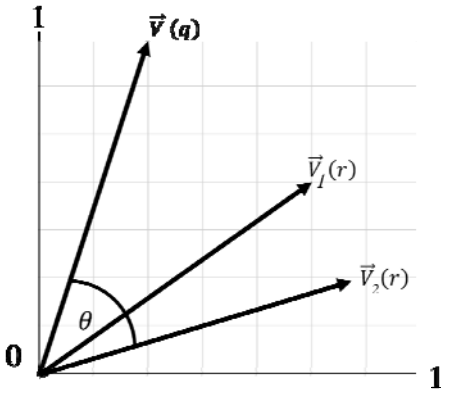

Figure 2.8: Cosine Similarity

The similarity value is represented by the cosine of angle between query and term-document. For example, the angle between them is the small degree. It means query vector and term-document vector are very similar. However, suppose the angle is the larger degree. It means these two vectors are not quite similar, and the similarity value can compute from the equation below.

$$sim(q,r) = \frac{\vec{V}(q) \cdot \vec{V}(r)}{|\vec{V}(q)||\vec{V}(r)|}$$

2.5 JavaScript Object Notation

The JavaScript Object Notation or JSON is the open standard format for storing and exchanging the data. [1] It stores the values as the object and each object represent the attribute-value pairs. The main advantages of this standard are human-readable and lightweight. Sometimes, the JSON may be used as the alternative to XML for transmission the data on the website.

The main structure of the JSON may build on two structures. The first structure is Collection of Name/Values pair's structure (Object). This structure represents the attributes of values on the left hand side and the value represented on the right hand side of the pair. Next, the Ordered List of Values structure (Array), it represents the list, sequence, vector, or array of the object. This data structure is represented the as square brackets at the beginning and ending of the list separated by the Comma. The example of these two structures are describes in Table 2.1.

Table 2.1 The Example Data of Two Structures of JSON

Structure	Example Data
Collection of Name/Values pair (Object)	“name” : “bob”, “age” : number(45)

Ordered List of Values
(Array)
	"pets": [
	"Dog",z
	"Cat"
	],
	"phoneNo": [
	{
	"type": "mobile",
	"no": "+66-8-9993-9293"
	},
	{
	"type": "home",
	"no": "+66-2-3544333"
	}
	],

The basic data types of JSON are four data types, the first type is String type. This type keeps the data as the sequence of Unicode characters. Next the Number type, it represents the signed number value. It supports the decimal value and exponential value. Next, the Boolean data type, it is represented the logical value either true or false. Last data type is Null data type. It represents the empty value. The examples of all four data types are shown in Table 2.2.

Table 2.2 The Example Data Type of JSON

Data Type	Example Data
String	"firstName": "Alice", "face": "\uD83D\nDE02"
Number	"weight": 73.4, "gpa": 3.75
Boolean	“student”: true, "active": false
Null	"children" : null, "school": null

The use of JSON is varying from application to application. For example, JSON use in Database Server such as MongoDB server and Web Service, and etc. The MongoDB use the concept of JSON format but it added some more specific data types in the object for instance Binary Value, Long Integer. The other application is the web service such as Twitter Web Service. The twitter website provides the web service for third-party application in order to retrieve the information of the twitter’s users. The service is very simple. The application needs to call the GET or POST method to the Resource URL and the information we return back as the JSON format.

CHAPTER III
LITERATURE REVIEWS

This chapter describes the related works of Threat identification in other techniques.

3.1 Correlation Technique in SIEM

The main function of the Correlation Technique is to reduce the false positive and prevent the false negative. Some of SIEM tools select one or more correlation techniques for their product in order to make the result more accurate. Normally, SIEM tools use a rule-based technique in correlation function, for example AlientVault OSSIM, HP ArcSight, Q1 Labs QRadar, Solar Winds Log & Event Manager, etc. [2, 3]. Table 3.1 shows the correlation technique in each SIEM tool. The first SIEM tool is AlientVault OSSIM, which is free software. It uses rule-based technique for correlation functions [4]. The results will be based-on vulnerability, risk, and priority which administrator defined. However, the Boolean operation does not support directly. The software supports a nested rule. For HP ArcSight, Q1 Labs Qradar, and Solar Winds Log & Event Manager, they use the rule-based technique and are able to support the Boolean operations. Figure 3.1 shows partial sample rules in XML format used by some software tools.

Table 3.1 The Correlation Technique in each SIEM Tools

SIEM Tools	Correlation Technique
AlientVault OSSIM	Rule-Based (Generic)
HP ArcSight Q1 Labs QRadar Solar Winds Log & Event Manager	Rule-Based (Support Boolean operation)

```
1  <directive id="4788276" name="My example directive" priority="5">
2       <rule   type="detector"   name="..."   reliability="..."   occurrence="1"
   from="..."
3       to="..." port_from="..." port_to="ANY" plugin_id="..." plugin_sid="...">
4            <rules>
                 <rule      type="detector"      name="..."      reliability="..."
   occurrence="20"
5                from="..." to="..." port_from="..." time_out="600" port_to="..."
6                plugin_id="..." plugin_sid="...">
7             </rules>
       </rule>
   <directive>
```

Figure 3.1: The Example Rules in XML Format from AlientVault OSSIM

The alternative methods and approaches of correlation techniques can be found in many techniques in difference areas. The most technique is using the Data Mining concept. Apriori algorithm in SIEM was introduced by Garbriel et al. [5]. The result generated consists of the association rules for identifying the relationship along with confidence and support values to a user for more accurate filtering. The examples of results are shown in Table 3.2.

Table 3.2 The Example of Association Rules that Applied in SIEM

High malware affection, if	Support %	Confidence %
user age category = IV and user gender = female	10.5	88.7
user age category = V and user gender = male and user is admin	5.5	70.5
…	…	…

Next technique is a scenario-based technique that was introduced by Rahayu [6]. They separate the pattern in their technique into two categories, victims and attackers. The pattern will create as a hierarchical graph and use tracing technique for identifying. This technique traces the patterns of the threat until the right victims and right attackers are found. The Figure 3.2 shows the tracing example of their model.

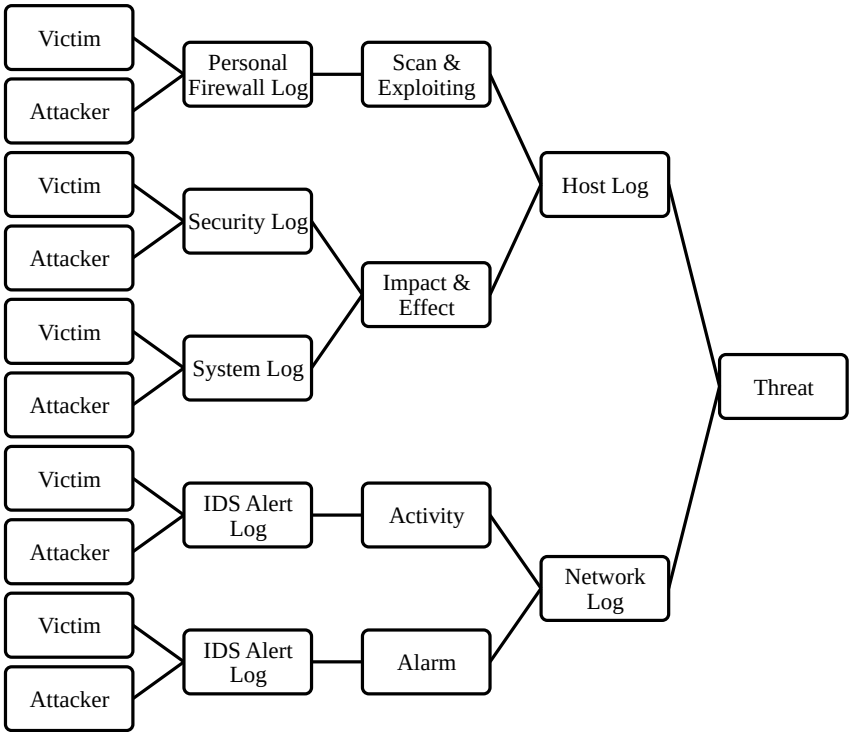

Figure 3.2: Multi-step (Victim/Attacker) Trace Pattern

The last technique is Streaming Mining with Supervised Learning. It was introduced by Pallabi Parveen, et al. [7]. This technique creates a cluster with dividers or Hyperplane. The Hyperplane will find the margin for dividing the node. The more margins create the longer distance between nodes. This technique needs to find the optimal Hyperplan by considering the nearest nodes. The figure 3.3 shows the example of nodes and Hyperplane.

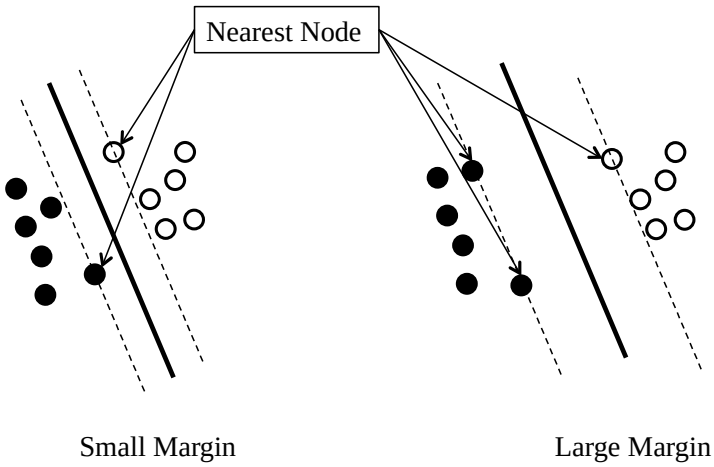

Figure 3.3: Hyperplane with Small and Large Margin

3.2 Comparison of Correlation Techniques

The survey of correlation techniques was research by Sergio Zamarripa López, et al. [8]. Table 3.3 presents the comparison among existing techniques.

Table 3.3 Comparison of Existing Correlation Technique

Technique	Pro	Contra
Finite State Machine	- Simple and easy to understand - Good as basic model	- Too easy and not suitable for practical application - No tolerance to noise
Rule-based	- Like a natural language - Modularity	- Time-consuming maintenance - Difficult to learn from experience

Table 3.3 Comparison of Existing Correlation Technique (cont.)

Technique	Pro	Contra
Finite State Machine	- Simple and easy to understand - Good as basic model	- Too easy and not suitable for practical application - No tolerance to noise
Rule-based	- Like a natural language - Modularity	- Time-consuming maintenance - Difficult to learn from experience
Case-based	- Learning Automatically - Compare the past experience is natural	- Automatic solutions adaption and reuse is difficult
Model-based	- Relies on a deep knowledge	- Difficult in practical to describe the description of behavior and structure
Codebook	- Fast and Robust - Change Adaptive	- Description of behavior manually is tedious - No notion of time
Bayesian networks	- Good theoretical foundation	- Probabilistic inference is NP-hard
Latent Semantic Analysis	- Like a natural language - Find the similarity patterns	- Complex Computation - Consume a lot of computing resources,

The rule-based is the natural way to let human understanding and easy to create. However, there are several drawbacks when use it in a complex network for threat identification [9]. The administrator needs to clarify the rules to handle all possible problems and symptoms. This may result in slowing down of overall system. Furthermore, the rule-based also gives a certain results. However, while solving the problem, administrator is always uncertain in reasoning. In conclusion, the uncertainty cannot be clarified by a rule-based system.

There is a solution for extending the rule-based system more uncertainly. A codebook system was introduced in 1995 by S. Kliger, et al. [10, 11]. In this technique, the administrator defines only the unique signature or symptom of the problem into the rule. Then, the system will find the matches or closet signature. The advantage of this technique is the administrator be able to identify the problem even the information is not completed. The efficiency of codebook was evaluated by Michael Tiffany. [12] He found that the codebook is more efficiency than the rule-based technique. The main reason is the less comparison for each event.

P. Dairinram, et al, introduced the Latent Semantic Analysis technique for threat identification [13]. This technique implements the LSA technique with SIEM for identification the threat. The man idea of this technique is to reduce the noise or unnecessary data from a huge data generated from network devices. Comparing to the non-reduced data, this technique also gave the same accuracy of answer. The pros of this technique are the users do not need to handle the rules for detecting every case. The administrator only defined the simple patterns of threat after that the technique will find the similarity value and ranks the result. The first rank will be the most relevant threat. However, the cons of this technique are complex computation and computing resource consuming such as CPU processing time and memory space. This technique performs the matrix calculation and computes the Single Value Decomposition (SVD).

3.3 Evaluation of Latent Semantic Analysis

The main advantage of LSA is reducing the size of input data. However, the data must have the same information. LSA performs a better efficiency than non-reduce data. There are many researches that measure the performance and accuracy of the LSA. Ch. Aswani Kumar and S. Srinivas compared the performance between the traditional Vector Space Model and Vector Space Model with LSA techniques. [14] They tested with four data sets. The number of terms is about 5000, and the number of documents is about 1000 – 3000.The query testingis about 30 – 200 queries.The result is the Vector Space Model with LSA technique performed more effective than the traditional approach about 30 percentages.

Another evaluation was researched by E.R. Jessup and J.H. Martin. [15] They measured the accuracy rate after compress the data by LSA. They compare the Term-Document Matrix (Full Rank Matrix) with Reduced Matrix (*k* Rank SVD) with Vector Space Model technique. The size of rank before compression is around 400 – 1200. The experimental results show that the rank that is acceptable should be in the range of 100 – 300.

CHAPTER IV
METHODOLOGY

This chapter presents the proposed threat identification method. The main component is the correlation function inside the SIEM. The first section describes the system overview of our proposed work. Next, the modules inside the SIEM are explained. Finally, the proposed threat identification system will be described in details.

4.1 System Overview

Our proposed system consists of four functions. The first one is Log consolidation function. It connects to the SIM and SEM. The proposed system obtains the data that is collected by SIM and SEM. Furthermore; it will perform data preprocessing and normalization. It is needed to normalize all alerts and event/logs into the same format, because, devices may generate a various formats of data. Next, it is Threat Correlation function. This function will correlate the current data and the existing threat patterns in the repository. Third function is Incident Management function. This function will collect the results after identified from previous function and create an action for solving the problem after an incident occurred. The last function is a Reporting function. It will generate the report to an administrator. All four functions can be figured out in Figure 4.1.

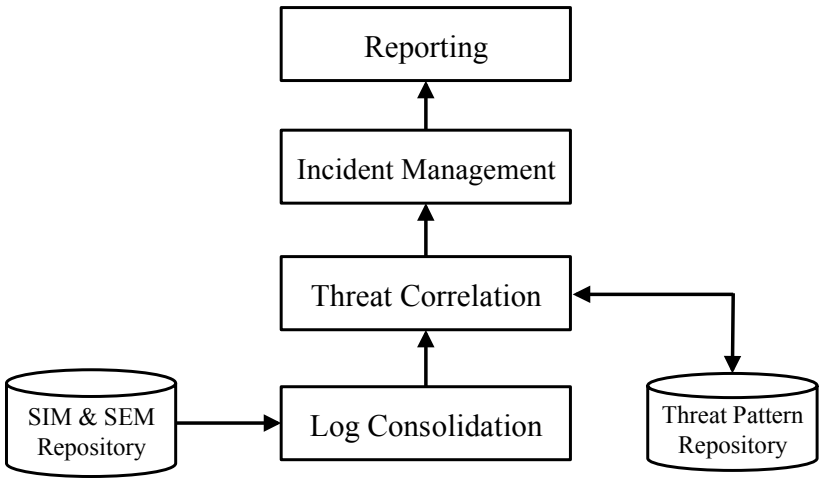

Figure 4.1: The Proposed System Overview

4.2 JSON Format

The proposed method introduces JSON format for keeping the data and log consolidation. The benefit of JSON format is simplicity, lightweight and human readable. This proposed method clarifies into two categories of data, which is kept in JSON format. There are the data of threat pattern and logs. The attribute name of both data should be defined as simple and same name in order to use for comparing. For example, the characteristic of threat "A" needs to consider the interval of time. The attribute name for threat defined as "intTime". The attribute name from network devices is defined as "intTime". These are the common attribute list that administrator will keep them in the SIEM repository in order to define the attribute name. The example of common attribute list is described in the Table 4.1.

Table 4.1 Common Attribute List

Data	Common Attribute Name	Value
Time Interval	intTime	Integer Number
Source IP Address and Port	srcIPPort	IP-Address:port Net mask:port

4.3 Analysis and Defining the Threat Pattern

The pattern of the threat or characteristics of the threat is the behaviors of the specific threat that behaving while attacking the victims. For example, the W32/Blaster worm, this worm was impacted to the PC that installed the Microsoft Windows 2000 and XP. The main characteristic of this threat is attacking the Remote Procedure Call (RPC). It will create the executable file, “MSBlast.exe” in the infected machine. The infected machine will spread out the worm and attack the victim by sending the packet that attacks the RPC service. It being attacking until the machine was crash and then automatically reboot in 60 seconds.

After study the threat patterns, next define all characteristics as JSON format and also defined the source that needs to collect by SIM and SEM. The example of JSON format of W32/Blaster worm shows in Figure 4.2.

```
{
  "Threat-ID": "4788276",
  "Threat-Name": "M.Blaster",
  "behaviors": [{
      "Source-ID": "IDS-SNORT",
      "info": [
        {"message": "TFTP Get"},
        {"message": "(portscan) TCP Portsweep"}]},{
      "Source-ID": "PC-WinXP-Windows-Security-Event-Logs",
      "info": [
        {"event-id": "592","img-file": "tftp.exe"},
        {"event-id": "592","img-file": "msblast.exe"}]},{
      "Source-ID": "workstation-personal-firewall",
      "info": [{
          "connection-type": "tcp-inbound",
          "srcIP-port": "*:135"},{
          "connection-type": "tcp-inbound",
          "srcIP-port": "*:444"},{
          "connection-type": "open udp",
          "srcIP-port": "*:69"
   }]}]
}
```

Figure 4.2: Example Characteristics of W32/Blaster for the System

Suppose there is a worm attacking the machines inside the network system. However, the administrator needs to consider other threats that being attack the specific assets or systems inside the network system. For example, the Enterprise Resource Planning (ERP) Server may be attacked from the staff inside the company. The administrator should clarify the possible characteristics that may effect to the assets of the organization.

Assume that the organization has the ERP system. The normal activity is the staffs in Sales and Accounting Department should use the system from 6am until 6pm. The others department will not be able to use the ERP system. The last threat, the system will not allow to sign in the system when users enter the wrong username and password more than 10 times in 5 minutes (Brute-force attack). The example of these threats are shown in Figures 4.3 to 4.5.

```
{
  "Threat-ID": "5237675",
  "Threat-Name": "ERP-Inappropriate Sign In",
  "behaviors": [
    {
      "Source-ID": "ERP-Authentication Log",
      "info":[ {
           "message": "Authentication Successful",
           "initialtime": "6pm",
           "interval": "43200" },]
    },
  ]}
```

Figure 4.3: Example of Inappropriate Sign In

```
{
  "Threat-ID": "5237676",
  "Threat-Name": "ERP Non-Permission Threat",
  "behaviors": [
    {
      "Source-ID": "Firewall ",
      "info":[ {
           “srcIPSubNet” : “10.34.15.0/24”,
           "srcIPSubNetName": "Miscellaneous Department",
    },
  ]
}
```

Figure 4.4: Example of Non-Permission Users Threat

```
{
  "Threat-ID": "5237677",
  "Threat-Name": "ERP Bruce-force attack",
  "behaviors": [
    {
      "Source-ID": "ERP-Authentication Log",
      "info":[ {
           "message": "Wrong Username and Password!",
           "number": "10",
           "interval": "300" },]
    },
  ]
}
```

Figure 4.5: Example of Brute-force Attack Threat

The administrator should clarify all-possible threats or impropriate actions. These may be declared as someone attacked the system. The best practice is the administrator should conform to the security standard in order to clarify the threat. The standard will defined which criteria that can prevent and protect the system. The examples of best practice for ISO/IEC 27000 Series are described in Table 4.2 [16].

Table 4.2 Best Practice for ISO/IEC 27000 Series

Standard	Outline	Best Practice
ISO/IEC 27002: 2013	9. Access Control 9.2 User Access Management	9.2.5 Review access rights at regular intervals
	13. Communication Security 13.2 Protect information transfers	13.2.3 Protect information sent using electronic messaging

For the best practice in Table 4.2, the administrator needs to create two threats pattern, the first threat was describes in Figure 4.3. The second threat is that the administrator needs to create the pattern as shown in Figure 4.6.

```
{
  "Threat-ID": "5237679",
  "Threat-Name": "Transferring File via LINE.",
  "behaviors": [
    {
      "Source-ID": "",
      "info":[ {
           "message": "Wrong Username and Password!",
           "number": "10",
           "interval": "300" },]
    },
  ]
}
```

Figure 4.6: Example of Transferring the Secure Information Threat

4.4 Log Consolidation Function

The SIM and SEM have responsibility only collecting the data from network devices. The collected data has different format. Some devices provide a SNORT format while some other devices provide SNMP format. In the worst case, they provides in the proprietary format. This is the issue mentioned in previous section. This issue needs to solve by normalizing the data before correlating.

Our proposed system was designed to keep all data in a JSON format. It will collect all attributes of all alert and events/logs. It also does some normalizing by removing the duplicated or similar attributes and keep only the same attributes that appear from different sources. Table 4.3 shows the example of logs from Personal Firewall that installed on the workstation, and the security logs that generate from Microsoft Windows. Table 4.4 describes attributes, which will be used for normalizing the data. After normalizing, all normalized attributed will check in the attribute common list for finding the existing clarified attribute name. Suppose the normalized attribute does not exist in list, administrator needs to clarify the attribute and add into the common attribute list.

Table 4.3 Examples of Attribute in Personal Firewall, Security Log in Microsoft Windows, and FTP Server

Logs	Sample Data	Attribute
Personal Firewall	… 2014-10-14 15:34:42 OPEN-INBOUND TCP 10.34.15.1 10.34.15.229 3993 135 … 2014-10-14 15:34:52 OPEN-INBOUND TCP 10.34.15.1 10.34.15.229 4002 4444 ...	- Date and Time - Connection Type - Source IP and Port - Destination IP and Port
Security Log	... 2014-10-14 16.01.02 Security Audit Detailed Tracking 592 NT AUTHORITY\SYSTEM “A new process has been created:”... Image File Name C:\WINDOWS\system32\hello.exe	- Date and Time - Event ID - Event Description - Image File Name

After the data was collected, the system analyzes the attributes that represent in the all data. Table 4.4 shows the normalized attributed which collected from workstation. The common attribute is “Date and time”, the Personal Firewall and Security Log represent the date and time data. This attribute should eliminate and normalize into a common attribute. And other attributes are unique and no need to normalize. In this example, the attribute was normalized from eight attributes to seven attributes.

Table 4.4 Normalized Attributes

Logs	Sample Data	Normalized Attribute
Personal Firewall	- Date and Time - Connection Type - Source IP and Port - Destination IP and Port	- Date and Time - Connection Type - Source IP and Port - Destination IP and Port
Security Log	- Date and Time - Event ID - Event Description - Image File Name	- Event ID - Event Description - Image File Name

Next, the system converts the normalized attributed into the attribute of JSON. Then, it finds the value that matched the defined normalized attribute and assigns the value. The final results are obtained as shown in Figure 4.7 and Figure 4.8.

```
{
   "_id": "53993c91ce0ca6e908d63af1"
   "workstation-personal-firewall" : [
         {
            "date-time": "20141014-153442",
            "connection-type" : "tcp-inbound",
            "srcIP-port" : "10.34.15.1:3993",
            "dstIP-port" : "10.34.15.229:135"
         },
    ]
};
```

```
{
   "_id": "53993c91ce0ca6e908d63af2"
   "workstation-personal-firewall" : [
         {
            "date-time": "20141014-153452",
            "connection-type" : "tcp-inbound",
            "srcIP-port" : "10.34.15.1:4002",
            "dstIP-port" : "10.34.15.229:4444"
         },
    ],
};
```

Figure 4.7: Message Sequence from Personal Firewall in Workstation

```
{
   “_id”: “53993c91ce0ca6e908d63af1”
   “PC-WinXP-Windows-Security-Event-Logs” : [
         {
            “date-time”: “20141014-153442”,
            “event-id” : “592”,
            “event-dsc”: “NT AUTHORITY\SYSTEM A new process has been created:”,
            “img-file” : “C:\WINDOWS\system32\hello.exe”
         },
    ]
};
```

Figure 4.8: Message from Security Log in Workstation

4.5 Correlation Function

The correlation function is one of the most important functions among the four functions of SIEM tools. This function performs four steps as commonly found in LSA technique.

4.5.1 Creating of Event-Threat Matrix

This step collects the threat information from Threat repository that was kept in JSON format and creates Event-Threat Matrix. For the correlation matrix, rows present the event and columns present the threats patterns. One column represents for one threat and one row represents one behavior or one event. The number of events must be the same number of events in the current interval. For instance, the current interval, there are 100 events represented in current information. Thus the number of rows in Event-Threat must be 100. Figure 4.9 shows sample information from SIM and SEM

Threat / Event	SIM	SEM
E_1	X	
E_2	X	X
E_3	X	
E_4		X
…		
E_N		X

Threat / Event	T1	T2	T3	T4
E_1	X		X	
E_2	X	X	X	X
E_3	X		X	
E_4		X		X
…				
E_N		X		X

Figure 4.9: Number of Events from SIM and SEM and Event-Threat Matrix

4.5.2 Weighting the Value

After creating the Event-Threat Matrix, the matrix needs to represent the weights. Weighting may be Binary levels scheme, which are represented by '1' and '0' for matched and not matched respectively. Figure 4.10 shows the example of four threats patterns. The threat T1 represents the behavior events E1, E2, E3, and E4. However, it does not represent events E5, E6, and E7. The system gives the value of weights for all threats.

Event / Threat	T1	T2	T3	T4
E_1	1	0	1	1
E_2	1	0	0	0
E_3	1	1	0	1
E_4	1	0	0	1
E_5	0	1	0	0
E_6	0	0	1	1
E_7	0	1	0	0

Figure 4.10: Example of Event-Threat Matrix

4.5.3 Applying SVD

The Event-Threat Normalized Matrix will be decomposed into three multiplication matrices by using Single Value Decomposition (SVD). The equation for SVD computation is given as follow.

$$A = U\Sigma V^T$$

where A is the event-threat matrix, U is the t by m orthonormal matrix, Σ is m by m diagonal matrix, and V is the m by d m orthogonal matrix. After applying the SVD, some rows and columns of U, Σ, and V are removed for eliminating the unnecessary noise.

After the equation generates the three matrices, the system uses the Σ matrix to identify the dimension of data determined by number of rows and columns of matrix. The first value needed to determine is k. It needs to pick up the value of the dimension used to query. It may use only one dimension or more. We will use k to determine number of columns (dimension) that will be selected for actual calculation. However, the next problem is which columns is good selection. The technique to get the suitable columns is to select from the value of a diagonal matrix Σ. It represents the values from the largest to smallest from top left to bottom right. Normally, we will keep the higher values and discard the smaller ones. The example of the three matrices are shown in Figure 4.11.

U x Σ x V^T

E1	0.21	0.08
E2	0.19	-0.23
E3	0.36	0.27
E4	0.68	-0.44
E5	0.13	0.53
E6	0.51	0.61

x

2.47	0
0	2.35

x

T1	T2	T3	T4	T5	T6
0.71	0.34	0.25	0.41	-0.47	0.28
0.62	-0.23	-0.35	0.29	-0.55	0.32

Figure 4.11: Example of Three Matrices with k=2

4.5.4 Querying and Ranking

The last process after use SVD function is querying and ranking the results. By using the information observed some interval of time, the search for threat's patterns can be started. The threat's patterns need to transform into a vector format by using the following equation.

$$\hat{q} = q^T U_k \Sigma_k^{-1}$$

where k is the number of dimension after reduction, U is the t by m orthogonal matrix, and Σ is m by m diagonal matrix. The example of queries and current information vector values are described in Table 4.5. The value of query vector and Current Information vector values presented in two dimensional tables.

Table 4.5 Vector Values for Threats and Current Information from SIM and SEM

Threat	**Query Vector**		**Current Information from SIM and SEM Vector**	
T1	-0.4509	-0.3873	-0.4904	0.3012
T2	-0.0288	0.1291	-0.4904	0.3012
T3	-0.0608	0.5164	-0.4904	0.3012

Next, applying all into the Vector Space Model for scoring, we compute the Cosine similarity. These values represent the cosine of angle between query vector and threats patterns. For example, the angle between them is close to 0 degree. It means query vector and threat patterns vectors are very similar. It may imply that threat is being attack to the network. However, the angle is close to 90-degree. It means these two vectors are not quite similar, and another meaning is a threat probably impossible being attack in a network. The examples of similarity value are shown in Table. 4.6.

Table 4.6 Similarity Value for each Threat

Threat	Query Vector		Current Information from SIM and SEM Vector		Similarity Value
T1	-0.4509	-0.3873	-0.4904	0.3012	0.3053
T2	-0.0288	0.1291	-0.4904	0.3012	0.6963
T3	-0.0608	0.5164	-0.4904	0.3012	0.6194

The last step, ranking the similarity value for each threat, we use the similarity value for each threat and rank the highest value to the lowest. The maximum value is 1, it means this threat may be possible being attack the network. However, the other ranks may probably too. The results can suggest us to eliminate some threats by making a threshold value. The administrator will use this ranking to decide, which threats are being attack in the network.

However, it is not possible that only single threat is being attack. Our proposed system supports the “AND” and “OR” for Boolean operation. The column may represent as more than one threat. It represent “AND” Boolean operation. Then “OR” operation represent in the final result. For example, the result shows the first rank as “T1 AND T2” is being attack and the second rank shows, as “T3” are being attack. It describes “T1 and T2” is being attack “OR” just “T3” is being attack. The example of Boolean operation implementation represents in table 4.7.

Table 4.7 Example of Event-Threat Matrix with Boolean Operation

Event / Threat	T1	T2	T3	T1 and T2	T1 and T3	T2 and T3	T1 and T2 and T3
E_1	1	0	1	1	1	1	1
E_2	1	0	0	1	1	0	1
E_3	1	1	0	1	1	1	1
E_4	1	0	0	1	1	0	1
E_5	0	1	0	1	1	1	1
E_6	0	0	1	0	1	1	1
E_7	0	1	0	1	0	1	1

Then compute the SVD and create the vector value again. The final result is the similarity value for all possible cases. The results show in Table 4.8. And the rank shows in Table 4.9.

Table 4.8 Similarity Value for each Threat

Threat	Query Vector		Current Information from SIM and SEM		Similarity Value
T1	-0.4509	-0.3873	-0.4904	0.3012	0.3053
T2	-0.0288	0.1291	-0.4904	0.3012	0.6963
T3	-0.0608	0.5164	-0.4904	0.3012	0.6194
T1 and T2	-0.4797	-0.2582	-0.4904	0.3012	0.5022
T1 and T3	-0.5117	0.1291	-0.4904	0.3012	0.9542
T2 and T3	-0.0895	0.6455	-0.4904	0.3012	0.6354
T1 and T2 and T3	-0.5405	0.2582	-0.4904	0.3012	0.9944

Table 4.9 Threat Ranking

Rank	Similarity Value	Threat
1st	0.9944	T1 and T2 and T3
2nd	0.9542	T1 and T3
3rd	0.6963	T2
4th	0.6354	T2 and T3
5th	0.6194	T3
6th	0.5022	T1 and T2
7th	0.3053	T1

4.6 Ranking Threshold

From the previous example, the ranking result will rank from the most relevant to the less. And the first rank is the most possible that threat is being attack the network. However, the second and the third are also possible to being attack the network. The SIEM may keep the threat patterns more than the example. And the rank may show both relevant and not relevant threats. The final result needs to adjust the rank for the administrator easy to make the decision. The threshold number is the one common technique for eliminates the non-relevant result. Our proposed method presents the average value approaches for eliminate the non-relevant result.

The common approach uses the average value. This approach assumes the all most relevant threat is rare rank. And the non-relevant is the common rank. It means the number non-relevant threat should more than the number of relevant rank. From this assumption, the average value represents the trends of these final rank results. The average calculates by this equation below.

$$\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

where the $\overline{X}$ is the average value, *n is* the number of all ranks, and X_i is the rank of each threat. From the Table 4.9, the average result equal to 0.6724. And the adjusting results with average threshold value presents in Table 4.10. The final results show only the first three ranks from seven ranks. And the correct answer is the T1 and T2 and T3 that remained in the final result. The second and third ranks may imply as the correct answer in the first rank.

Table 4.10 Threat Ranking with Average Threshold Adjustment

Rank	Similarity Value	Threat
1st	0.9944	T1 and T2 and T3
2nd	0.9542	T1 and T3
3rd	0.6963	T2

CHAPTER V
EXPERIMENTAL RESULTS

In this chapter, we implement the proposed system. The experiments will be separated into two phases with two types of data sets. The first experiment will be conducted with the real data from closed network. The second experiment compares all three techniques by using the data set from the CERT Division Carnegie Mellon University [17]. The data is generated as a set of synthetic insider threat test data sets. These data sets provide both simulated background data and data from malicious code. This chapter presents the results of the implementation showing efficiency, calculation time, and accuracy of rule-based techniques comparing with our proposed technique.

5.1 Experiments in the Closed Network Environment

The proposed method was performed in a closed network with nine machines that installed the Microsoft Windows XP and Microsoft Windows 7 and separating them into three zones. First, it is the attacker's zone; the three machines are infected by two malicious codes (e.g. W32.Blaster.Worm and W32.Mydoom.A@mm) and installed the Brute-force Attack tools. Next, it is called network devices' zone; three machines were installed the SNORT and Firewall and also install the SIEM tool. The last zones called the victim's zone; they were installed with the FTP Server, which is used by Brute-force Attacker Tool. The network diagram for this experiment is shown in Figure 5.1.

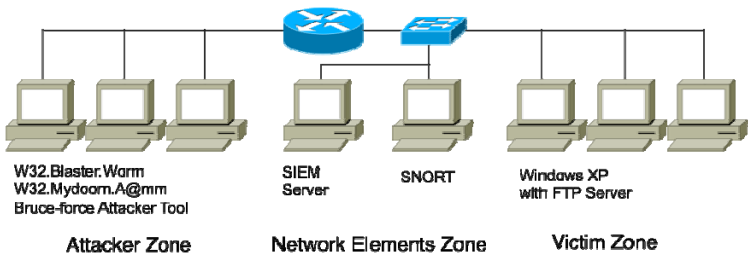

Figure 5.1: Diagram of Closed Network in the Experimental

5.1.1 Preprocessing

The threats need to clarify their characteristics and create the threat pattern in the JSON format. There are three threats used for these two experiments. The first threat is W32.Blaster Worm. The behavior of this threat is described below.[18]

1) The protocols it used are TCP and UDP. There are three ports used for attacking, which are TCP Port 135 (DCOM RPC), UDP Port 69 (TFTP), and TCP Port 4444 (Remote Shell).

2) The infected machine will create the two execution files and run the background process in order to spread out the worm. There are the tftp.exe file in C:\Windows\System32\tftp.exe directory and the msblast.exe file in C:\Windows\System32\mablast.exe directory.

3) The worm will call the RPC service and then the machine will restart in 60 seconds.

After analyze the pattern of threats, it needs to clarify which information and source that can be used for threat identification. The first pattern is the protocol that this worm uses. It uses both TCP and UDP protocols. The information that is needed to consider is the network traffic logs. The device that serves this information is the firewall. The firewall keeps the network traffic of both be source and destination machines. Any machines using the worm's ports need to be considered and kept in the repository. Next, the worm creates the execution file for spreading out itself to other computers. The information that needs to be considered is the Windows Security Logs. It keeps the Microsoft Windows activities and it also provides the ID for referencing the activities. The ID of create the execution file activity is 592. The message of ID 592 has the word kept in Security Log Message as "A new process has been created:". The Security Log also provides the execution file name when it is created. This file will be used for checking the worm's execution file. The last pattern needed to consider is the RPC Service. The source providing it is Windows System Logs. It also provides the ID's for referencing as 7031 and 1074. The ID 7031 provides the information about the RPC that was terminated unexpectedly. The message for this ID is "The Remote Procedure Call (RPC) service terminated unexpectedly". Another ID provides the message like "Windows must now restart because the Remote Procedure Call (RPC) service terminated unexpectedly". Another information that is added for

this experiment is the source from SNORT-IDS. The execution of this worm is scanning the machines inside the network and spread out itself. The source that can collect this information is SNORT-IDS. The message from SNORT, while this worm is attacking, is “NETBIOS DCERPC NCACN-IP-TCP ISystemActivator Remote CreateInstance attempt”. The summary of all patterns is described in Table 5.1.

Table 5.1 Patterns of W32.Blaster Worm

Threat	Source	Information
32.Blaster.Worm	Security Event Log	EventId = 592 & Image Files =msblast.exe
		EventId = 592 & Image Files =tftp.exe
		EventID = 1074 & Msg = "Windows must restart.."
		EventID = 7031 & Msg = "The Remote Procedure Call (RPC).."
	Firewall	TCP : Src.Port = ANY & Dst.Port = 135
		TCP : Src.Port = ANY & Dst.Port = 4444
		UDP: Src.Port = ANY & Dst.Port = 69
	SNORT-IDS	SHELLCODE x86 NOOP
		NETBIOS DCERPC NCACN-IP-TCP ISystemActivator Remote CreateInstance attempt

The second threat that is needed to clarify is W32.Netsky @mm worm. This worm attacks by sending an email and spread out itself via the network. After infected, this worm will change the Windows Registry and spread itself to other victims. It uses the SMTP server list for sending the message. This information was

collected by the Firewall. The example list of this worm that query the MX record for sending the email is shown in the Table 5.2 [19].

Table 5.2 IP Address of SMTP Server for the W32.Netsky @mm

IP Address		
212.185.252.73	193.193.144.12	193.141.40.42
212.185.253.70	212.7.128.162	145.253.2.171
212.185.252.136	212.7.128.165	193.189.244.205
194.25.2.129	193.193.158.10	213.191.74.19
194.25.2.130	194.25.2.131	151.189.13.35
195.20.224.234	194.25.2.132	195.185.185.195
217.5.97.137	194.25.2.133	212.44.160.8
194.25.2.129	194.25.2.134	

Then the summary of W32.Netsky @mm patterns is described in Table 5.3.

Table 5.3 Patterns of W32.Netsky @mm

Threat	Source	Information
W32.Netsky @mm	Firewall	Query DNS Type MX (one of the list)

The last threat for this experiment is Brute-force attacking. This threat is difference from the previous two threats. The human performs it. So in this experiment, the FTP Server is setup for attacking by Brute-force program. It performs the login process to the FTP server by guessing the username and password. The username and password using in guessing are collected from the username and password dictionary. The information that needed to analyze is the abnormal activities on the server. This information is obtained from FTP Server logs. Another information

that is added for this experiment is the source from Firewall. The summary of all patterns is described in Table 5.4.

Table 5.4 Patterns of Brute-force Attack

Threat	Source	Information
Brute-force Attack	FTP Server Log	User Authentication Failed
	Firewall	Src.Port = ANY & Dst.Port = 21

5.1.2 Proposed Method Experiments

Before experiments, the threat behavior will analyze for two techniques. All three threats will create the pattern as JSON format. The three patterns was created in JSON format and shown in Figure 5.2 to Figure 5.4.

```
{
  "Threat-ID": "4788276",
  "Threat-Name": "32.Blaster.Worm",
  "behaviors": [
    {
      "Source-ID": "IDS-SNORT",
      "info": [
        {"message": "TFTP Get"},
        {"message": "(portscan) TCP Portsweep"}
      ]},{
      "Source-ID": "PC-WinXP-Windows-Security-Event-Logs",
      "info": [
        {"event-id": "592","img-file": "tftp.exe"},
        {"event-id": "592","img-file": "msblast.exe"}
      ]},{
      "Source-ID": "workstation-personal-firewall",
      "info": [
        {"connection-type": "tcp-inbound",
          "srcIP-port": "*:135"},
        {"connection-type": "tcp-inbound",
          "srcIP-port": "*:444"},
        {"connection-type": "open udp",
          "srcIP-port": "*:69"
}]}]}
```

Figure 5.2: Threat Pattern of 32.Blaster.Worm in JSON Format

```
{
  "Threat-ID": "4788277",
  "Threat-Name": " W32.Netsky @mm",
  "behaviors": [
    {
      "Source-ID": "network-firewall",
      "info": [
        {
          "connection-type": "udp",
          "dstIP-Port": "*:53",
          "ipList": [
            "212.185.252.73",
            "212.185.253.70",
            "212.185.252.136",
            "194.25.2.129",
            "194.25.2.130",
            "195.20.224.234",
            "217.5.97.137",
            "194.25.2.129",
            "193.193.144.12",
            "212.7.128.162",
            "212.7.128.165",
            "193.193.158.10",
            "194.25.2.131",
            "194.25.2.132",
            "194.25.2.133",
            "194.25.2.134",
            "193.141.40.42",
            "145.253.2.171",
            "193.189.244.205",
            "213.191.74.19",
            "151.189.13.35",
            "195.185.185.195",
            "212.44.160.8"
          ]
        }
      ]
    }
  ]
}
```


Figure 5.3: Threat Pattern of W32.Netsky @mm in JSON Format

```
{
  "Threat-ID": "4788277",
  "Threat-Name": "FTP Brute-force",
  "behaviors": [{
      "Source-ID": "FTP Server",
      "info": [{
          "message": "530 Login or password incorrect!",
          "number": "10",
          "interval": "300"
        }]
    }]
}
```

Figure 5.4: Threat Pattern of Brute-force in JSON Format

After administrator creates the threat pattern in JSON format, it will be assigned the Event number for representing itself. Table 5.5 shows the assigned event number for three threats.

Table 5.5 Assigned Event Number for Patterns of Three Threats

Threat	Source	Information	Event
32.Blaster.Worm	Security Event Log	EventID = 592 & Image Files =msblast.exe	E1
		EventID = 592 & Image Files =tftp.exe	E2
		EventID = 1074 & Msg = "Windows must restart.."	E3
		EventID = 7031 & Msg = "The Remote Procedure Call (RPC).."	E4
	Firewall	TCP : Src.Port = ANY & Dst.Port = 135	E5
		TCP : Src.Port = ANY & Dst.Port = 4444	E6

Table 5.5 Assigned Event Number for Patterns of Three Threats (cont.)

Threat	Source	Information	Event
32.Blaster.Worm	Firewall	UDP: Src.Port = ANY & Dst.Port = 69	E7
	SNORT-IDS	SHELLCODE x86 NOOP	E8
		NETBIOS DCERPC NCACN-IP-TCP ISystemActivator Remote CreateInstance attempt	E9
W32.Netsky @mm	Firewall	Query DNS Type MX (one of the list)	E10
Brute-force Attack	FTP Server Log	User Authentication Failed	E11
	Firewall	Src.Port = ANY & Dst.Port = 21	E12

Next, we create a threat pattern from the threat's characteristic. Table 5.6 shows the results from three threats and 12 events. Given T1 is the W32.Blaster.Worm, T2 is the W32.Mydoom.A@mm, and T3 is the FTP attack from Brute-forced tool.

Table 5.6 Event-Threat Matrix

Event / Threat	T1	T2	T3	T1 AND T2	T1 AND T3	T2 AND T3	T1 AND T2 AND T3
E_1	1	0	0	1	1	0	1
E_2	1	0	0	1	1	0	1
E_3	1	0	0	1	1	0	1

Table 5.6 Event-Threat Matrix (cont.)

Event / Threat	T1	T2	T3	T1 AND T2	T1 AND T3	T2 AND T3	T1 AND T2 AND T3
E_4	1	0	0	1	1	0	1
E_5	1	0	0	1	1	0	1
E_6	1	0	0	1	1	0	1
E_7	1	0	0	1	1	0	1
E_8	1	0	0	1	1	0	1
E_9	1	0	0	1	1	0	1
E_{10}	0	1	0	1	0	1	1
E_{11}	0	0	1	0	1	1	1
E_{12}	0	0	1	0	1	1	1

Then, we convert the threat pattern to vector format by using the LSA technique. The vector value of each threat is shown in Table 5.7.

Table 5.7 Query Vector for Experiments in Closed Network Environment

Threat	Vector Value	
T1	-0.4509	-0.3873
T2	-0.0288	0.1291
T3	-0.0608	0.5164
T1 and T2	-0.4797	-0.2582
T1 and T3	-0.5117	0.1291
T2 and T3	-0.0895	0.6455
T1 and T2 and T3	-0.5405	0.2582

We conduct the experimental by collecting the data from device in 15 minutes with three times. Figure 5.5 to Figure 5.7 show the example of information from SIM and SEM that is converted into JSON format by the log consolidation process.

```
{
   “_id”: “53993c91ce0ca6e908d63af1”
   "Source-ID": "workstation-personal-firewall”,
      "info" : [{
            “connection-type” : “tcp-inbound”,
            “srcIP-port” : “10.34.15.1:3993”,
            “dstIP-port” : “10.34.15.229:135”
         }]
};
```

```
{
   “_id”: “54653b52ce0ca6c7aad63af1”
   "Source-ID": "PC-WinXP-Windows-Security-Event-Logs",
      "info" : [{
            “event-id” : “592”,
            “event-dsc”: “NT AUTHORITY\SYSTEM A new process has been created:”,
            “img-file” : “C:\WINDOWS\system32\msblast.exe”
         }]
};
```

Figure 5.5: Message Sequence in SIM and SEM (W32.Blaster Worm)

```
{
   “_id”: “4fc3300ba149ce7ef86deb3d”
      "Source-ID": "network-firewall",
      "info": [{
           "connection-type": "udp",
           "dstIP-Port": "194.25.2.133:53",
         }]
};
```

```
{
   “_id”: “4fc3300ba149ce7ef86deb3c”
      "Source-ID": "network-firewall",
      "info": [{
           "connection-type": "udp",
           "dstIP-Port": "212.7.128.165:53",
         }]
};
```

Figure 5.6: Message Sequence in SIM and SEM (W32.Netsky Worm)

```
{
   “_id”: “4fc3300aa149ce7ef86dea04”
      "Source-ID": "FTP Server",
      "info": [
         {
            “message” : “530 Login or password incorrect!”,
         },
    ]
};
```

Figure 5.7: Message Sequence in SIM and SEM (FTP Brute-force Attack)

We assign the weight of “1” for matched threat patterns. Table 5.8 shows the result from each machine in attacker zone and victim zone. These data collected by SIM and SEM and also these results are used for generating the vector value.

Table 5.8 Matrix Keeps in SIM and SEM in the Closed Network Environment

Event / Hosts	Attack A	Attack B	Attack C	Victim A	Victim B	Victim C
E_1	0	0	0	0	1	1
E_2	0	0	0	0	0	0
E_3	0	0	0	0	1	1
E_4	0	0	0	0	1	1
E_5	1	0	1	0	1	0
E_6	0	1	0	0	1	1
E_7	0	1	0	0	1	1
E_8	0	1	0	0	1	1
E_9	0	1	0	0	1	1
E_{10}	0	1	1	0	0	0
E_{11}	0	0	0	1	0	0
E_{12}	0	0	1	0	0	0

After that the matrix from Table 5.8 is used to convert vector value. There are two experiments. Table 5.9 are shown the query vector with k = 2.

Table 5.9 Query Vector for Two Experiments in Closed Network Environment

Event	Current Information from SIM and SEM	
	Three Threats Attacking	**Two Threats Attacking**
E_1	1	1
E_2	0	0
E_3	1	1
E_4	1	1
E_5	1	1
E_6	1	1
E_7	1	1
E_8	1	1
E_9	1	1
E_{10}	1	1
E_{11}	1	0
E_{12}	1	0
Query Vector	-0.490383, 0.301230	-0.429609, -0.215164

Next, calculate the similarity by using the vector space model. The similarity shows the relevancy between the current information and threat pattern. Then, we rank the similarity values in descending order. The ranking from two experiments is described in Table 5.10. The first experiments attacked from three threats, the first rank is correct. The second experiment, the T3 did not being attacked a network, but it appears on the third rank with T1.

Table 5.10 Ranking of Two Experiments in Closed Network Environment

Ranks	Three Threats Attacking		Two Threats Attacking	
	Threats	Similarity Value	Threats	Similarity Value
1st	T1 & T2 & T3	0.994478	T1 & T2	0.999568
2nd	T1 & T3	0.954251	T1	0.970070
3rd	T2	0.696335	T1 & T3	0.757363
4th	T2 & T3	0.635429	T1 & T2 & T3	0.613709
5th	T3	0.619408	T2	-0.242462
6th	T1 & T2	0.502274	T2 & T3	-0.320842
7th	T1	0.305382	T3	-0.340260

Then adjust the final results by the average threshold value, and theses two values are shown in Table 5.11.

Table 5.11 Threshold Value

Three Threats Attacking	Two Threats Attacking
0.672508	0.348164

Ranks were adjusted by the thresholds value from Table 5.11. The ranks that have the similarity lower than the threshold will eliminate.

Table 5.12 Adjusting Ranking of Two Experiments

Ranks	Three Threats Attacking		Two Threats Attacking	
	Threats	Similarity Value	Threats	Similarity Value
1st	T1 & T2 & T3	0.994478	T1 & T2	0.999568
2nd	T1 & T3	0.954251	T1	0.970070
3rd	T2	0.696335	T1 & T3	0.757363
4th	-	-	T1 & T2 & T3	0.613709

After adjusting the results, the correct answer is found in the ranking. The correct answer appears at the first rank. For these results, it is accepted as the correct answer when the correct answer appears in any ranks. Comparing to the answer, any rank that appears as a part the correct answer's threat, it will be marked as the incorrect answer.

Table 5.13 The Result of Two Techniques

Techniques	Experiments	Answer
LSA Technique	3 Threats	(1st Rank) T1, T2, T3 (Correct)
	2 Threats	(1st Rank) T1, T2 (Correct)

5.1.3 Experiments from Rule-based techniques

Next, we compare the performance with another tradition method that is the rule-based technique. The rule based implemented based on the JAVA syntax and all three threats are needed to be clarified as shown in Figure 5.8 to Figure 5.10.

```
ThreatName = null;

if(Source-ID == “IDS-SNORT”) {
   if(message == ”TFTP Get” && message == “(portscan) TCP Portsweep”) {
      if(Source-ID == “PC-WinXP-Windows-Security-Event-Logs”) {
          if((event-id == "592" && img-file == "tftp.exe") &&
             (event-id == "592" && img-file == "msblast.exe")) {
             if(Source-ID == "workstation-personal-firewall") {
                if((connection-type == "tcp-inbound" && srcIP-port == "*:135") &&
                   (connection-type == "tcp-inbound" && srcIP-port == "*:444") &&
                   (connection-type == "open udp " && srcIP-port == "*:69")) {
                      ThreatName = “32.Blaster.Worm”
                 }
              }
          }
       }
   }
}
```

Figure 5.8: Threat Pattern of 32.Blaster.Worm in Rule-based

```
ThreatName = null;

if(Source-ID == “network-firewall”) {
   if(connection-type == ”udp” && dstIP-Port == "*:53" {
      if(ipList == "212.185.252.73"  || ipList == "212.185.253.70" ||
         ipList == "212.185.252.136" || ipList == "194.25.2.129" ||
         ipList == "194.25.2.130"    || ipList == "195.20.224.234" ||
         ipList == "217.5.97.137"    || ipList == "194.25.2.129" ||
         ipList == "193.193.144.12"  || ipList == "212.7.128.162" ||
         ipList == "212.7.128.165"   || ipList == "193.193.158.10" ||
         ipList == "194.25.2.131"    || ipList == "194.25.2.132" ||
         ipList == "194.25.2.133"    || ipList == "194.25.2.134" ||
         ipList == "193.141.40.42"   || ipList == "145.253.2.171" ||
         ipList == "193.189.244.205" || ipList == "213.191.74.19" ||
         ipList == "151.189.13.35"   || ipList == "195.185.185.195" ||
         ipList == "212.44.160.8") {
            ThreatName = “W32.Netsky @mm”
      }
   }
}
```

Figure 5.9: Threat Pattern of W32.Netsky @mm in Rule-based

```
ThreatName = null;

if(Source-ID == "FTP Server") {
   if(messageCount >= 10 && intervalTime == 300){
      if(message == “530 Login or password incorrect!”) {
         ThreatName = “FTP Brute-force”
      }
   }
}
```

Figure 5.10: Threat Pattern of Brute-force in Rule-based

The answers of the rule-based techniques of two experiments are shown in Table 5.14. The criterion for assessment the answer is also the same as LSA technique. Both two experiments are incorrect answers, because the result of two experiments missed one event for the T1, which is E2 event. The current information from SIM and SEM cannot collect the Event ID number 592. At the first time the rule is defined as “AND” operator so the rules be cannot identify the T1 threat. Thus the rule-based techniques should be revised again by performing “OR” operator at the E2 event. The new revised rules are shown in Figure 5.11.

Table 5.14 The Result of Two Experiments for Two Revisions

Techniques	Experiments	Answer
Rule-baesd Technique (First Revision)	3 Threats	T2, T3 (Incorrect)
	2 Threats	T2 (Incorrect)
Rule-baesd Technique (Boolean Operator Revision)	3 Threats	T1, T2, T3 (Correct)
	2 Threats	T1, T2 (Correct)

```
ThreatName = null;
if(Source-ID == “IDS-SNORT”) {
   if(message == ”TFTP Get” && message == “(portscan) TCP Portsweep”) {
      if(Source-ID == “PC-WinXP-Windows-Security-Event-Logs”) {
          if((event-id == "592" || img-file == "tftp.exe") &&
             (event-id == "592" && img-file == "msblast.exe")) {
             if(Source-ID == "workstation-personal-firewall") {
                if((connection-type == "tcp-inbound" && srcIP-port == "*:135") &&
                   (connection-type == "tcp-inbound" && srcIP-port == "*:444") &&
                   (connection-type == "open udp " && srcIP-port == "*:69")) {
                      ThreatName = “32.Blaster.Worm”
                 }
              }
          }
}}}
```

Figure 5.11: Threat Pattern of 32.Blaster.Worm in Rule-based

5.2 Experiments with the CERT Data Set

From the closed network experiment, the data set is very small and the identification time of two techniques is very short. We need to measure for the efficiency and performance of each technique. Thus, this experiment uses the larger size of data set, which generated from CERT and the results should show more concrete for the computation time. There are ten data sets and their sizes Table 5.15.

Table 5.15 The Size of CERT Data Set

Data Set Release	Size (Compressed)
1	84 MB
2	8.1 GB
3.1	4.0 GB
3.2	4.0 GB
4.1	4.7 GB
4.2	4.5 GB
5.1	11 GB
5.2	11 GB

Table 5.15 The Size of CERT Data Set (cont.)

Data Set Release	Size (Compressed)
6.1	19 GB
6.2	23 GB

The data set used in experiments consists of six difference types from difference sources such as Network Devices, Proxy Server, Workstation, LDAP Server, Email Server, and Files. Figure 5.12 shows a CERT data set structure. The first file is device.csv file. This file contains a list of workstation's connectivity to the network. Next, the http.csv file contains the traffic of HTTP from workstation and it shows the website that users have visited. The third file is Logon.csv. This file is obtained from workstation. It contains the logon and logoff activities. The next file is collected from LDAP server and stored as "YYYY.MM.csv" file. This file name is varied with month and year of data collection. The first four letters represent the year and later two letters are month. The files contain the activities of users who authenticate to the system. It also stores the users' information. The next file is collected from the Email Server. It contains the information of sender and recipient of email. It includes the information of the attachments files, size of email, and the content inside email. The last file was collected from the files that are opened or sent on that PC.

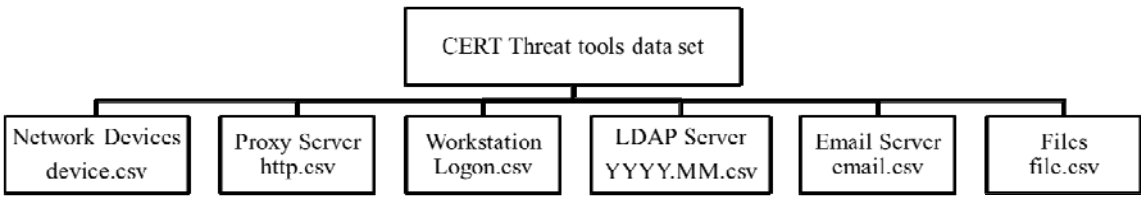

Figure 5.12: A CERT Data Set Structure

The example of each file is shown in the Table 5.16.

Table 5.16 Data Set Files Clarification

Data Set	Attribute	Format	Example Data
Device.csv	ID	String	{S7A7-Y8QZ65MW-8738SAZP}
	Date	Date/Time	01/04/2010 07:12:31
	User	String	DTAA/RES0962
	PC	String	PC-3736
	Activity	String	Connect (Connect/Disconnected)
http.csv	ID	String	{M8H9-W9NL75TH-1322KOLO}
	Date	Date/Time	01/04/2010 07:08:47
	User	String	DTAA/AMA0606
	PC	String	PC-1514
	URL	String	http://groupon.com/1980_.....
	Content	String	alone on they t1517 residual m7 73 sample subsequently changes rods
Logon.csv	ID	String	{Y6O4-A7KC67IN-0899AOZK}
	Date	Date/Time	01/04/2010 00:10:37
	User	String	DTAA/KEE0997
	PC	String	PC-1914
	Activity	String	Logon (Logon/Logoff)
YYYY. MM.csv	employee_name	String	Burton M Stephenson
	User_id	String	BMS0001
	Domain	String	dtaa.com
	Email	String	Burton.M.Stephenson@dtaa.com
	Role	String	Security

Table 5.16 Data Set Files Clarification (cont.)

Data Set	Attribute	Format	Example Data
Email.csv	ID	String	{X5L7-D1UH90YS-1333JYFH}
	Date	Date/Time	01/02/2010 07:14:06
	To	String	Elmo.Brendan.Haynes@dtaa.com
	From	String	Yeo.Xyla.Garner@dtaa.com
	Size	Number	22828
	Attachments	Number	3
	Content	String	... motorscooters cantherigerus …
File	ID	String	{Q5M2-N0JR22AA-1727ALDU}
	Date	Date/Time	01/02/2010 05:15:35
	User	String	WCR0044
	PC	String	PC-9174
	Filename	String	KQTA0DRL.pdf
	Content	String	25-50-44-46-2D laugh statements references place….

5.2.1 Preprocessing

The data set focus on more usual threat that most of organizations are familiar. It does not concern on more technical, however it is needed to collect various data from many sources to identify the threat. The author of this data set described the scenario of the threat as the following.

1) Threat A: Some user never uses the portable drive or work after hours beginning logging in after hours. He or she uses a portable drive and uploading the files to wikileaks.org website and leaves the organization immediately.

2) Threat B: User opens the jobs finder websites and got some invitation from the competitor. Before leaving the company, he or she uses the portable drive to steal the data. It means he or she frequently uses USB with storage media drive more than common activity.

3) Threat C: Some administrators download the Keylogger software and uses their portable drive to install it in supervisor's machine. Then, later, he uses the collected data from its and login an email as supervisor and send the alarming mass email in the organization. After that he leaves the company shortly.

4) Threat D: User uses another user's computer and he or she finds the unauthorized files and send them to his or her email. This threat was occurred more frequently over three months.

5) Threat E: A user who was layoffs from the company and before user left, he or she upload the document files to Dropbox in order to plan to reuse as personal usage.

The threat's behavior from previous scenario was defined as the Answer file. The Answer file contains the data that transforms the scenario into the true positive information and use this data for query in the data set. However, complete all scenario needs to use the data set in a release number 6. It is almost 42 Gigabytes for the size of compressed data set. It is needed to use high performance computer. This research uses only release number 3.1 and 4.1 for this experiment. The size of the CERT data set with uncompressing for this experiment shows in Table 5.17.

Table 5.17 Size with Uncompressed of CERT Data Set

Data Set Release	Size (Compressed)
3.1	13.8 GB
4.1	15.6 GB

The number of records before reducing by LSA is shown in Table 5.18.

Table 5.18 Number of Records of CERT Data Set

Data Set Release Number	Sources	Non-reduced Matrix	
		Column	Row
3.1	Logon.csv	5	842,109
	Devices	5	417,588
	http.csv	6	23,555,144
	Email.csv	7	1,994,063
4.1	Logon.csv	5	899,118
	Devices	5	437,168
	http.csv	6	29,553,383
	Email.csv	11	2,733,360
	File.csv	6	414,556

The data set release number 3.1 and 4.1 contain the first three threats, so only threat A, B, and C are needed to clarify.

Table 5.19 Patterns of Threat A and B for CERT Data Set Release No.3.1

Threat	Source	Information
A	Logon.csv	Activity = “Logon”
		Activity = “Logoff”
	Devices.csv	Activity = “Insert”
		Activity = “Remove”
	http.csv	url = “http://wikileaks.org”
B	http.csv	url = “*job*”
	Device.csv	Activity = “Connect”
		Activity = “Disconnect”
	Email,csv	Recipient = “*@lockheed.com”

Table 5.20 Patterns of Threat A, B and C for CERT Data Set Release No. 4.1

Threat	Source	Information
A	Logon.csv	Activity = “Logon”
		Activity = “Logoff”
	Devices.csv	Activity = “Connect”
		Activity = “Disconnect”
	http.csv	url = “http://wikileaks.org”
B	http.csv	url = “*job*”
	Device.csv	Activity = “Connect”
		Activity = “Disconnect”
	Email,csv	Recipient = “*@lockheed.com”
C	Logon.csv	Activity = “Logon”
		Activity = “Logoff”
	Devices.csv	Activity = “Connect”
		Activity = “Disconnect”
	http.csv	url = “*keylogger*”
	File.csv	“4D7W09A2.exe”
	Email,csv	“*@dtaa.com”

5.2.2 Proposed Method Experiments

Before experiments, the threat behavior will be analyzed by using two data sets. All two threats will create the pattern in JSON format. The two patterns are created in JSON format and are shown in Figure 5.13 to Figure 5.15.

```
{
  "Threat-ID": "4788276",
  "Threat-Name": "Threat A DataSet 3.1 & 4.1",
  "behaviors": [{
      "Source-ID": "Logon",
      "info": [
        {"activity": "Logon"},
        {"activity ": "Logoff"}
      ]},{
      "Source-ID": "Devices",
      "info": [
        {"activity ": "Connect”},
        {"activity ": "Disconnect"}
      ]},{
      "Source-ID": "HTTP",
      "info": [{
          "url": "http://wikileaks.org"
        },] } ] }
```

Figure 5.13: Threat Pattern of Threat A (Data Set Release No.3.1 and 4.1)

```
{
  "Threat-ID": "4788276",
  "Threat-Name": "Threat B DataSet 3.1 & 4.1",
  "behaviors": [{
      "Source-ID": "email",
      "info": [
        {"Recipient": “*@lockheed.com”},
      ]},{
      "Source-ID": "Devices",
      "info": [
        {"activity ": "Insert”},
        {"activity ": "Remove"}
      ]},{
      "Source-ID": "HTTP",
      "info": [{
          "url": "*job*"
        },]
    }]
}
```

Figure 5.14: Threat Pattern of Threat B (Data Set Release No.3.1 and 4.1)

```
{
  "Threat-ID": "4788276",
  "Threat-Name": "Threat A DataSet 3.1 & 4.1",
  "behaviors": [{
      "Source-ID": "Logon",
      "info": [
        {"activity": "Logon"},
        {"activity ": "Logoff"}
      ]},{
      "Source-ID": "Devices",
      "info": [
        {"activity ": "Connect”},
        {"activity ": "Disconnect"}
      ]},{
      "Source-ID": "HTTP",
      "info": [{
          "url": "http://wikileaks.org"
        },] } ] }
```

Figure 5.15: Threat Pattern of Threat C in JSON Format (Data Set Release No 4.1)

Table 5.21 Patterns of Two Threats in the Data Set Release No.3.1

Data Set Release Number	Threat	Source	Information	Event
3.1	A	Logon.csv	Activity = “Logon”	E3A1
			Activity = “Logoff”	E3A2
		Devices	Activity = “Connect”	E3A3
			Activity = “Disconnect”	E3A4
		http.csv	url = “http://wikileaks.org”	E3A5
	B	http.csv	url = “*job*”	E3B1
		Device.csv	Activity = “Connect”	E3B2
			Activity = “Disconnect”	E3B3
		email	Recipient = “*@lockheed.com”	E3B4

Table 5.22 Patterns of Two Threats in the Data Set Release No.4.1

Data Set Release Number	Threat	Source	Information	Event
4.1	A	Logon.csv	Activity = “Logon”	E4A1
			Activity = “Logoff”	E4A2
		Devices	Activity = “Connect”	E4A3
			Activity = “Disconnect”	E4A4
		http.csv	url = “http://wikileaks.org”	E4A5
	B	http.csv	url = “*job*”	E4B1
		Device.csv	Activity = “Connect”	E4B2
			Activity = “Disconnect”	E4B3
		email	Recipient = “*@lockheed.com”	E4B4
	C	Logon.csv	Activity = “Logon”	E4C1
			Activity = “Logoff”	E4C2
		Devices	Activity = “Connect”	E4C3
			Activity = “Disconnect”	E4C4
		http.csv	url = “*keylogger*”	E4C5
		File.csv	Filename = “4D7W09A2.exe”	E4C6
		Email,csv	Recipient = “*@dtaa.com”	E4C7

Next, we create a threat pattern from the threat’s characteristic. Table 5.23 and Table 5.24 show the results from two threats for Data Set release number 3.1 and 4.1.

Table 5.23 Event-Threat Matrix of Data Set Release No.3.1

Event / Threat	Data Set Release No.3.1		
	A	**B**	**A & B**
E3A1	1	0	1
E3A2	1	0	1
E3A3	1	0	1
E3A4	1	0	1
E3A5	1	0	1
E3B1	0	1	1
E3B2	0	1	1
E3B3	0	1	1
E3B4	0	1	1

Table 5.24 Event-Threat Matrix of Data Set Release No.4.1

Event / Threat	Data Set Release No.4.1						
	A	**B**	**C**	**A & B**	**A & C**	**B & C**	**A & B & C**
E4A1	1	0	0	1	1	0	1
E4A2	1	0	0	1	1	0	1
E4A3	1	0	0	1	1	0	1
E4A4	1	0	0	1	1	0	1
E4A5	1	0	0	1	1	0	1
E4B1	0	1	0	1	0	1	**1**
E4B2	0	1	0	1	0	1	1
E4B3	0	1	0	1	0	1	1
E4B4	0	1	0	1	0	1	1
E4C1	0	0	1	0	1	1	1
E4C2	0	0	1	0	1	1	1
E4C3	0	0	1	0	1	1	1

Table 5.24 Event-Threat Matrix of Data Set Release No.4.1 (cont.)

Event / Threat	Data Set Release No.4.1						
	A	**B**	**C**	**A & B**	**A & C**	**B & C**	**A & B & C**
E4C4	0	0	1	0	1	1	1
E4C5	0	0	1	0	1	1	1
E4C6	0	0	1	0	1	1	1
E4C7	0	0	1	0	1	1	1

Then we convert the threat pattern to vector format by using the LSA technique. The vector value of each threat is shown in Table 5.25.

Table 5.25 Query Vector for Experiments in the Data Set Release No.3.1 and 4.1

Data Set Release Number	Threat	Vector Value	
3.1	A	-0.473523	-0.665163
	B	-0.339287	0.742665
	A & B	-0.812810	0.077502
4.1	A	-0.180849	-0.436853
	B	-0.136517	-0.167727
	C	-0.287577	0.523953
	A & B	-0.317366	-0.604580
	A & C	-0.468426	0.087100
	B & C	-0.424094	0.356225
	A & B & C	-0.604943	-0.080627

Every record inside the CERT data set is clarified the date and time. Then use the Date/Time attribute to indicate the time interval for collecting the data. Do the experiments by collecting the data from CERT data set in every hour. Then, we assign the weight of "1" for matched threat patterns. Table 5.26 and 5.27 show the results from SIM and SEM.

Table 5.26 Matrix of SIM and SEM in CERT Data Set Release No. 3.1

Event / Hosts	Interval					
	1st	2nd	3rd	4th	…	nth
E3A1	0	0	0	0	1	1
E3A2	0	0	0	0	0	0
E3A3	0	0	0	0	1	1
E3A4	0	0	0	0	1	1
E3A5	1	0	1	0	1	0
E3B1	0	1	0	0	1	1
E3B2	0	1	0	0	1	1
E3B3	0	1	0	0	1	1
E3B4	0	1	0	0	1	1

Table 5.27 Matrix of SIM and SEM in CERT Data Set Release No. 4.1

Event / Hosts	Interval					
	1st	2nd	3rd	4th	…	nth
E4A1	0	0	0	0	1	1
E4A2	1	0	1	0	0	0
E4A3	0	0	1	0	1	1
E4A4	0	0	0	1	1	1
E4A5	1	0	1	0	1	0
E4B1	0	1	0	0	1	1
E4B2	0	1	0	0	1	1
E4B3	0	1	1	0	1	1
E4B4	0	1	0	0	1	1

Table 5.27 Matrix of SIM and SEM in CERT Data Set Release No. 4.1 (cont.)

Event / Hosts	Interval					
	1st	2nd	3rd	4th	…	nth
E4C1	0	0	0	0	1	1
E4C2	0	1	0	1	1	1
E4C3	1	0	1	0	1	0
E4C4	0	1	0	0	1	1
E4C5	0	1	1	1	1	1
E4C6	1	1	0	1	1	1
E4C7	0	1	0	1	1	1

After that the matrix from Table 5.26 and 5.27 are used to convert vector value. Table 5.28 and Table 5.29 show the query vector with k = 2. This matrix will create in every interval, which defined as an hour.

Table 5.28 Query Vector for Some Intervals of Data Set Release No. 3.1

Event	Weight
E3A1	1
E3A2	1
E3A3	1
E3A4	1
E3A5	1
E3B1	1
E3B2	1
E3B3	1
E3B4	1
Query Vector	-0.812810, 0.077502

Table 5.29 Query Vector for Some Intervals of Data Set Release No. 4.1

Event	Weight
E4A1	1
E4A2	1
E4A3	1
E4A4	1
E4A5	1
E4B1	1
E4B2	1
E4B3	1
E4B4	1
E4C1	1
E4C2	1
E4C3	1
E4C4	1
E4C5	1
E4C6	1
E4C7	1
Query Vector	-0.563861, -0.155478

Then we rank the similarity values in descending order for both data sets. The results are shown in Table 5.30 and Table 5.31.

Table 5.30 Threat Ranking Result from Data Set Release No.3.1

Ranks	Threat	Similarity Value
1st	A & B	1.000000
2nd	B	0.500001
3rd	A	0.499999

Table 5.31 Threat Ranking Result from Data Set Release No.4.1

Ranks	Threat	Similarity Value
1st	A & B & C	1.000000
2nd	B & C	0.950379
3rd	B	0.728187
4th	A & C	0.674034
5th	A & B	0.577690
6th	A	0.501213
7th	C	0.361119

Then adjust the final results by the average threshold value, and theses two values are shown in Table 5.32.

Table 5.32 Threshold Values

Data Set Release No.3.1	Data Set Release No.4.1
0.666666667	0.892855333

Ranks are adjusted by the thresholds value from Table 5.32. The ranks that have the similarity lower than the threshold will be eliminated.

Table 5.33 Threat Ranking Result from Data Set Release No.3.1

Ranks	Threat	Similarity Value
1st	A & B	1.000000

Table 5.34 Threat Ranking Result from Data Set Release No.4.1

Ranks	Threat	Similarity Value
1st	A & B & C	1.000000
2nd	B & C	0.950379

5.2.3 Experiments from Rule-based Techniques

For the next step, we compare with another tradition methods including the rule-based technique. The rule based is implemented based on the JAVA syntax and all three threats need to clarify following the Figure 5.16 to Figure 5.18.

```
ThreatName = null;

if(Source-ID == “Logon”) {
   if(activity[] == ”Logon” && activity[] == ”Logoff”) {
      if(Source-ID == “Device”) {
         if(activity[] == ”Connect” && activity[] == ”Disconnect”) {
            if(Source-ID == “http”) {
               if(url == “http://wikileaks.org”) {
                  ThreatName = “ThreatA”;
               }
            }
         }
      }
   }
}
```

Figure 5.16: Threat Pattern of Threat A in Rule-based

```
ThreatName = null;

if(Source-ID == “email”) {
   if(recipient[] == ”*@lockheed.com”) {
      if(Source-ID == “Device”) {
         if(activity[] == ”Connect” && activity[] == ”Disconnect”) {
            if(Source-ID == “http”) {
               if(url == “*job*”) {
                  ThreatName = “ThreatB”;
               }
}}}}}
```

Figure 5.17: Threat Pattern of Threat B in Rule-based

```
ThreatName = null;

if(Source-ID == “Logon”) {
   if(activity[] == ”Logon” && activity[] == ”Logoff”) {
      if(Source-ID == “Device”) {
         if(activity[] == ”Connect” && activity[] == ”Disconnect”) {
            if(Source-ID == “http”) {
               if(url == “*keylogger*”) {
                  if(Source-ID == “file”) {
                     if(filename == ” 4D7W09A2.exe”) {
                        if(Source-ID == “email”) {
                           if(Recipient == “*@dtaa.com”) {
                              ThreatName = “ThreatC”;
                           }
                        }
                     }
                  }
               }
            }
         }
      }
   }
}
```

Figure 5.18: Threat Pattern of Threat C in Rule-based

The answers of the rule-based techniques of two data sets are shown in Table 5.35. The criterion for assessment the answer is also the same as LSA technique. Both two experiments are correct answers and no need to revise the rules.

Table 5.35 Result from Two Experiments for Two Release Data Sets

Data Set Release No.	Answer
3.1	A, B (Correct)
4.1	A, B, C (Correct)

5.3 Variation of Threats

Actually in the real situation, the administrator does not know the threat's behavior before. The best thing that he or she can make is creating the system that monitors all assets and updates the knowledge of the vulnerability of the system. The variation of threats mentioned before are also included in experiments presented in this section.

This experiment adapted from the previous experiment. The name of website under threat A is changed from wikileage.org to ict.mahidol.ac.th. Table 5.36 shows the characteristic of threat A revision 2.

Table 5.36 Patterns of Threat A Revision 2

Threat	Source	Information
A Rev.2	Logon.csv	Activity = "Logon"
		Activity = "Logoff"
	Devices.csv	Activity = "Insert"
		Activity = "Remove"
	http.csv	url = "http://ict.mahidol.ac.th"

5.3.1 Variation of Threat with LSA Technique

The LSA technique provides the ranking of results by looking the similarity value. So the result from LSA, it shows the most relevant threat. The revision 2 of Threat A is the variation from the original Threat A. So the result should rank the threat A appears on the first rank. So, the approach we use the same as the previous experiment. Table 5.37 and Table 5.38 are shown the query vector from SIM and SEM.

Table 5.37 Query Vector for Some Intervals of Data Set Release No. 3.1

Event	Weight
E3A1	1
E3A2	1
E3A3	1
E3A4	1
E3A5	1
E3B1	1
E3B2	1
E3B3	1
E3B4	1
Query Vector	-0.812810, 0.077502

Table 5.38 Query Vector for Some Intervals of Data Set Release No. 4.1

Event	Weight
E4A1	1
E4A2	1
E4A3	1
E4A4	1
E4A5	0
E4B1	1
E4B2	1
E4B3	1
E4B4	1
E4C1	1
E4C2	1
E4C3	1
E4C4	1
E4C5	1

Table 5.38 Query Vector for Some Interval of Data Set Release No. 4.1 (cont.)

Event	Weight
E4C6	1
E4C7	1
Query Vector	-0.568773, 0.006743

The results from experiment are shown in the Table 5.39 and Table 5.40.

Table 5.39 Threat Ranking Result from Data Set Release No.3.1

Ranks	Threat	Similarity Value
1st	A	-0.165983
2nd	B	-0.432642
3rd	A & B	-0.432642

Table 5.40 Threat Ranking Result from Data Set Release No.4.1

Ranks	Threat	Similarity Value
1st	A & B & C	0.989599007
2nd	B & C	0.985246569
3rd	A & C	0.773288026
4th	B	0.622019157
5th	C	0.491509798
6th	A&B	0.454260803
7th	A	0.371520384

5.3.2 Variation of Threat with Rule-based Technique

The answers of the rule-based techniques from two data sets are shown in Table 5.41. The criterion for assessment the answer is also the same as LSA technique. Both experiments result in incorrect answers, because the result of two experiments missed one event for the A, which is E4A5 event. At the beginning, the rule is defined as “AND” operator so the rules cannot identify the A revision 2 threat. Thus the rule-based techniques should be revised again by performing “OR” operator at the E4A5 event.

Table 5.41 The Result of Two Experiments for Two Revisions

Techniques	Experiments	Answer
Rule-baesd Technique (First Revision)	3.1	B (Incorrect)
	4.1	A, B, C (Incorrect)
Rule-baesd Technique (Boolean Operator Revision)	3.1	A, B (Correct)
	4.1	A, B, C (Correct)

5.4 Discussion

There are three criteria that needed to consider and commpare the LSA technique among other techniques.

5.4.1 Precision and Recall

The result of LSA technique shows the relevant and non-relevant threat. However, actual result will retrieve the non-relevant in the ranks result. For example the number of threat is 10,000, there are 9,999 threats that are not relevant but they are retrieved in the rank. So the identification is not accurate for these experiments. The solution is to use the Precision and Recall in Information Retrieval area. However, they are difference from the Precision and Accuracy in statistics.

Table 5.42 Contingency Table

	Relevant	**Nonrelevant**
Retrieved	true positives (tp)	false positives (fp)
Not retrieved	false negatives (fn)	true negatives (tn)

The equation of Precision and Recall is shown in the equation below.

$$Precision = tp / (tp + fp)$$

$$Recall = tp / (tp + fn)$$

According to the rank result with threshold value, there is some rank that was eliminated and remains results will use them for indicating the precision and recall.

Table 5.43 Precision and Recall of Experiment with Closed Network Environment

Techniques	**Experiments**	**Precision**	**Recall**
Latent Semantic	3 Threats	1 / 1 + 2 = 0.333	1 / 1 + 0 = 1.000
Analysis	2 Threats	1 / 1 + 3 = 0.250	1 / 1 + 0 = 1.000
Rule-baesd Technique	3 Threats	0 / 0 + 2 = 0.000	0 / 0 + 1 = 0.000
(Non-revised)	2 Threats	0 / 0 + 1 = 0.000	0 / 0 + 1 = 0.000
Rule-baesd Technique	3 Threats	1 / 1 + 0 = 1.000	1 / 1 + 0 = 1.000
(Revised)	2 Threats	1 / 1 + 0 = 1.000	1 / 1 + 0 = 1.000

Table 5.44 Precision and Recall of CERT Data Set Release No 3.1 and 4.1 (Non-revised)

Techniques	Release No.	Precision	Recall
Latent Semantic Analysis	3.1	1 / 1 + 0 = 1.000	1 / 1 + 0 = 1.000
	4.1	1 / 1 + 1 = 0.500	1 / 1 + 0 = 1.000
Rule-baesd Technique	3.1	1 / 1 + 0 = 1.000	1 / 1 + 0 = 1.000
(Non-revised)	4.1	1 / 1 + 0 = 1.000	1 / 1 + 0 = 1.000

The variation of threat was experiments in the CERT data set. The threat A will define as Threat A revision 2. The result when the threat A is appears in the list is acceptable. It is the true positive. The precision and recall of the experiments are shown in the Table 5.45.

Table 5.45 Precision and Recall of CERT Data Set Release No 3.1 and 4.1 (Revised)

Techniques	Release No.	Precision	Recall
Latent Semantic Analysis	3.1	1 / 1 + 0 = 1.000	1 / 1 + 0 = 1.000
	4.1	1 / 1 + 1 = 0.500	1 / 1 + 0 = 1.000
Rule-baesd Technique	3.1	0 / 0 + 0 = 0.000	0 / 0 + 1 = 0.000
(Non-revised)	4.1	0 / 0 + 0 = 0.000	0 / 0 + 1 = 0.000
Rule-baesd Technique	3.1	1 / 1 + 0 = 1.000	1 / 1 + 0 = 1.000
(Revised)	4.1	1 / 1 + 0 = 1.000	1 / 1 + 0 = 1.000

The LSA has the high precision when compare with the non-revised rule-based technique. So after revision of the rule-based technique the precision value of rule-based is higher than the LSA technique. Both techniques performed the high recall value.

5.4.2 Efficiency

The proposed technique also works closely to the rule-based technique. However, the main difference from the traditional technique is the complexity of rules. The rule is needed to define all possible cases in rule-based technique. However, our proposed technique, the administrator needs to clarify only one rule and needs some clarify of all possible threats being attack. The number of possible threat being attack does not need to perform checking again. It can use the number from another threat that contains only one threat only (AND operator). From the first experimental, the rules of three threats are expanding into all possible cases with Boolean operators. The numbers of rules are presented in Table 5.46.

Table 5.46 Example of Operations in Closed Network Experiment

Threat	Number of rules (First Revision)	Number of rules (Boolean Operator Revision)
32.Blaster.Worm	1	256
W32.Netsky @mm	1	2
Brute force Attack	1	2
Total	3	260

Table 5.47 Example of Operations with CERT Data Set (Known Threat)

Techniques / Types	No. of Threat	Number of comparing operation	Number of all possible Set of Rules (Threats)
LSA	3	12	7
technique	2	10	3
Rule-based technique	3	12	3
(non-revised)	2	10	2
Rule-based technique	3	2,048	3
(Revised)	2	256	2

Comparing to CERT data set experiment, the rules are needed to clarify for all possible cases in rule-based technique. Next, we conduct the experimental again and the results are shown in Table 5.47 and Table 5.48.

Table 5.48 Example of Operations between Two Techniques with CERT Data Set (Unknown Threat)

Techniques / Types	Release No.	Number of comparing operation	Number of all possible Set of Rules (Threats)
LSA technique	3.1	9	3
	4.1	16	7
Rule-based technique (non-revised)	3.1	9	2
	4.1	16	3
Rule-based technique (Revised)	3.1	256	2
	4.1	32,768	3

The proposed technique uses less number of checking. However, the number of threats is more than rule-based. This means that it needs more memory with LSA technique. Thus the rule-based needs more CPU usage for matching.

Inside the correlation technique, the rule-based uses a simple comparison of current data with the defined rules. It does not need to use a complex computation. However, LSA needs to perform a matrix computation, and vector computation. It includes the SVD approach that uses high computation resource and need more memory. The summarizations of efficiency are shown in Table 5.49.

Table 5.49 The Comparison of Efficiency

Techniques / Types	Computation	CPU Usage	Memory Usage
LSA technique	Complex	High	High
Rule-based technique	Simple	Medium	Low

5.4.3 Time Efficiency

The complexity of the computation of LSA techniques has been already mentioned in previous section. In the experiments, the identification processing time was measure from the starting time of network until the system shows the results. The experiments are conducted on both the proposed techniques and rule-based technique that contains a revised rule. The time interval for the experiments is fifteen minutes.

The Hardware specifications for implementation are shown in Table 5.50. However, the three experiments use difference machines for computation.

Table 5.50 The Computer Specification for Two Experiments

Experiments	Specification	
Closed Network	Model	DELL OptiPlex 755
	CPU	Intel Core 2 Duo E6550 2.33 GHz
	RAM	DDR2 1GB PC5300 (667 MHz)
	Hard Disk	160 Gigabytes (7200 rpm)
	OS	Windows 7 Enterprise
CERT Data Set	Model	Apple iMac 27-inch, Mid 2011
	CPU	Intel Core i5 2500s 2.7 GHz
	RAM	DDR3 12GB PC10600 (1333 MHz)
	Hard Disk	1 Terabyte (7200 rpm)
	OS	OS X 10.9.4 (Mavericks)

For software specification, the correlations obtained from all three experiments are computed by using the same tools with the following specifications.

1) Computer Language, the software consists of Java Language for implementation. The SDK Version of JAVA is Java SE 7 (update 21)

2) Database tools, this experiments use MongoDB 2.4 for keeping the threat’s patterns, raw data set, and

3) Software library, the LSA technique requires a complex computation of LSA, SVD, and VSM. The tools for implementing LSA and SVD are S-Space from Natural Language Processing group at UCLA led by David Jurgens, Keith Stevens, and Michael Dyer. Another technique is implemented by simple comparison, which does not need another library for computing the complex calculations.

The execution times for the first experiment (Closed Network) are shown in Table 5.51. The LSA technique is slower than rule-based technique about 126 seconds. Actually after starting attack, all threats need to have a time, dormant period, before activation to spread out. Table 5.52 shows the comparison of processing time among the three identification techniques.

Table 5.51 Comparison of Processing Time (Closed Network)

Techniques / Types	Duration
LSA technique	13 Minutes 14 Seconds
Rule-based technique	11 Minutes 8 Seconds

The computation time of CERT data set shows in the Table 5.52.

Table 5.52 Comparison of Processing Time (CERT Data Set)

Techniques / Types	Release No.	Number of comparing operation	Duration Time
LSA technique	3.1	9	49:14 Hours
	4.1	16	52:31 Hours
Rule-based technique (non-revised)	3.1	9	16:15 Hours
	4.1	16	17:08 Hours

Table 5.52 Comparison of Processing Time (CERT Data Set) (cont.)

Techniques / Types	Release No.	Number of comparing operation	Duration Time
Rule-based technique	3.1	256	20:42 Hours
(Revised)	4.1	32,768	20:57 Hours

CHAPTER VI
CONCLUSIONS

This chapter analyzes the result of the experiment to find out a conclusion.

6.1 Conclusions

In this research work, we have introduced an alternative approach for identifying the threat in a network by using LSA. This technique provides the correct results in all experiments conducted. The accuracy of LSA technique is not change even the information was changed by dimension reduction. This work uses the Precision and Recall to measure the accuracy of the proposed and rule-based techniques. The LSA also retrieved the incorrect answer in the rank but it was eliminated by the threshold value. The reason behind is the result from the rule-based is a certain results while the result from LSA is an uncertain results. Using Precision and Recall or Accuracy rate may not suitable for LSA. However, the result which appears on the first rank may be acceptable.

For the efficiency, the LSA consumes more computer resources than the rule-based technique. However, the rule-based, in both experiments, needs to revise the rules already defined. In the closed network experiment, the SIM and SEM missed one event. The rule-based is defined with “AND” operator. So It needs to revise the rules with “OR” operator in order to get the more accurate result. So the number of comparison operations in rule-based is more than that of the LSA technique. The LSA needs to defined only one pattern. In conclusion, the LSA is very simple for human defining the rules. But it consumes a lot of resource for identifying the threats. However, the rule-based needs to clarify all possible rules with Boolean operators but consumes less computing resource.

For time efficiency, a large data set shows that LSA is slower than rule-based technique. The main reason is that the rule-based uses only the basic matching method for every interval. While the LSA uses complicate computations for the vector values in every interval. The LSA approach also performed a lot of computation such as Matrix Multiplication, Dot Product, and etc. So, from the experimental result using the CERT data set, it shows the difference of time efficiency between LSA and rule-based techniques in threat identification.

6.2 Future works

The LSA approach may be improved for better efficiency. In the ranking step, there is a non-relevant case appearing on the second rank. We can consider the adjustment for suitable threshold value in order to obtain more accurate results. Suppose it is defined with the too high value, it may remove the relevant results. On the contrary, if it is defined with the too low value, it may include a lot of non-relevant. Thus, we need to examine carefully on threshold value setting. We also need to investigate more with a large amount of events and threat to confirm the result accuracy.

For alternative new approaches, other weighting techniques should be explored in order to enhance threat identification, e.g., the use of relevance feedback from the users for adjusting the query and the results.

REFERENCES

1 Bray, T. RFC 7159–the javascript object notation (json) data interchange format. 2014.

2 Miller, D., Harris, S., Harper, A., VanDyke, S., Blask, C. Security Information and Event Management (SIEM) Implementation. New York: McGraw Hill Osborne Media; 2010.

3 Karlzen H. An Analysis of Security Information and Event Management Systems. Master of Science Thesis in the Programme Secure and Dependable Computer Systems Management and Analysis Chalmers University of Technology. University of Gothenburg; 2009.

4 Müller, A., Göldi, C., Tellenbach, B., & Plattner, B. Event Correlation Engine. Master of Science Thesis Swiss Federal Institute of Technology Zurich. Spring; 2009.

5 Gabriel, R., Hoppe, T., Pastwa, A., & Sowa, S. Analyzing Malware Log Data to Support Security Information and Event Management: Some Research Results. In: Advances in Databases, Knowledge, and Data Applications; 2009;108-113.

6 Rahayu, S. S., Robiah, Y., Shahrin, S., Zaki, M. M., Irda, R., & Faizal, M. A .Scenario Based Worm Trace Pattern Identification Technique. IJCSIS 2010;7(1):1-9.

7 Parveen, P., Weger, Z. R., Thuraisingham, B., Hamlen, K., & Khan, L (2011). Supervised Learning for Insider Threat Detection Using Stream Mining. In: ICTAI '11 Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence. Washington DC: IEEE Computer Society; 2011:1032-7.

8 Villanueva, Emitzá G., Tobias R., Benoit G. and Newres A. H. D4.1: State-of-the-art of event correlation and event processing. 2010 June 1:[77 Pages]. Available from: http://fastfixrsm.sourceforge.net/fastfix-project/sites/default/files/Deliverables/D4.1_FINAL.pdf. Accessed August 8, 2013.

9 Hanemann A. A Hybrid Rule-Based/Case-Based Reasoning Approach for Service Fault Diagnosis. Advanced Information Networking and Applications AINA;2006:2

10 Stolfo, S., Ohsie, D., Yemini, Y., Yemini, S., & Kliger, S. A coding approach to event correlation. In: Proceedings of the fourth international symposium on Integrated network management IV. United Kingdom. London: Chapman & Hall, Ltd.;1995:266-11.

11 Minaxi G. and Mani S. Preprocessor Algorithm for Network Management Codebook. In: Proceedings of the Workshop on Intrusion Detection and Network Monitoring. Califonia: USENIX Association Berkeley;1999.

12 Michael T. A Survey of Event Correlation Techniques and Related Topics. Georgia Institute of Technology;2002.

13 Dairinram, P., Wongsawang, D., & Pengsart, P. SIEM with LSA technique for Threat identification. In Networks (ICON), 2013 19th IEEE International Conference on. IEEE. 2013.

14 Aswani Kumar Ch. , and Srinivas S. On the Performance of Latent Semantic Indexing-based Information Retrieval. Journal of Computing and Information Technology - CIT;2009:259–264.

15 Jessup, E. R., and Martin, J. H. Taking a New Look at the Latent Semantic Analysis Approach to Information Retrieval. Computational information retrieval;2001:121-144.

16 ISO/IEC 27002 Code of practice. IsecT Ltd. :[1 Screen]. Available from: http://www.iso27001security.com/html/27002.html. Accessed August 14,2013

17 Insider Threat Tools | The CERT Division. Software Engineering Institute, Carnegie Mellon University :[1 Screen]. Available from: https://www.cert.org/insider-threat/tools/. Accessed August 14,2013

18 Hoogstraten J. V., SANS Malware FAQ: What is W32/Blaster worm?:[1 Screen]. Available from: http://www.sans.org/security-resources/malwarefaq/w32_blasterworm.php. Accessed August 4, 2013.

19 Netsky.AB | ESET Threat Encyclopedia. ESET North America :[1 Screen]. Available from: http://www.eset.com/us/threat-center/encyclopedia/threats/netskyab. Accessed August 4,2013

BIOGRAPHY

NAME	Pavarit Dairinram
DATE OF BIRTH	17 May 1986
PLACE OF BIRTH	Surin, Thailand
INSTITUTIONS ATTENDED	Mahidol University, 2005-2008 Bachelor of Science (ICT) Mahidol University, 2009-2014 Master of Science (Computer Science)
HOME ADDRESS	229 Moo 9 Thoongpore Road, Tumbol Nok-Maung, Maung Surin, Surin 32000 0899939293 me@pavarit.com
EMPLOYMENT ADDRESS	229 Moo 9 Thoongpore Road, Tumbol Nok-Maung, Maung Surin, Surin 32000 0899939293
PUBLICATION / PRESENTATION	2013 The 19th IEEE International Conference on Networks (ICON 2013) Publisher: IEEE