ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Layer Based Log Analysis for Enhancing Security of Enterprise Data Center: The case of Ethiopian Education and Research Network

Samuel Getachew Tadesse

A THESIS SUBMITED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCINECE IN COMPUTER SCIENCE

February, 2015

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Layer Based Log Analysis for Enhancing Security of Enterprise Data Center: The case of Ethiopian Education and Research Network

Samuel Getachew Tadesse

Advisor: Dejene Ejigu (PhD)

Signature of the Board of Examiners for Approval:

	Name	Signature
1	Dejene Ejigu (PhD)	_____
2	_____	_____
3	_____	_____

February, 2015

DEDICATED TO:

To my mother and brother

Acknowledgments

First and foremost, from bottom of my heart thanks to God who endowed me with the strength to tackle problems with support and being with me throughout my life. I would also like to express my deepest gratitude to my advisor Dr. Dejene Ejigu in which the thesis work would not be possible without his motivation, enthusiasm, continuous as well as constructive supervision, and encouragements during the entire process of working this research.

I want to extend my appreciation to CEO of EthERNet Mr. Zelalem Assefa, and other members of the organization for providing me necessary information related to this thesis. I also wish to express my gratitude to Addis Ababa University Information Communication Technology office members Mr. Yosef and Mr. Abiy for their active response concerning the research issue in different manner. They were willing to give us an input data that create way to develop the entire system.

My paramount gratitude goes to my mother, Workeye Getachew, my aunt, Fekertu Getachew and my wife Zabish Demeke. I am grateful to have their love, care and lessons they gave me in life. Without their sacrifice, support and encouragement, none of these would become real.

Special thanks go to my undergraduate and graduate friends for sharing knowledge and their continual help from the beginning to the end of this thesis work. Finally, I want to thank my colleagues, families, teachers, friends and others who have contributed in one way or another on successful accomplishment of this thesis work.

Table of Contents

List of Figures

List of Tables

List of Algorithms

ACRONYMS

ANB	Attack Knowledge Base
CE	Central Engine
CEE	Common Event Expression
CEF	Common Event Format
CLF	Common Log Format
EM	Expectation Maximization
EthERNet	Ethiopian Education and Research Network
GFL	Generic Format Language
GLF	Generic Log Format
GLS	Global Local Server
GPS	Global Policy Server
IDMEF	Intrusion detection Message Exchange Format
IDPS	Intrusion Detection and Prevention Systems
IPS	Intrusion Prevention Systems
IPSec	Internet Protocol Security
LFAL	Log File Analysis Language
LFPP	Log File Pre-Processer
LFR	Log File Repository
LPS	Local Policy Server
LR	Log Repository
NREN	National Research and Education networks
SIEM	Security Information and Event Management
SLCT	Simple Log Clustering Tool
SNMP	Simple Network Management Protocol
SOM	Self Organized Map
SOTM	Scan Of The Month
UHAD	Unsupervised Heterogeneous Anomaly Detection
VA	Vulnerability Assessment

ABSTRACT

The development of various Internet technologies recently leads many organizations to connect their data center with the global networking infrastructure for communication and sharing of resources. This proves the concept of “global village” that can foster corporation among organizations. However, the issues of security are becoming important as society is moving to the digital information age. For enterprises to evaluate, know data center health as well as conduct their business in a secured manner they must incorporate security. Though, existing security systems use limited log files type and formats for analysis, lead to unclear picture for administrators to decide the existence of attacks in their data center infrastructure.

In this thesis, log analysis technique was used to identify intrusions found at different layers of organizations data center through scrutinizing log events recorded by various network devices, applications and others. Log analysis is an approach that provides valuable information by utilizing various collections of log files gathered from critical data center devices. Thus, to discover a wide range of anomalies (attacks) the considerations of heterogeneous log files are basis for analysis and provides surplus amount of information about the health status of the data center. In our work, central engine is composed of two major components to perform log analysis tasks. Those are clustering module and correlation process which is core of the log analyzer and work together with attack knowledge base to identify attacks.

The collected log files are well organized together into common format and analyzed based on their features to identify anomalies. Clustering algorithms such as Expectation Maximization, K-means were used to determine the number of clusters and filter events based on filtering threshold respectively. On the other hand, correlation finds a relationship among log events and investigates new attack definitions. We have evaluated the prototype of our proposed system and obtained an encouraging result. Further study and implementation of log analysis like we developed can significantly enhance data center security of an organization.

Key words: Log analysis, Data center security, Layered security, Attack identification.

Chapter One - Introduction

1.1 Overview

The world is becoming more interconnected with the advent of Internet and new networking technologies. There are large amounts of personal, educational, commercial, military, and governmental information connected to the global networking infrastructures. With this respect, data center plays vital role for the establishment of network communication among enterprises. Since data center houses large volume of valuable information of the organization, proper security is essential [1, 4, 11, 13]. However, security is left aside and becomes great concern for many organizations. This results a bottleneck for data centers growth in terms of producing attacks in the data center. The posed attacks can be targeted from anywhere for various reasons (such as personal, organizational and other) using their own techniques. For organizations to provide secured services accurately and at the required level they need to address security issues adequately.

The data center structure itself provides suitable environment for intruders to create security events. When the architecture of the data center is planned, it is assumed that it can reduce the possible attacks that can be sent across the infrastructure. Understanding the issues associated with security of a data center requires awareness of the factors that contribute to their proliferation [4]. Knowing attacks approach or technique enable administrators to take appropriate security measures on demand. Attacks are unstable in their nature and lead administrators not to bring one size fit solution, in which organizations secure their data center using their own means such as:- physical security, hardening techniques (operating systems, network, and others), firewall, encryption mechanism, others.

The purpose of securing data center is not to limit legitimate users from accessing information rather to prohibit unauthorized users, intruders and hackers [2]. In fact, insufficient security can negatively affect the public trust in organizations and willingness to use public services, which affects the development initiatives adversely. This implies that there is a need to integrate security within the organization and create monitoring mechanisms for successful delivery of information and evaluate it through auditing process. Security audit of organizations is

conducted based on their security solutions, policies, standards, processes and procedures [8, 9, 10] that exist in the different security management domains as defined by ISO. This facilitates to know the existing security status of the data center.

Ethiopian Education and Research Network (EthERNet) is one of National Research and Education networks (NREN) that participates in UbuntuNet alliance for the country's research and educational activities [17]. It aimed to deliver a highly interconnected communication network with good performance for universities and research institutions as a service provider. This enables them to share educational resources as well as to collaborate with research works both within the country and globally at large. There are different services that EthERNet currently offers categorized under four major areas. These includes: data center, video conference, e-library and technical support. From those data center is the central focus of this research work.

The data center network design of EthERNet has been done so far. However, its security was not taken into consideration as a big deal. Hence, security must be part of the current infrastructure for proper functionality of data center. Addis Ababa University (AAU) is one of the members of EthERNet which has its own data center to provide Internet facility and educational resources to its members which faced such security challenges. Thus, the data center security issue of AAU is taken as a case for our thesis work.

The meaning of a framework is a real or conceptual structure intended to serve as a guide for building the architecture of the entire system. Security framework comes from a variety of sources in order to addresses a number of objectives [7, 15, 18]. Hence, our work aimed to provide a framework used to analyze log files for identifying different holes in the data center network and give a clue to prepare corrective measure based on accompanying policies, procedures and standards of the data center.

Security events of interest can be discovered by analyzing several different sources of machine data, including log files [3, 21]. Log files provide us detail occurrences of an event in certain application or device at some time. Therefore, by analyzing them important information about attacks against infrastructures can be discovered. Data center network traffic is also another

means to examine temporal and spatial variation in the link loads [5]. This will help us to assess the security gaps occurred during communication within and out of the enterprise. Therefore, different security measurements can be taken accordingly for better provision of the data centers.

Log files are set of log records collected from log generating devices to provide details about an event. And they are considered as valuable source of information for security management [22, 23, 24, 25, 26]. In order to understand an attack and develop defenses against them, security professionals need to understand detail about the attack. In this regard, log files are important because they entail and help us to identify abnormal activities in the data center environment.

By collecting and analyzing log files, security professionals can determine loopholes in the data center of an organization, and accordingly execute proactive mitigation strategies for better protection of companies' assets against data breaches or attacks.

In general, our work aims at making an investigative analysis to produce log analysis system in layered data center security approach for organizations aligned with their security policies, procedures, and standards. This work also gives directions and references to highest security advantages of organizations continuity.

1.2 Motivation

The motivation for this thesis work is in many organizations having data center especially educational institutions they do not formulate means or technique for organized data center security. Even those institutions which incorporate security under consideration they do not prepare proper evaluation techniques. This situation is a better choice for intruders to perform any kind malicious activities without any challenge.

However, if an administrator who is in charge to analyze on network devices generated log files for monitoring the entire data center environment, the occurrence of an attack will be minimal. The growth of Internet is linearly proportional with the emergence of various attacks which produce burden for administrators to identify security holes over the data center and fix the

problems [27]. Understanding such type of events using log file analysis is a continual process to bring appropriate corrective measure.

Another motivating factor is the availability of various underutilized critical data center devices generated log files which help us to come up with proposed solution. The aim of the system is to produce valuable information and in turn to develop an application of log analysis system. This has a positive implication for knowing the state of data center and its security will be guaranteed.

1.3 Statement of the Problem

Security is a very important aspect of any business. Each day, organizations data center serve huge amounts of sensitive and critical data flow. Losses of corporate data due to data breaches have significant negative impact in the business. In order to protect confidential (sensitive) data, organizations invest a lot for security technologies such as firewalls, intrusion detection and prevention systems, vulnerability scanners, anti-malware systems, and others.

The use of data center in many organizations is for the sake of delivering the service that it offers. Apart from this, data centers face serious problems related to security. Security is one of the issues which need to be addressed in order to gain the intended benefits. It should be examined continuously to identify security holes and design safeguards for them. A reliable, timely and robust solution is required to control security challenges of the enterprise.

Even though, there is no single clearly formulated security attacks and threats identification, analysis and corrective measure approach enterprises use their own solution for their wellbeing. Effective management of data center security is important for the success of organizations. This calls for identification of current security situations in general which helps to remedy weaknesses and harness the expected benefits in the organizations operations.

Organizations take data center security as part of their responsibility but its evaluation is left aside. The growth of IT industry has brought a number of challenges in relation to security of data center in different ways [27]. One of the reasons is the absence of proper, continual and consistent security mechanisms which leads for the occurrence of attacks and threats. Hence,

systematic approach is helpful for enterprises to enforce their policies, procedures and standards properly.

These days every company has log generating devices in their data center to keep track of network activities. Those log files provide security professionals to obtain detail information for the purpose of research in order to analyze security incidents. Consequently, log analysis becomes a key component in security process. The continuous growth of logs, limits organizations to have an ineffective log management system to aggregate, correlate, and analyze log files captured by data center devices.

Enterprises need to have secured and effective log management systems for better protection of their critical assets from cyber attacks. Analysis of heterogeneous log files through considering different security parameters enables an enterprise to have an overall outlook of its data center security status. Security is the state of immunity of enterprise from various internal and external threats to ensure that the most productive use of enterprise resources.

Therefore, our work emphasized to develop log analyzer system to enhance data center security through extraction of information about different emerging security holes within the organization and to create a unified approach for guaranteeing security.

1.4 Objectives

1.4.1 General Objective

The general objective of this research work is to design and develop a layer based log analysis system to enhance an enterprise data center security.

1.4.2 Specific Objectives

The specific objectives devised to accomplish the general objective are:

- Assessing the current trends of data center security and its implication along with potential security attacks, threats and others.
- Analyzing the different ways for occurrence of security holes in the data center using mainly enterprise procedure, policies, standards and others point of view.

- Identifying different security parameter requirements.
- Designing a log analyzer based security model.
- Building a prototype system for the proposed work.
- Testing and evaluating the proposed system.

1.5 Scope and Limitations

The scope of this work is to develop a layer based log file analysis system for organizations. This study focuses on enhancing the security of data center through processing huge amount of log files obtained from critical devices appearing in the network. However, our work is limited to deliver the processing of log files conducted in offline manner. This means that the analyzer processes and tells us about the incident after an event has been taken place.

1.6 Methodology

In order to accomplish the general and specific objectives of this study different methodology will be employed.

Literature review

First, we conducted an extensive review of relevant literatures to acquire a deeper understanding of the research area and its problem domains. Previous researches in the areas of data center security, log file analysis and related issues are investigated to visualize their importance towards this research. Existing works related to this research work are also assessed to identify and point direction in providing solution to identified problems. Moreover, the considerations of metrics for the design of data center security architecture are also reviewed.

Data collection

In our study, data were collected from data center of AAU. Different data collection techniques were applied for successful completion of the study. It includes observing data center environment, interviewing different staff members of institutions, preparing and distributing questionnaires. This let us to examine the existing security trend of the institutions data center and to gather their recommendation for trusted data center.

We also used for our implementation Scan of the Month #34 (SOTM#34)[1] from the honeynet project which has been analyzed and validated by multiple contestants (Andrew[2], Kronberg[3] and Richard[4]) of the challenge are examined [28, 29]. The dataset includes evidence from several logs i.e., Apache Server, Linux syslogs, Snort NIDS and iptables firewall which were recorded by different sources in a honeynet system. Those data enables us to conduct the implementation as well as the testing activity within certain period of time.

Design and Implementation

Having the inputs from conducted review of literatures, useful models and design requirements are then identified to arrive at sensible solutions. Based on these requirements, a model of log analyzer using layered approach of data center security architecture is proposed which can be a guide in implementation of our applications. Later, development tools and techniques specific to the design of the architecture are then labeled for the implementation of the prototype system. The prototype considers the necessary identified activities to prove the importance of the proposed solution so that it should carefully follow the requirement provided in the architecture.

Testing and Evaluation

Finally, in order to validate our prototype and check its functionality, testing and evaluation techniques are identified and used.

1.7 Significance of the Study

The significance of this work is to provide an application in order to preserve the security of data center through determining any kind of incident. The proposed system can also be an input for further works in different domains that requires such analysis. It can be provided as a pre-step in other related studies. For instance, the result of the log files analysis taken as an input for auditing the organization. It helps to strengthen service delivery of governmental or

[1] http://old.honeynet.org/scans/scan34

[2] http://old.honeynet.org/scans /scan34/sols/3/sotm

[3] http://old.honeynet.org/ scans/scan34/sols/2/proc.pdf

[4] http://www.honeynet.org/scans/scan34/sols/1/index.html

nongovernmental organizations and for other domains with a similar setting. There are many reasons to conduct log analysis. Among which are:

- **Forensics Analysis:** Information stored in log files is important to gather forensic evidence of investigated actions or attacks against the system. The log analysis process is an important part of forensics posture. The process of doing forensics using log analysis is similar to using Internet to search information from the web. Sometimes anyone can get exactly what they are looking for or it's a trial-and-error process.
- **Compliance with audit and security policies:** provides a comprehensive review of an organization's adherence to regulatory guidelines. IT consultants evaluate the strength and thoroughness of compliance preparations. Auditors review security polices, user access controls and risk management procedures over the course of a compliance audit.
- **Security investigation:** System and network device log files are essential to incident investigators. During an incident investigation, network administrators should be able to identify which hosts have communicated with which IP addresses and what type of traffic was generated. This situation helps administrators to propose the kind of action to take as a solution to fix the problem.
- **Proactive protection:** The detail analysis of an event enables administrators to prepare an immediate with corrective measures.
- **Determination of the network health:** Due to log file analysis different organizations are beneficial to have a better insight about the current status of their data center health.

In general, our work can be applied to organizations data centers that require their entire business to establish a secured environment and for grubbing more advantages at large.

1.8 Thesis Outline

The remaining of the thesis is organized in the following way. Chapter 2 starts with review of literature on data center and the basics for data center security. Then concepts of defense in depth, layered security, and comparison between them as well as log file analysis are provided. The ways to acquire log files with support of various layered devices with existing log acquisition technologies are next discussed. Approaches to security information event

management techniques for data typically log files from various sources into central repository for analysis are also presented in detail.

Next, review of related works is presented in Chapter 3. In this chapter works that have significant relation with this thesis are assessed. Architectures, and works proposed in other domains but are related to the thesis are taken into consideration.

Chapter 4 concentrated on providing design of log file analyzer system. It focuses on discussing components relevant for the functionality of the system which includes: Log file repository, log file pre-processor, log repository, attack knowledge base, central engine, that are basis for design and implementation of the system.

Chapter 5 presents implementation and discussion of a prototype system based on the proposed architecture. Different results were obtained from the collected data of the university case during testing. In addition, our system is evaluated accordingly.

Finally, Chapter 6 summarizes and the contribution made in the thesis, and critically assesses the shortcomings detected in the course of this research. Furthermore, new issues that have been surfacing while working on the thesis are suggested as future works.

Chapter Two - Literature Review

This Chapter presents a review of literatures conducted in order to understand the problem associated with the realm of the objectives of this thesis and also help to identify appropriate direction. An extensive review is made on areas of data center, data center security, trends for data center security, log file analysis and its approaches as well as discusses technologies that support the effectiveness of the proposed architecture.

2.1 Introduction

Data center is defined as a pool of resources and facility used to house various network components, data storage systems and servers [1, 11, 13]. It often includes backup power supplies, redundant network connection lines, policy based security systems for running the enterprise's core applications as well as environmental conditions control (like air conditioning, humidity control and fire systems). It is a physical or virtual infrastructure used by many enterprises to put their networking systems and components for the company's information technology needs, which typically involve storing, processing and serving large amounts of mission critical data to clients. Typically, data center networking creates a network infrastructure which is [12]:

- Stable, secure and reliable.
- In line with the organization regulations and meets organization customers or users need.
- Supports modern technologies such as virtualization and cloud computing.
- Scalable and can easily meet the requirements of organizations network communications in peak usage.

In today's information era data center represents the core of many organizations for achieving their own business objectives. Organizations highly relied on data stored in their data center to interact with its employees and customers [13, 14]. The components and technologies that make up data center networking generally include:

- Networking equipment (routers, switches, modems, etc.)
- Network cabling (LAN/WAN and network interface cabling)
- Network addressing scheme(IP v4 or IP v6)

- Network security (security protocols/encryption algorithms, firewalls, IDS, etc.)
- Internet connectivity (satellite, DSL, wireless, optical)

Data center is not a choice rather an integral part of modern organizations. It can be considered as an area that holds, a means of hosting critical data, applications, and servers, as well as contains basic assets of customer information, intellectual property, and other business critical data. Communication among organizations is delivered through creating connection within them. However, the proliferation of Internet based technologies makes the data center to be more exposed to security attacks [4]. Security attacks on data center may destroy the whole organization's network and data. With the fast growth of Internet various types of attacks are released to the data center environment and hinder its development [27]. Hence, proper security mechanisms have to be proposed for reliable delivery of services.

Now a day's keeping data center to be secured is such an imperative task. This is due to attacks come from various sources such as hardware failures, software flaws, tentative probing and malicious attacks [24]. In order to overcome those attacks utilizing various data center devices generated information is necessary to produce meaningful security paradigm. Analyzing log files from different devices to detect suspicious activities is one form of defense. However, the entire size of logs and variety of log format makes human log analysis to be difficult. Furthermore, data mining techniques can be applied to carry out the analysis of logs to obtain valuable information [22, 23, 24, 25, 26].

A moderate to large network tends to collect sheer size of network activities and generating huge log files which make human inspection impossible [22]. Traditionally, most log analyzers on the market are based on pattern matching techniques. They compare the log entries to sets of pre defined patterns. These sets of patterns have to be manually updated frequently by security experts to handle all kinds of attacks they know so far. Apparently, those emerging attacks could easily outpace the updating speed for those patterns. This is impractical to handle all today's emerging attacks due to their dynamic nature with traditional approaches of log analyzers.

The layered security architecture is a modern solution to the complex security problems faced in the data center [11, 19, 20]. It consists of a set of tools designed specifically to protect critical

network resources that reside on the data center network. If one of the devices in the layer fails, the next layer will stop the attack and limit the damage that may occur. This implies that applying such architecture can enhance data center security of an organization. In addition, the architecture provides suitable way to obtain required device level information at each layer and combination all are accepted and then processed in log file analyzer.

A log file ensures valuable insight to previous history of data center network usages. Using information from log files can help to improve the security feature of the infrastructure and future development of the infrastructure. However, log files often contain huge amount of data which require significant amount of time for processing. Log analysis is an important way to keep track of computers and networks state [30]. In general, the central focus of this thesis is to enhance the security of data center through analyzing various device generated log files and entail the administrator to monitor the data center accordingly.

2.2 Characteristics of Data Center

The process of building data center facility appears to be simple on the inception but, it has several aspects that must be done correctly. Data centers have a number of unique and important features in which organizations must take into consideration during design their data center [40]. The following are some of the characteristics of data center.

- **Manageability*:*** is core attribute of a data center that should be in the first place. A data center should provide easy and integrated management of all its elements. That can be achieved through automation and reduction of human intervention in common tasks.
- **Availability**: a data center should function and be accessible everyday for assuring the availability of information when ever required. In short, it means that there is no downtime. Unavailability of information leads to loss of information and could cost a lot to the business of an organization.
- **Fault Tolerance:** is the property that enables a data center to continue operating properly in the event of the failure (one or more faults) of its components. If its operating quality decreases at all then, it is proportional to the severity of the failure. Fault tolerance is particularly required after highest availability.

- **Security**: is a notion such that standards, policies and procedures are central component and to meet together to prevent unauthorized access to the information.

- **Scalability:** is a planned, monitored, predictable nature for the growth of data center infrastructure. Business growth is almost in a continual progress that always requires deploying more severs, applications and additional databases etc.
- **Performance:** is a means to measure the state of all elements found in the data center infrastructure to establish a comfortable environment for service delivery. Performance management is to make sure that all the elements of the data center provide optimal functionality at the required level.
- **Capacity:** is a necessity of an organization relay upon their data center to provide the service. When capacity requires increase, the data center must provide additional capacity without interrupting availability or with minimal disruption.
- **Monitoring:** is a continuous process of gathering information on various elements and services running in the data center. The reason is to come up on with predicting unknown events in the data center.
- **Reporting:** is a contextual generation of information about resources performance, capacity and other utilization information gathered together at certain point of time.

2.3 Data Center Security

Data center security is the establishment of practices that makes a data center to be more secured from a range of different kinds of threats and attacks [11, 12, 13]. Data center primary requirement for organizations is to preserve a kind of dedicated security. As data centre becomes more diverse, heterogeneous and flexible, the environment that it operates becomes increasingly open. This area becomes an ideal for illegal intruders to perform their business. Data center security aims to make that data less accessible to hackers or anyone else who may seek unauthorized access.

One type of data center security is physical security. Experts may recommend various kinds of facilities or site security such as setbacks, landscaping, thick walls, and other aspects of a building that will create physical barriers around the data center which is less efficient to control the incidents. Another type of data center security is network security. Because data centers are

served by networks, security experts need to plan adequate protection mechanisms into those network trajectories that run to a data center. That means installing firewalls, antivirus programs, or anything that prevents data breaches or other security issues. Data center security varies according to the type of data center in question. For example, identifying a data center according to levels shows how fault tolerant the infrastructure is and what kind of security it may need. Generally, a lot of administrators tend to recommend redundant utilities for data centers, such as better security strategy, multiple power sources, multiple environmental controls and more.

2.3.1 The Need for Data Center Security

Data centers are the key component of any organizations infrastructure. Since data center comprises various critical devices, applications, and data there is a need for proper and continual security to its existence [20]. Losing data and applications can impact the organizations ability to conduct its business. In the ever changing world more than global data communications, fast paced software development the concern of security is becoming more and more of an issue [7, 27].

Security is now a basic requirement because global computing is inherently insecure. As your data goes from point A to point B on the Internet, for example, it may pass through several other points along the way, giving other users the opportunity to intercept, and even alter it. The meaning of securing data center is real when there is organized way of monitoring the infrastructure such that it can pretend malicious user activities.

The large volume of information and the criticality of the services housed in the data centers make them likely an ideal target. In fact the number of reported attacks, including those that affect data center continues to grow year by year as stated by Computer Security Investigations /Federal Bureau of Investigations (CSI/FBI) 2006 report [31]. Denial of service (DoS), theft of confidential information, data alteration, and data loss are some of the common security problems that affect data center environment. New forms of attacks are continuously developed; both attacks and attack tools are becoming more sophisticated as well as attack frequency is increasing which leads the security of data center to be a critical [3, 27]. This implies that there has to be a planned way of building a security strategy to counteract on the posed attacks.

In order to understand the issues associated with attacks on data center require an awareness of identifying the factors that contribute to their existence. In fact the expansion of Internet and growing complexity of protocols and applications used in the data center results in an increasing number of exploitable vulnerabilities [27]. On the other hand, hackers use the openness of organizations network to communicate and develop automated tools that facilitate the identification and exploitation of those vulnerabilities. Many attack tools are widely available on the Internet and designed to execute highly sophisticated attacks using simple user interface which makes data centers accessible to anyone.

The origination of attacks can be either from internal (local) or external to the organization. Many attacks are mainly initiated by internal trusted personnel according to 2006 CSI/FBI report "computer criminal security survey" [31]. Moreover, studies show that internal attack tends to be more damaging because of its variety and the amount of information available inside organization. The term "hacker" which has traditionally been associated with people external to an organization but now it also includes people internal to the organization. In general, the necessity of data center security is unquestionable for organizations to achieve their business objectives.

2.3.2 Data Center Security Architecture Elements

A foundation task that teams in the data center should do before building an enterprise network is to develop enterprise security architecture [11, 13]. The development is according to the accompanying policy, standards, and procedures of the organization. The aim of enterprise security architecture is to create a conceptual design for the security of the data center elements and prepare security mechanism like security policies and procedures. It can link the components of the security infrastructure into single cohesive unit. This enables organizations to protect and manage its critical information asset easily.

Organizations do not follow a documented security management standards rather choose to write their own. This will flourish inconsistent security implementation to be applied across different organizations. In fact there is no one size fit to all policy that will be well suited for the needs of all organizations. However, organizations have to incorporate enterprise security architecture as part of their business strategy and create consistent service.

Enterprise security architecture ensures confidentiality, integrity and availability throughout the enterprise and aligned it with the organization's business objectives. Once the architecture is completed the next task is to identify security elements to be placed in the data center.

The elements of data center security help to determine the enterprise security issues and isolate attacks. Security architecture enables all of the elements to provide protection of information assets. Failure to include single element for the security of the infrastructure leaves large holes in protection and results in loss of information. The following are major elements for designing data center security architecture.

2.3.2.1 Security policy

Security policy is a formal statement of the rules by which people are given access to an organization's technology and information asset. The creation of enterprise security architecture begins by defining security policy that everyone in the organization accepts and supports [32]. The policy must be enforced through all levels of management extended down to every user. It provides confidentiality, integrity and availability service throughout the enterprise.

In order to develop security policy a though understanding of the data center environment is necessary [19, 20, 33]. This is achieved by analyzing various security attacks, risks, vulnerability and their countermeasure. In security policy development experts first define what needs to be protected then balance the security need with cost and other expenses. In addition, a major consideration of security policy, standard or guideline is that it must be in a manner such that the reader clearly understands its objectives.

2.3.2.2 Basic Data Center Security Technologies

Information stored in data center must be protected from any security attacks as well as associated risk that may destroy or modify it in any unwanted way. Different solutions for the security challenges can be used together to achieve the highest possible protection of valuable information [11]. Some of these technologies include:

- Firewall
- Network Intrusion Detection and Prevention systems.

- Virtual Local Area Network (VLAN).
- Virtual Private Network (VPN) and IPSec.

Firewall: is a device or a software configured to permit, deny or proxy all traffic between different networks that have different security levels typically between an internal network and external network based upon a set of rules and other criteria [11, 34]. A firewall acts as a guard in the network which works in collaboration with several equipments such as: router, gateway server, and an authentication server for providing better protection.

Firewall basically protects sensitive data in the internal network from outside treats and also within the network itself from inside users. There is no host directly accessible from the external network user and no external host is directly accessible by internal network user. During communication all network traffic will be ultimately monitored by the firewall. The placement of firewall depends on organization network design.

The major aspect in the design of any network is how to be connected with the Internet in the secured manner. This task will be achieved by creating what is called a demilitarized zone "DMZ". It is a network that exists between the protected (trusted) and unprotected (un trusted) network. Also the firewall is used to protect the DMZ from any external attack coming from the Internet. Internet users can freely enter the DMZ to access public web servers, but screening firewalls exist at the access point to filter out unwanted traffic, such as floods of packet from hackers who are attempting to deny the service.

A firewall can be integrated into a router or switch in the form of either software based; hardware based or can be implemented as additional component. The firewall basically has three interfaces [11]. These include internal interface, DMZ interface and the external interface. Each of these interfaces is connected to one network either internal, DMZ or external. Also the firewall may act as a proxy that makes any connection request to the Internet on behalf of a host [20]. A firewall always stands between the protected and the unprotected network domain. Figure 2.1 shows the different option for the placement of firewall in an enterprise network.

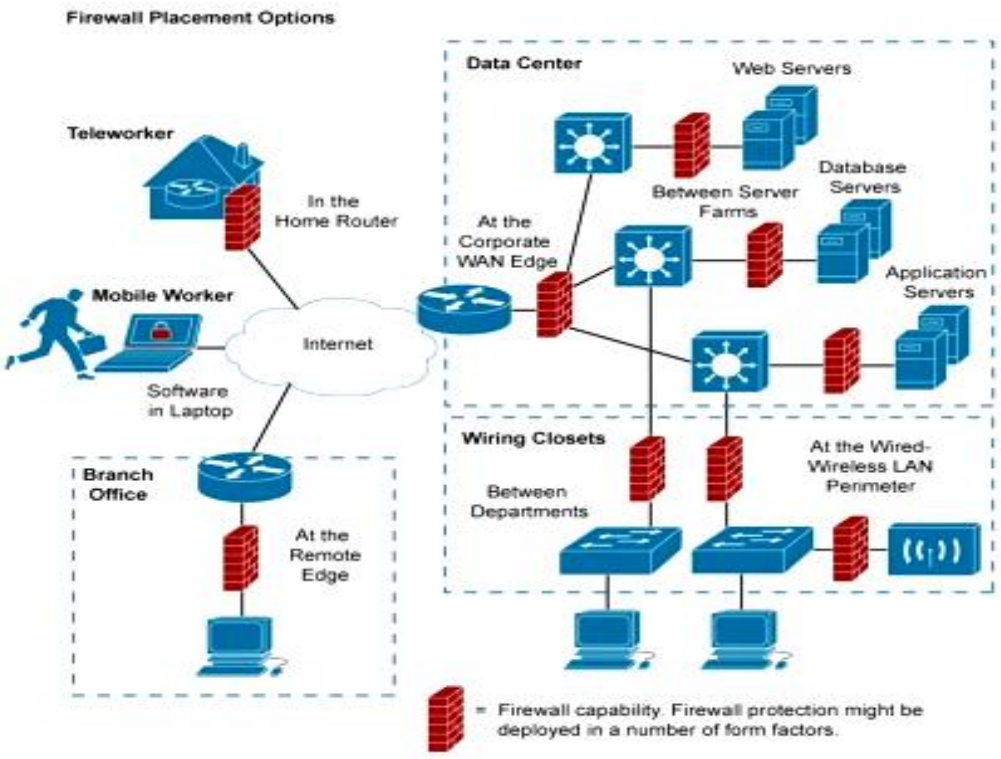


Figure 2.1 Sample Enterprise Firewall Placement Options[5]

Network Intrusion Detection and Prevention systems (IDS/IPS):- are deployed at the data center in order to provide a good level of protection for the data center. They are the second level of protection for server farm components next to the firewall. An IDS has a capability to detect the so called "bad traffic" based on either their signature or protocol anomaly detection. On the other hand, an IPS is not only detects the "bad traffic" but also drops and blocks the connection transporting real bad traffic [23, 24, 35].

IDS sensors can be logically configured to reside behind the firewall. This allows the sensor to avoid network attacks that were not filtered while passing through the firewall. In the server farm, many servers often exist in the same subnet. If one server is compromised, the possibility of other servers as being compromised increases. Alternatively, if the server is secured and uncompromised then the attacker is unable to gain control of the switch, data traffic to and from

[5] http://www.cisco.com/c/en/us/products/collateral/security/ios-firewall/prod_white_paper0900aecd8057f042.html

the server(s). However, IDS does not work well to handle attack patterns that do not match patterns such as zero day attacks.

Virtual Local Area Networks(VLANs):- refers to the ability of switches and routers to allow any random collection of virtual LAN segments within the network to be combined into an independent user group, appearing as a single LAN [11, 36]. It has similar attributes as a physical LAN, but it allows hosts to be grouped together even if they are not located on the same LAN segment. Network configuration can be done through software instead of physically relocating devices.

VLANs offer various benefits such as efficient use of bandwidth, flexibility, scalability, performance, and security. If one of VLAN is compromised by hackers, then it can be isolated from the other network segments to minimize damage. In addition, the traffic between different VLANs can be controlled according to the predefined Access Control List (ACL) at the firewall (no user can access other VLAN unless its VLAN is granted permission to access that VLAN). In some particular situations it is important to control traffic between different hosts belonging to different VLANs. This will help to keep the confidentiality and integrity of the application from malicious users.

Virtual Private Network (VPN) and IPSec: is a private data network that makes use of the public telecommunication infrastructure to maintain confidentiality through the use of a tunneling protocol and security measures [11, 37]. It is more cost effective than dedicated private lines and represents secure way for different corporations to provide users access to the company network and for remote networks to communicate with each other across the Internet.

A VPN basically has two parts. These are: - i) the protected (internal) network, which provides physical and administrative security for secure data exchange. ii) The unsecured network, (usually the Internet). VPN makes it possible to have the same protected sharing of public resources for data. Companies today are using VPNs for both extranets and wide-area intranets.

Many VPN client programs are configured to require that all IP traffic pass through the tunnel (while the VPN connection is active) for increased security. This means that when VPN connection is active, all access outside the secure network must pass through the same firewall

as if the users were physically connected to the inside of the secured network. This reduces the risk of an attacker to gain access to the secured network by attacking the VPN client's host machine.

Tunneling is the transmission of data through a public network in such a way that routing nodes in the public network are unaware that the transmission is part of a private network [37]. Tunneling is generally done by encapsulating the private network data and protocol information within the public network protocol data so that the tunneled data is not available to anyone examining the transmitted data frames. Secured VPNs use cryptographic tunneling protocols to provide the following functionalities:

- Confidentiality (blocking snooping and packet sniffing)
- Origin authentication (blocking identity spoofing)
- Message integrity (blocking message modification)

Secure VPN technologies can also be used to enhance security within the organization's data center network infrastructure. In addition, they include Secure Socket Layer/Transport Layer Security (SSL/TLS), which are cryptographic protocols used to provide a secure connection for the Internet, and IPSec (IP security) [37].

2.3.3 Trends for Data Center Security

2.3.3.1 Defense in Depth

Defense in Depth is a strategy used by many organizations to maintain and protect their information asset [19, 20, 33, 35, 38]. It is used to prevent attackers from getting into the network by putting up multiple barriers around the network to slow down the attack. This strategy was first developed by National Security Agency (NSA) to strengthen and maintain their security. This strategy also brings another idea called "layered security", which uses firewalls and other associated technologies to mitigate and prevent an attack.

Depending on the technology they accommodate defense in depth can be used to defend against, malware, DDOS, spoofing, intruders, and many other types of attacks on a system. This also includes a plan on what would happen if an attack occurred, and what the organization should do for this type of event. The principle of defense in depth is similar to that of layered security which increases security of the data center as a whole. If an attack causes one security

mechanism to fail, the other may still provide necessary security to protect the infrastructure. In general, implementing defense in depth strategy has limitation since it adds complexity in which new protection functionality creates trouble and new risks associated with it. Figure 2.2 shows defense in depth model.

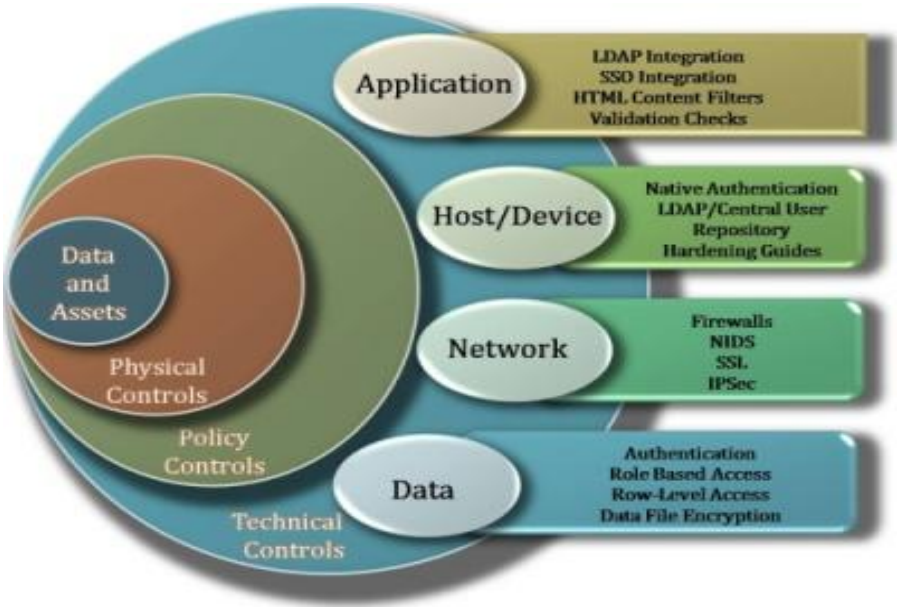


Figure 2.2 Defense in Depth Model[6]

2.3.3.2 Layered Security

Layered security is the best practice, modern solution that aims to prevent intruders or hackers from introducing a kind of malicious activities in the organization at various levels [11, 13]. The concept of "layered security" is if one of the devices operating in the layer fails the next layer will take care and stops an attack. Later on the person who is in charge of the data center security will fix the problems accordingly. Layered security ensures that the network integrity is safe and is protected multiple times at various stages. If firewall was used, for example, and an intruder bypasses this, the network is in danger because a firewall cannot detect intruders once they pass the firewall technology.

[6] http://resources.arcgis.com/en/communities/enterprise-gis/01n200000030000000.htm

Layered security and defense in depth seems similar but are two different concepts with a lot of overlaps. Layered security strategy is extremely important to protect data center resources. On the other hands, defense in depth approach to security widens the scope to security and encourages flexible policy that responds well to new conditions, helping to ensure that are not blindsided by unexpected threats. According to layered security, utilizing firewall and IDS will allow extra layer of protection. IDS can identify threats inside the network and alert administrators about this issue [22, 23, 24, 41]. This is the basic reason why layered security is important in order to preserve the integrity of network and prevent information from leaking out of the network. However, only using two of these technologies will leave a lot of other vulnerabilities opened.

Hence, layered security adds more extra layers compared to defense in depth in which it provides information about the factors by which an intruder comprises the infrastructure to gain unauthorized access. In general, layered security technique gives a capability for administrators to have up-to-date knowledge about their organization data center network. Figure 2.3, shows layered data center security elements.

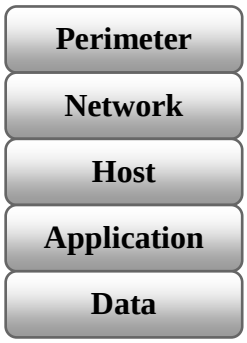


Figure 2.3 Layered Data Center Security Elements

2.3.3.3 Defense in Depth vs. Layered Security

Many people used the terms “layered security” and “defense in depth” interchangeably assuming that they refer the same concept. Though, they seems similar in their context, the application of each is totally different [39]. Layered defense or layered security is a means to layer or cover additional defenses to compensate for the gaps of other security measures at each layer.

On the other hands, defense in depth was originally a military concept which is similar to layered security but it addresses the strategy of network defense as opposed to the actual defense of attacks. Basically, the use of firewalls, IDPS and antivirus software are components of layered defense. Layered defense itself is one component of a defense in depth strategy. According to Chad Perrin [40] layered security is extremely important to protect ICT resources. A defense in depth approach to security widens the scope of attentions to security and encourages flexible policy that responds well to new conditions.

2.3.4 Security Attacks and Threats to Data Center

Today is an information era in which Internet comes up with new things drastically. Personal, governmental, and business applications continue multiply to put their information asset and offer services using Internet, to provide immediate benefits to end users. However, various devices, applications and services are ideal target to create security threats and attacks in the data center at individual or organizational level which leads to lose valuable information resources. In fact, information is an asset that must be protected.

Without adequate security, many individuals and organizations will be at risk to lose their own asset. It is known that the appearance for any kind of events or incidents is part of any data center. Hence, in this research the following terms are often used in relation to define the context of security in data center area.

- **Threat**: is an event that poses some harm to the data center.
- **Vulnerability:** is a deficiency or weakness on a system, application, service, device, or resource whose exploitation lead to the materialization of the treat.
- **Attack:** is the actual exploitation of vulnerabilities to make a threat to happen in reality.

There are a number of different ways attackers use to gain control of sensitive data and information. Microsoft TechNet provides a list of the common types of network attack to guard against [42]. Table 2.1 summarizes some of the common attacks.

Table 2.1 Major Classes of Attacks

Attack Name	Attack Description
Eavesdropping	Means to listen or sniff the network without the consent of users' knowledge and allows eavesdropper to monitor the network. Ensuring strong encryption help to preserve when data traverses the network.
Data Modification	Attackers can modify the data in the packet without the consent of the sender or receiver.
IP Address Spoofing	Enable attackers to falsely assume an organization IP address allowing them to modify, reroute, or delete your data.
Password Based Attacks	Attacker finds a valid user account and has the same rights and privileges as the real user.
Denial-of-Service(DoS) Attack	Prevents normal use of network by valid users by flooding a computer or the entire network with traffic until a shutdown occurs because of the overload

2.3.5 Layered Security for Data Center

Data center security is now a serious concern for enterprises, government agencies, and organizations of all sizes. Today's advanced attacks from cyber-terrorists, disgruntled employees, and hackers demand a methodical approach to maintain data center security. In many organizations an enhanced security is not a preference rather it's mandatory.

Layered security architecture provides a scalable and modular approach for deploying security between multiple data center tiers [35, 39, 40]. Data center security is based on effective security policy that defines the connection, access requirements to protect resources from internal and external threats and to ensure data privacy and integrity.

The layered approach is both a technical and scalable strategy that ensures adequate measures to be put in place at different levels within the network infrastructure of an organization [11]. It basically focuses on maintaining appropriate security measures and procedures at five different levels within the data center environment. The layered security elements are shown above in

Figure 2.3. Those levels include: perimeter, network, host, application and data level security. The detail description of each level of layered security is discussed below as follows.

Perimeter level security: is the first line of defense for any security threats coming from an untrusted or external network [11, 39]. The perimeter is an area where the internal network ends and the external network (Internet) begins. It consists of one or more firewalls and protected servers located in the Demilitarized Zone (DMZ). The DMZ contains web servers, email gateways, network antivirus, and DNS servers which are exposed to the Internet attacks. The firewall has strict rules about what can enter inside the network as well as rules about how servers in the DMZ can interact with the Internet and the inside network.

The perimeter, in short, is a gateway to the outside world and, conversely, the outside world's gateway to organizations network. A compromised perimeter can adversely challenge an organization's ability to conduct its business. In general firewall, network based antivirus and Virtual Private Network provide security at the perimeter level.

Network level security: is the second level of the layered-security model which refers to internal or Local Area Network (LAN) and outside or Wide Area Network (WAN) security. An internal network may include desktops and application servers or may be more complex with point-to-point frame relay connections to remote machines.

Most networks today are fairly open behind their perimeter; once intruders are inside, then they can travel across the network with no restriction. This is especially true for most small to medium size organizations, which makes them an ideal targets for hackers and other malicious individuals. In general, Intrusion Detection systems (IDs) and Prevention systems (IPs), Network vulnerability assessment (VA), and Access Control (authentication) provide security at this level.

Host level security:-is the third level in layered security model, which emphasizes in the protection of the individual devices, such as servers, desktops, switches, routers, etc., on the network. Each of the devices has a number of configurable parameters. These parameters include registry settings, services (applications) operating on the device, or patches to the operating system or important applications.

But if an administrator set the parameters inappropriately, then it will be easy for any hackers or malicious users to gain and produce exploitable security holes. In general, Host-based Intrusion Detection systems (HIDs), Host-based Vulnerability Assessment (VA), antivirus and access control (authentication) provide security at this level.

Application level security: is fourth level security in which it receives great attention in many organizations due to emergence of various attacks. Poorly protected applications can be considered as providing easy access to confidential data and records. The reason behind for the occurrence of different attacks at this level is that most programmers don't code with security in mind. Mostly, administrators may become aware of security shortcomings in the application software, yet they may be powerless to correct them.

Different applications are being placed on the web for access by customers, partners or even remote employees with increasing frequency. Those applications are possible target for hackers to inject malicious content towards an organization. Therefore, it is especially important to impose a comprehensive security strategy for each of application. Generally, application shield, access control (authentication) and input validation are some which provide security at this level.

Data level security: is the final layer which implies a combination of policy and encryption of an organization information asset. Encrypting data where it resides and as it travels across your network is a recommended best practice and last option because, if all other security measures fail, a strong encryption scheme protects those proprietary data.

Data security is highly dependent on organizations security policies that govern who have access to data, what activities authorized users can do with it, and who has ultimate responsibility for its integrity and its safekeeping. Determining the owner and custodian of the data lets to identify the appropriate access policies and security measures that should be applied. Generally, encryption, access control (authentication) provides security at this level.

2.4 Log File Analysis

Log files are rich sources of information exhibiting the actions performed during the usage of resource in the data center [39]. They are detail recorded history about the health of the infrastructure. The term “log analysis” (log processing) refers to an art and science seeking to make sense out of device generated records (also called logs) [22, 28, 44, 45, 46]. Log file is generally in American Standard Code for Information Interchange (ASCII) a file format having a .log extension. It is generated by devices with different functioning logs and alert services [49]. The process of creating such records is called data logging.

Routine log analysis is beneficial for identifying security incidents, policy violations, fraudulent activities, and operational problems. Log files can be stored in either human readable format or machine language [22]. They are emitted by network devices, operating systems, applications and others in which they are intelligent or programmable device [45, 46]. A stream of messages in time-sequence often comprises a log. They are also directed to files and stored on disk, or directed as a network stream to a log collector module.

Log messages must usually be interpreted with respect to the internal state of its source and announce security-relevant or operations-relevant events (e.g., a user login, a systems error or others). Previously, logs were often created by software developers to aid in the debugging of the operation of an application [47]. The syntax and semantics of data within log messages are usually application or vendor-specific. Terminology may also vary; for example, the authentication of a user to an application may be described as a login, a logon, or authentication event.

Hence, log analysis interprets messages within the context of an applications, vendors, system or configuration in order to make useful comparisons to messages from different log sources. Log message format or content may not always be fully documented. A task of the log analyst is to induce the system to emit full range of messages in order to understand the complete domain from which the messages can be interpreted.

Log management is essential to ensure securities of log records are stored in sufficient detail for appropriate period of time [56]. It is the process of generating, transmitting, storing, analyzing,

and disposing log data. The fundamental problem of log management is effectively balancing limited quantity of log management resources with continuous supply of log data. Log generation and storage can be complicated by several factors, such as high number of log sources, inconsistent log content, formats, and timestamps among sources; and increasingly large volumes of log data [28, 29, 43, 56]. Log management also involves protecting the confidentiality, integrity, and availability of logs. Another problem with log management is ensuring security of systems and network regularly to perform effective analysis of log data. Figure 2.4 shows a sample log file taken from [51].

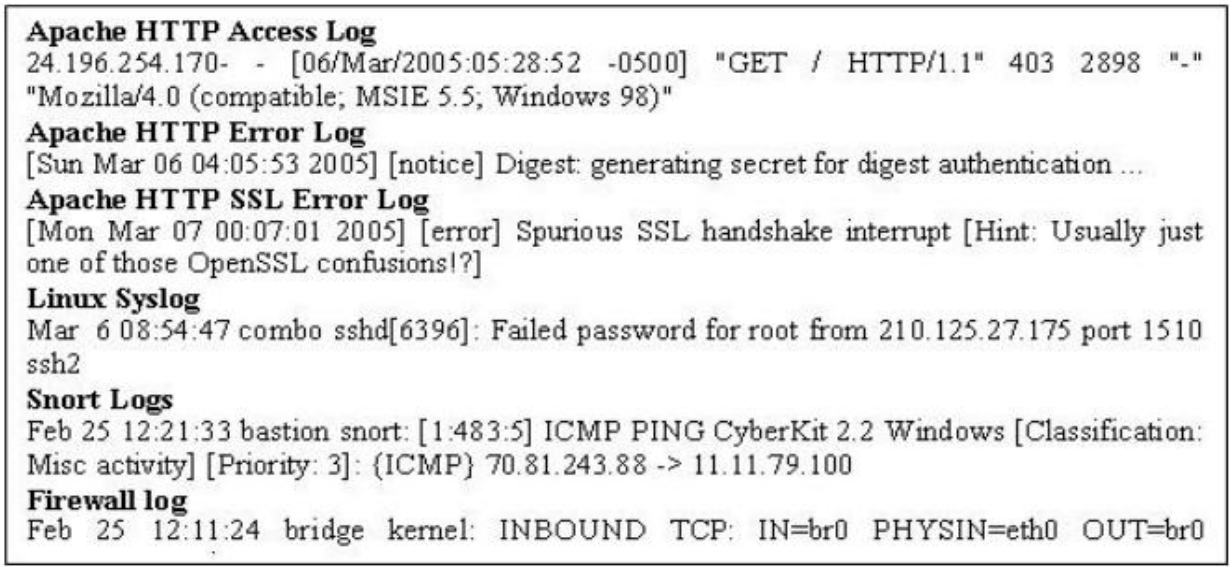


Figure 2.4 Sample Log File

A log analyst maps varying terminology from different log sources into a uniform, normalized terminology so that reports and statistics can be derived from a heterogeneous environment [18]. For example, log messages from Windows, UNIX, network firewalls, and databases may be aggregated into a "normalized" report for the auditor.

Different systems produce different message priorities with a different vocabulary, such as "error", "warning" Vs. "err", "warn". Log file analysis has proven to be a good defense mechanism as log files provide an accessible record of network activities in the form of device generated messages [28, 43].

There are a number of points that should be kept in mind if long term analysis of log file is the goal [48]. The following are important points in which administrators must consider for the process of log analysis.

- **Configure everything with long-term analysis in mind:** Log files are commonly used as a short-term troubleshooting tool without regard to what the long-term implications might be. Set device configuration standards for concepts like naming, and timing (NTP) and apply them consistently.
- **More data is better:** Set logging facility to the highest level that doesn't interfere with the operation of the device. Filtering data is easier than working around missing or incomplete data.
- **Need context:** Knowing about external events is a valuable asset. Keep track of anything that might have a bearing on the logs either by inserting time stamped messages in the log file or by maintaining a separate database.
- **Normalize the data:** Even with Cisco logs, there are variations between devices. Adding different devices format makes the problem worse. Preprocess log files to conform a standard template.
- **Examine log files:** Mine the data regularly using any techniques or tricks that might isolate unusual and interesting patterns.

2.4.1 Log File Collection

Log files are excellent sources to determine the health status of a network and used to capture the events occurred within an organization's system and networks. Log file collector is responsible to gather various kinds of log files with different format from those devices allocated in the network.

In the layer based data center network, they have the capability to produce logs of various types that can be in a category of perimeter, network, host, application, and data level. After that all log files will be stored in a central pool which have many importances and provide experts to have multidimensional views of organizational users' activities, devices, system and applications state through log analysis.

There are various log data acquisition strategies of which one is to create a central server in the network implementing a protocol called syslog which is widely used in agent having automatic program to access and normative data safety system such as security logs, alarm, information, and etc. Those various devices have different log formats. The data collection activity can be conducted in either with agent-based or agent-less based approach.

An agent based log collection approach obtains log files from those generating devices and put them for further processing nearly in real time and put into the destination log file repository. This helps us to ensure a minimal chance of logs being modified or deleted by malicious users and to obtain reliable evidence of successful attack. On the other hand, in agent-less environment it usually leads to non trivial, incur a higher administrative overhead to implement and additional management duties. In general, data collection phase deals with agents designed to simplify the means and technique of retrieving required log data from the intended networking devices.

2.4.2 Approach in Security Information and Event Management

Security Information and Event Management (SIEM) system is industry specific term used in computer security referring to the collection of data typically log files from various sources into central repository for analysis. Event logs can be generated by various networking devices, operating systems and application servers. It will give us raw input of all activity happening in the data center infrastructure of any organization. The raw data act like input to SIEM system which provide us security alerts, reports as an output. The processing of all raw data is achieved using data mining technique [23, 24, 41].

The application of data mining is becoming increasingly common both in private and public sectors [23, 24]. If we extend capabilities of data mining to all events log generated by various devices in the network, then the enterprise security will be enhanced drastically. The major problems in today's enterprise security is the amount of logs generated by different systems and organizations often put too much load in their firewall, operating systems or antivirus software.

Security mechanisms could either perform as well tuned pieces that play wonderfully by themselves but bring a trouble when they are all brought into the same room. Each individual

security component normally could be doing its task by protecting certain section of the data center network, but the security function may be lost when it is time to interrelate or communicate with another security component.

In general, SIEM systems help to take advantage of a security architectural view of an enterprise, data flow in and out of the data center, how data is accessed, modified and monitored at different points, and how all the security solutions relate to each other in different situations.

2.4.3 Alert Correlation Approaches

Alert correlation can be defined as an integration of multiple component processes to analyze alerts providing a high level vision of the security state of the supervised network [55, 58]. Its main use is to identify attack strategies and their objectives. IDS field, has been investigated to try to predict the next step of an attack by monitoring the intrusion behavior. The alert correlation provides means to group alerts logically connected to build attack scenarios. A widely accepted classification [28] of the alert correlation techniques is:

- **Similarity based approaches:** group alerts based on the similarity between alert attributes such as IP addresses and port numbers. Essentially, clustering techniques helps in reducing alerts but they cannot discover causal relationships between alerts. Works in this area are Spice, probabilistic alert correlation and alert aggregation.
- **Pre-defined attack scenario based approaches:** The process is done by correlating alerts based on well-known attack sequences. These sequences can be defined in a variety of languages like STATL and LAMBDA or can be learned through data mining techniques. The alert correlation in this case is causal but none of the works so far can identify novel attack scenarios.
- **Pre/post-condition based approaches:** Based on the principle that the goal of an attack is achieved in multiple steps; the alerts can be correlated to represent these steps through conditions. The post-condition of one attack is linked to the pre-condition to form and detect novel attack scenarios. However, specifying pre-conditions and post-conditions for each attack is time-consuming and error-prone. JIGSAW and MI-RADOR are representative works of this approach.

- **Multiple information based approaches:** Due to multiple information sources, a global vision of the system is provided. The mission-impact based approach ranks the alerts based on the overall impact to the mission of the networks. M2D2 proposes a formal model to describe the concepts and relations about various security systems.

2.5 Summary

In this Chapter we have tried to review concepts concerning the characteristics of data center, data center security and log file analysis in detail in order to enhance organizations data center security. The state of art of data center security in log analysis is still evolving. Defining architecture of a data center always encompasses the notion of security. It is a means to guarantee the service delivery in terms of ultimately keeping the availability, integrity and confidentiality of information. Data center security is a continual process that requires proper management for the infrastructures existence.

Log file analysis plays a vital role for data center security since it provides surplus amount of information with respect to health of data center. It focuses on identifying abnormal events from normal through different approaches which is a post defense approach. The type of features identified depends on the approach used. In general, log file analysis is a core concept for determination of attacks at the host or network level by scrutinizing the log events recorded by the operating systems, application and devices.

Chapter Three - Related Works

In this Chapter, previously conducted researches in areas related to the concept of log analysis along with layered architecture to build an enhanced data center security is presented and deeper discussion on various proposed log analysis techniques is made. Also, specific works in relation to our proposed log analyzer having different approaches will be elaborated in the next sections which are related and relevant in terms of their objectives.

3.1 Related Works

Now a day's security gets more attention in many organizations than ever before. This is due to the growth of Internet and dynamic nature of emerging attacks towards an organizations data center [27]. When organizations ensure security in their business strategy, then the confidentiality, integrity and availability of data will be assured in the data center. Security requirement has a direct relationship with the growth of a data center. It plays fundamental role in contributing towards the development of organizations.

Therefore, network devices generated information (log file) is considered as a means to identify, detect, analyze and take a remedy action accordingly. This enables administrators to easily handle the monitoring activity of the entire data center infrastructure with minimal data loss, time, effort and other expenses.

Recently, the expansion of Internet leads many organizations to be victim of various attacks and create channel for easy dissemination across organizations' data centers freely. Security becomes great point of interest in which it is accomplished through the process of log file analysis. In such circumstances several research dimensions are conducted towards the data center for guaranteeing security at the required level. The works done so far can be taken as inputs and used to bring a newly proposed solution aimed to enhance data center security.

Log files are rich source of information and have been analyzed in the past for a variety of purposes and reasons, such as system maintenance, software testing and validation, forensic

analysis and for anomaly detection. The following section will briefly discuss works which has been done related to our thesis work. Here we have categorized the works which are done so far based on purpose for the usage of log files as discussed below.

3.2.1 Log Analysis as a Security Aid

Leite Jorge [50] collected log files and use an information retrieval open source tools to index log files' fields and search for patterns of suspected behaviors, which may indicate intrusions. The aim of the work was to use the tool for indexing and searching for attack like patterns on log files. The application indexes every occurrence of specific strings (e.g. denial of access, wrong credential, and so on). After that, the system tries to find events within similar occurrences compared with the data searched for. Fuzzy searches have been used to detect same attack like patterns such as brute force attack.

However, the application is very limited to analyze few number of log files type with known log formats and patterns. Hence, multiple types of log need to be considered and also correlating such logs from myriad of recourses is necessary. And also, the system did not use a well prepared log data by incorporating a preprocessing task. An attack knowledge base which is dynamically updated must be constructed in order to easily handle the analysis process.

Herrerias J. and Gomez R. [28] proposed an automated forensic diagnosis system to reconstruct the attacks actions after a security incident has been occurred. Their system analyzes a set of log files created by the different applications running in the network. The system is composed of four modules: event collection, event pre-processing, event correlation, and attack graph generator; all of them working on victim system log files to recreate in an automated fashion and come up with the attacker actions represented by attack scenario. First, event collection module gathers the log files in their original format then, the events pre-processing module adjusted timestamp of log files, normalizes the attributes of the log files and saves them in a repository (event container). Later, the event correlation phase proceed by: first, atomic attack definitions from an attack knowledge base are used to find specific attack actions, and second, the attack actions found are then correlated to build attack scenario describing complex multi-steps attacks. Finally, actions are represented in graphs to facilitate interpretation to end-user.

However, the proposed system did not consider the pattern of attack knowledge base will be outdated. And also, in preprocessing stage the log record may be incomplete for unknown reason or has different log format but the system did not put any metrics to clean a data. This can reduce the detection accuracy of the system. In attack definition port is left as an attribute which is important parameter like IP address to have a better insight about intruders activity from log analysis.

3.2.2 Log Analysis using Data Mining

Asif-Iqbal H., Nur Izura *et al.* [51] propose a system that parse/isolate logs from various sources and then cluster the logs using data mining tool (WEKA). The framework first collects unlabelled heterogeneous logs, then parse each raw log individually and isolate log entries when necessary. Secondly, process of clustering of log entries before filtering. Thirdly, again parse the clustered logs to make it visible for filtering. Later on, the process of filtering proceeds to filter the clustered events. Finally, the system combines the filtered events attribute values which are exactly alike.

However, the proposed work lacks to create common log format through log normalization in a preprocessing module for identification of log in its proprietary log format. In addition, it is better to construct an updated Attack Knowledge Base and compare each filtered event against the knowledge base. Additionally, in their system finding association (correlation) among log records is not considered for attack identification.

Ghani Abdul [52] develop Unsupervised Heterogeneous Anomaly Detection system (UHAD) which scrutinizes heterogeneous logs, without using a trained model on traffic behavior or knowledge of anomalies, and uses a two step strategy in clustering normal and abnormal events. They introduce new algorithm for filtering, by which the filtering threshold is calculated based on the volume of log events and the number of log events clusters. First, the component log preprocessing was used to extract the data from the logs. Secondly, the event clustering component separates abnormal events from the normal ones using various logs and also finds possible number of clusters (K) to group the events using expectation maximization algorithm. Thirdly, filtering clustered events component remove the normal events whilst retaining the

abnormal events for further processing. Later on, aggregation of filtered in events component combines the redundant events thereby reducing the events in the filtered log. Then, transferring events component extract the features from various aggregated logs as stated in Generic Format (GF) to store in Generic Format Log (GFL). Finally, the system detects anomalous events by analyzing features such as IP address analysis and port number analysis.

However, the system lacks preparing and using heterogeneous log files in pre processing and no training (learning) and test (prediction) dataset to produce a better detection results. It also requires building attack knowledge base to extract new atomic attack definitions for increased attack detection accuracy. In their system the concept of correlation among log events was not taken into consideration to come up with more attack definitions.

Vaarandi Risto [57] discusses a data mining tools Simple Log Clustering Tool (SLCT) and LogHound that were designed for assisting system management to extract knowledge from event logs. The automation of event log analysis is important research concept in network and system management. In order to tackle such problem proposed a data mining technique to obtain knowledge about events. SLCT is basically employs clustering algorithm for analyzing textual log events where each log record represents certain event. On the other hand, LogHound employs a frequent item set mining algorithm for discovering frequent patterns from event logs.

However, the research did not consider the notion of attack detection methods as part of the work when combing SLCT and LogHound, in order to build an event log attack detection system. Another drawback is the proposed system did not include more processing techniques to produce efficient log analysis system.

3.2.3 Log Analysis in Real Time

Kiatwonghong N., and Maneewongvatana S. [44] proposed real time log analyzer system that begins with collecting log data from devices in the network into central server by removing garbage data and define relationship between them to bring them into one common table. Then, it converts them into one common format. Customized learning algorithms such as association rule, tf-idf, kmeans clustering, and decision tree were used to analyze and interpret data to get important information from log data. Finally, the system converts again into the graphical

formats for easy understanding. The system uses adaptive learning algorithm to process the data stream and data model that changes along with time and to flush out the old model when the model is too old. In general, the system target was to detect the abnormal activities using combination of signature-based and learning algorithms based techniques. Their proposed architecture of log analyzer system is shown below in Figure 3.1.

Even though, the work come up with low false positive rate but still it lacks further refinement to process the log files collected with more algorithms to gain a better log analyzer. During conversion of log data to common format they ignore how to extract important features from log file which have direct relationship with detection accuracy of the log analyzer. The application of tf-idf algorithm is less important for detecting attacks using log files since it results more irrelevant detection results.

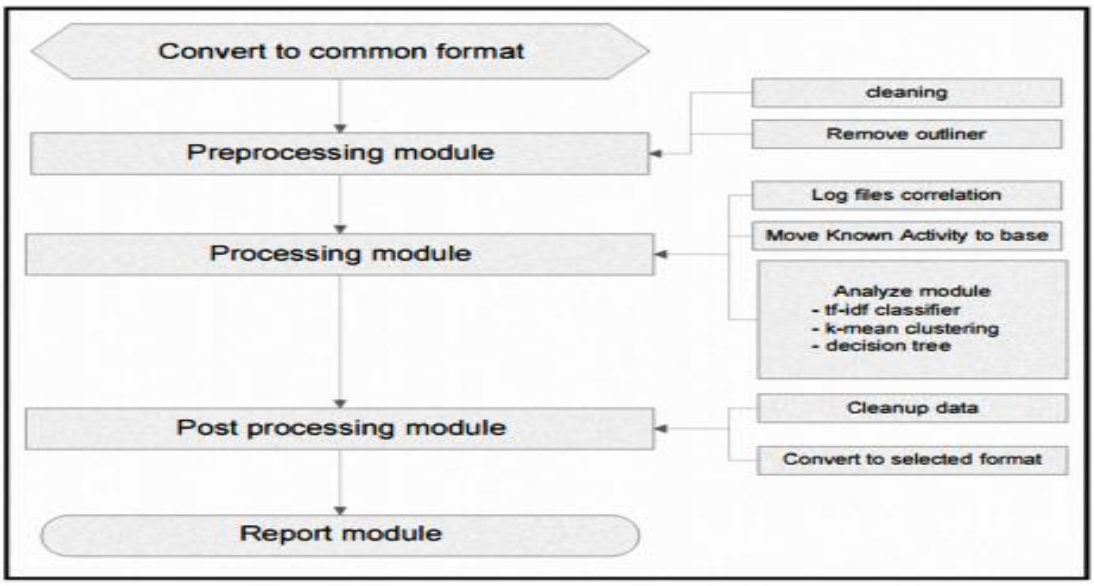


Figure 3.1 Real-time Log Analyzer System Architecture

3.2.4 Log Analysis for Log Management

Shengyan Shi, Shen Xiaoliu *et al.* [53] developed a method for using the multi-agent technology to collect and analyze log data generated by network devices and security devices, and then generating a fixed format data structure and build log collection and analysis systems by

incorporating preprocessing operation. The system analyzes the status of the network and information security, and created the centralized storage to provide services for the later research. At first, the log collection agent collects the data of all network devices; security devices, applications systems and host system then, normalize these log data into pre-defined log format and store in the central storage. And then, filter it according to the rules defined in the rule module. Users have flexibly to adjust the parameters of filter and filter out the data in which the user may be in interested from the many log information according to the actual needs.

However, the proposed system is pattern based in which currently emerging attacks do not go along with the defined rule which create security hole in the organization. The absence of structured and up-to-date attack knowledgebase and absence of log correlation bring bottleneck for analysis of log files. The preprocessing module task is not well known for making the system more robust. In addition, in their system the type of activities carried out in log analysis stage is not well stated.

Söderström, Esmiralda Moradian, *et al.* [54] proposed an approach for receiving, storing and administrating log events. They presented a secure audit log management system focusing on security, flexibility, performance, and portability. Furthermore, they come up with a design solution that allows organizations in distributed environments to send audit log transactions from different local networks to one centralized server in a secure way. They proposed system which has the capability to analyze logs from different log sources such as, firewalls, IDS, servers, and clients. The proposed system consists of one centralized server located on a secured location connected to the inner parts of the network of the supported system. To collect those log events syslog and Simple Network Management Protocol (SNMP) protocols are used. Then, the agents read local logs and transfer the information with the syslog or/and SNMP protocol to the log server. The security audit logs can also be transmitted over the network to the system using standard User Datagram Protocol (UDP), syslog protocol or Transmission Control Protocol (TCP). In such a way the system collects log information from all types of clients, servers, firewalls and network equipments. Moreover, the log server is able to detect activation and deactivation of nodes and network equipment on the supervised network, i.e. the network where logs are collected from.

However, the system lacks works related to the preparation and implementation of standardized log formats for heterogeneous log data, as well as aggregating of logs in standardized way. In addition, the issue concerned with the approach or technique used for detecting anomalous events from the log file was not considered.

Herrerias Jorge and Roberto Gomez [55] developed a model composed by a set of agents to collect, filter, normalize, and to correlate events coming from diverse devices. The model provides a capability for analyst in the evidence search process of a forensic investigation. Agents collect log files and send them to an event container. Afterwards, events are filtered by reducing the number of events which are not related with the attack and then normalize to standardize the information of logs. Once all logs are in the same place and format, the correlation engine processes the events and generates a diagnosis of how the system was penetrated. Correlation analysis assigns relationship between multiple events related directly or indirectly with the system violations. In other words, correlating events help at identifying the attacker actions by analyzing events of diverse applications all together.

However, the proposed system limited to correlation (association of events) and exclude log mining algorithms for multispectral identification of attack to obtain better attack detection results. In addition, the system did not use fine tuned filtered event from event container which means if the event contains incomplete information then the system consider as an attack which is not true.

Sayed Omid Azarkasb and Saeed Shiry Ghidary [58] brought a new method for correlating intrusion logs with the main processes of intrusion detection system. It is based on a centralized log correlation system which is composed of six components: data provider, preprocessor, analyzer, manager and controller, responder, and evaluator. In the proposed system, data provider collects data from network logs audited data file (off-line mode) or live network logs (on-line mode) and sends text data to the processor component. Then, the preprocessor converts text data into numeric one and if necessary converts numeric data into binary or normalized form, and sends them to a Self Organizing Map (SOM) neural net based analyzer. In preprocessor, after extracting features from each record, each feature is converted from text or symbolic form into numerical form. In the next step preprocessing convert data into binary, or

normalized and scaled form. For normalizing feature values, a statistical analysis is performed on the values of each feature, based on the existing data from dataset and then acceptable maximum value for each feature is determined. The analyzer uses data either for training and testing its SOM neural net or for analyzing and detecting intrusions/attacks. An IDS evaluator component provides a facility for reporting true detection rate, true type detection rate, false positive detection rate, false negative detection rate, and other criteria such as detection rate of three attacks categories, to evaluate their log correlation in the intrusion detection system.

However, the proposed approach that results less detection accuracy comparing to data mining learning schema and limits correlation process to be less effective to identify more intrusions/attacks in the IDS log. The proposed system uses approach for log data collection in online or offline mode in similar way but parameters must be set for proper identification of the log data mode. In the preprocessing component data cleaning is not included which helps to obtain more features.

Kant K., Meixing Le., *et al.* [12] a prototype system was developed and implemented based on relational algebra to build the chain of evidence. It is used to process the real generated data from logs and classify suspicious users based on decision tree. The proposed work describes the nature of the event information and the extent to which it is correlated such event information despite its heterogeneous nature and origins. First, the system begins by extracting log files of the web server and firewall and stored in the central location. In this stage, the data are transformed in a suitable format for conducting effective analysis. Secondly, the chain of evidence analyzer takes the firewall log and web log from the centralized log. It applies the rule based correlation by URLs and Time techniques and creates the training data set. Later on, a decision tree is constructed from the resultant training data set by applying decision tree algorithm which helps to take a proper a decision in suspicious users.

However, URL and time is not enough for building correlation system and the system did not have organized way to analyze log files. This means that it simply takes a garbage collected log records as it is without preprocessing and no common log format which leads a system to be less efficient to identify a malicious users. In addition, the system is limited to use decision tree algorithms in which applying more log mining algorithms gives better detection accuracy.

Figure 3.2 shows the proposed log analyzer framework.

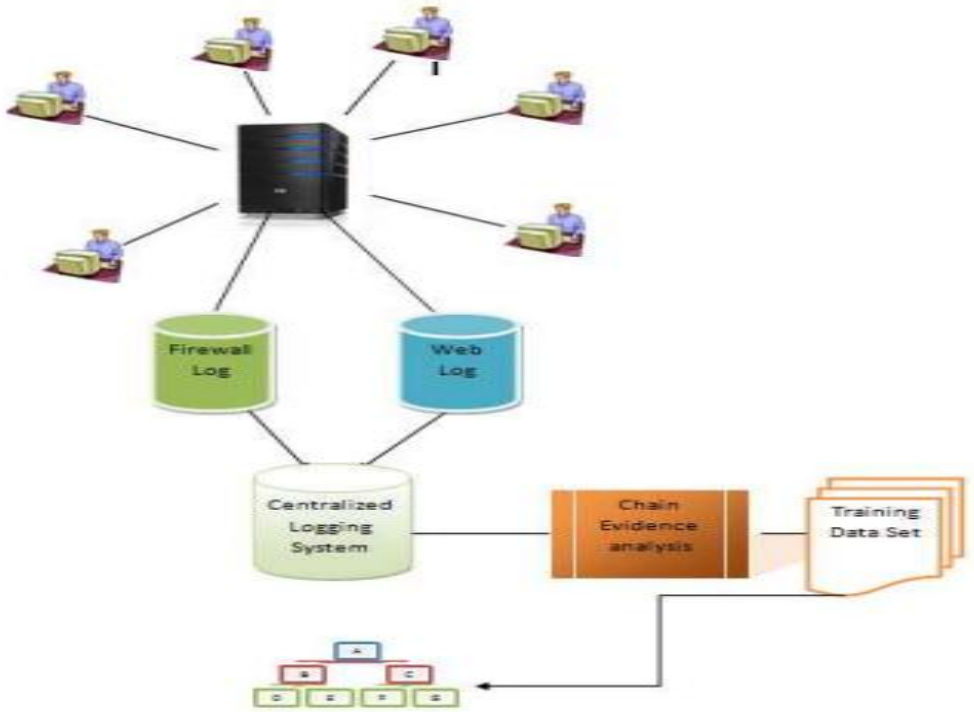


Figure 3.2 Log Analyzer Framework

Kowalski K. and Beheshti M. [59] proposed a system used to analyze intersections of log files that come from different applications and firewalls installed on one computer, and intersections resulting from log files coming from different computers. And also, it is concerned with the issues involving large scale log processing which helps to analyze log records. They have used firewalls' log files coming from web server and from regular desktop computer (in both cases coming from the same period of time), and web server's access-log file from the same time period. During the initial preprocessing stage they have removed from all logs entries that were related to intranet, and they have left only those entries that came from outside of their LAN.

However, in the proposed system log preprocessing stage entries are selected based on the source they originated (i.e. can be from intranet or outside LAN) which has no importance to identify attacks in the network. The system did not put any technique or systematic method for detecting attacks from log records.

3.2.5 Build Log Management Architecture

Madani Afsaneh, Saed Rezayi, *et al.* [56] aimed to suggest log management architecture with more common functions that are used by vendors. They proposed log management architecture having collection server which is the first module for collecting received logs from various log generator devices such as firewalls, NIDS, operating systems, application systems, etc. Then, log generators send logs by transmitting protocols like syslog, IDMEF, CEE, CEF and SNMP. Thus, collection server must be able to understand all log formats. After studying various SIEM vendor architectures on log management the most important functionalities are considered as follows: Normalization, filtering, reduction, rotation, time synchronization, aggregation and integrity check. Finally, storage server keeps logs for forensic, auditing and off line analysis. In addition, they consider log security in their architecture.

However, the proposed architecture functionality is not evaluated and tested. And also the architecture did not consider log processing or analysis component as a core in log analysis.

Nen-Fu Huang, Chia-Nan Kao, *et al.* [38] proposed a defense in depth network security architecture and applies the data mining technologies to analyze the alerts collected from distributed intrusion detection and prevention systems (IDS/IPS). The key component of the Global Policy Server (GPS) is the security information management (SIM) which consists of an online detecting phase and an offline training phase. The system consists of four main components in the online detecting phase: First, the online data miner, which classifies the records in active database to detect attacks. Then, the rules tuner which runs the machine learning algorithm tunes the parameters of rules accordingly. Later, the GLS, which receives logs from LPSs stores them into an active database. Finally, policy dispatcher waits for the commands from the online miner.

However, the experimental results demonstrate the proposed work is highly effective only for detecting the DDOS attacks which is not for other attack. It also did not show if we take a shorter time interval between the events it is difficult to suggest about occurrence of the false alarm rate. Also, the model only uses classifier as mining technique.

3.2 Summary

Even though, the reviewed papers propose their log file analysis system for various purposes and try to address different issues, there are shortcomings in those works that we propose to address in our work. Data center suffers from security risk that arises from absence of proper security techniques and evaluation mechanisms. For such case log file analysis plays significant role in addressing the issue. Most research works done so far considers few type and format of log files and used specific approach for analysis which produces less important information about attacks. Thus, the use of heterogeneous log files, and considering variety of log file format makes their analysis to be impractical. The consideration of various log files improves the capability of detecting abnormal events which is part of our thesis work. In general, the drawbacks of each work discussed in this Chapter are basis for the proposed solution and provide a means for identifying security holes.

Chapter Four - Design of Layer Based Log File Analyzer

Healthy delivery of services by data centers requires sound security mechanisms. This Chapter discusses the proposed data center security log analyzer using layered approach data center security. It also shows how the model of log analyzer can be a solution to track attacks on the data center network as well as enable to identify security holes in the infrastructure. In general, in this Chapter, the proposed model with its components is discussed.

4.1 Introduction

Security nowadays is one of the major challenges and issues in which organizations assure confidential information in their data center to be free from any corruption. It is not a capability of finding sudden incident in a single moment rather it needs continuous evaluation for healthy state of the network. Hence, in our work we use log files as input which are considered as a means used to capture the events happened and contain recorded history of the network.

Logs are collection of entries generated by many sources such as servers, operating systems, workstations, networking equipments and other security software, such as antivirus software, firewalls, intrusion detection and prevention systems and many other applications. Each entry contains information related to specific events that have taken place within a network. Monitoring log of events in the network is important and thus plays a vital role in order to keep the entire infrastructure safe, so that, sudden events don't harm any resources of the network system. Thus, the analysis of network logs is becoming a wide security research area.

The exposition of organizational network resources leads most attackers to have a plethora of means to perform any kind of activity they intended to do. In addition, organizations are reluctant to expose their logs due to risk of attackers stealing sensitive information out of their respective critical network device logs. Hence, systematic methods of handling of resources through keeping log records in a secured manner is very important and utilize required information from each log records through log analysis gives potential clues about the security state of data center for administrators. The following section briefly discusses our proposed architecture.

4.2 Components of the Proposed Architecture

In fact, the development of system is determined based on the composition of many subcomponents (parts) of the entire system. Hence, the integration and interoperability of those components will produce the expected system so that its objective will be met.

Our model of log file analyzer with layer based data center security consists of the following major components, including Log Files Repository (LFR), Log File Pre-processor (LFPP), Log Repository (LR), Attack knowledge Base (ANB), Central Engine (CE), Action Center or Remediation Area, and Audit Reporter as shown in Figure 4.1. Log file collector is simply concerned with gathering heterogeneous log files generated by those leveled devices in the data center network and put them in central repository (LFR) for preprocessing which is second component of the system.

The aim of the log file pre-processor is to adjust or prepare various input log files considering different metrics and feed to the processing unit called central engine. Log parsing, log cleaning, log normalization, and log aggregation are the major activities conducted in this component. Once this part is completed, then the preprocessed log files are stored in LR component and ready for later usage by central engine.

Log repository is a module that works as storage for the raw of log data which are pre-processed and received from log file preprocessing component. Attack knowledge base is part of the system which consists of atomic attack definition received from the central engine and provides required atomic attack to the central engine.

The central engine component is the heart (core) of the system which is responsible to perform the overall log file processing. To achieve this subcomponents include clustering and correlation or association module which comprise their own components. On the other hand, an action center or a remediation section is a component that deals with providing reasonable response towards emerging incident to the administrator through the user interface. It incorporates alert production, notification through SMS, generating report, and visualizations. Finally, the audit reporter module is a repository for the generated information from action center component for long term usage. Therefore, our research is mainly concerned with enhancing data center

security trends of organizations through building log file analyzer (i.e. intended to identify and inform the state of the data center within certain period of time). The following section will briefly elaborate the concept related to our proposed architectural design for log files analysis using layered approach of data center security.

4.3 Architecture of the Proposed Log file Analyzer

Log analysis is the process of generating, transmitting, storing, analyzing, and disposing of log data. It is beneficial for identifying security incidents, policy violations, fraudulent activity, and operational problems. The overall process of log analysis for delivering services is described based on the built in combination of each component in system. The next subsection discusses details on each component of the architecture which is outlined in Figure 4.1.

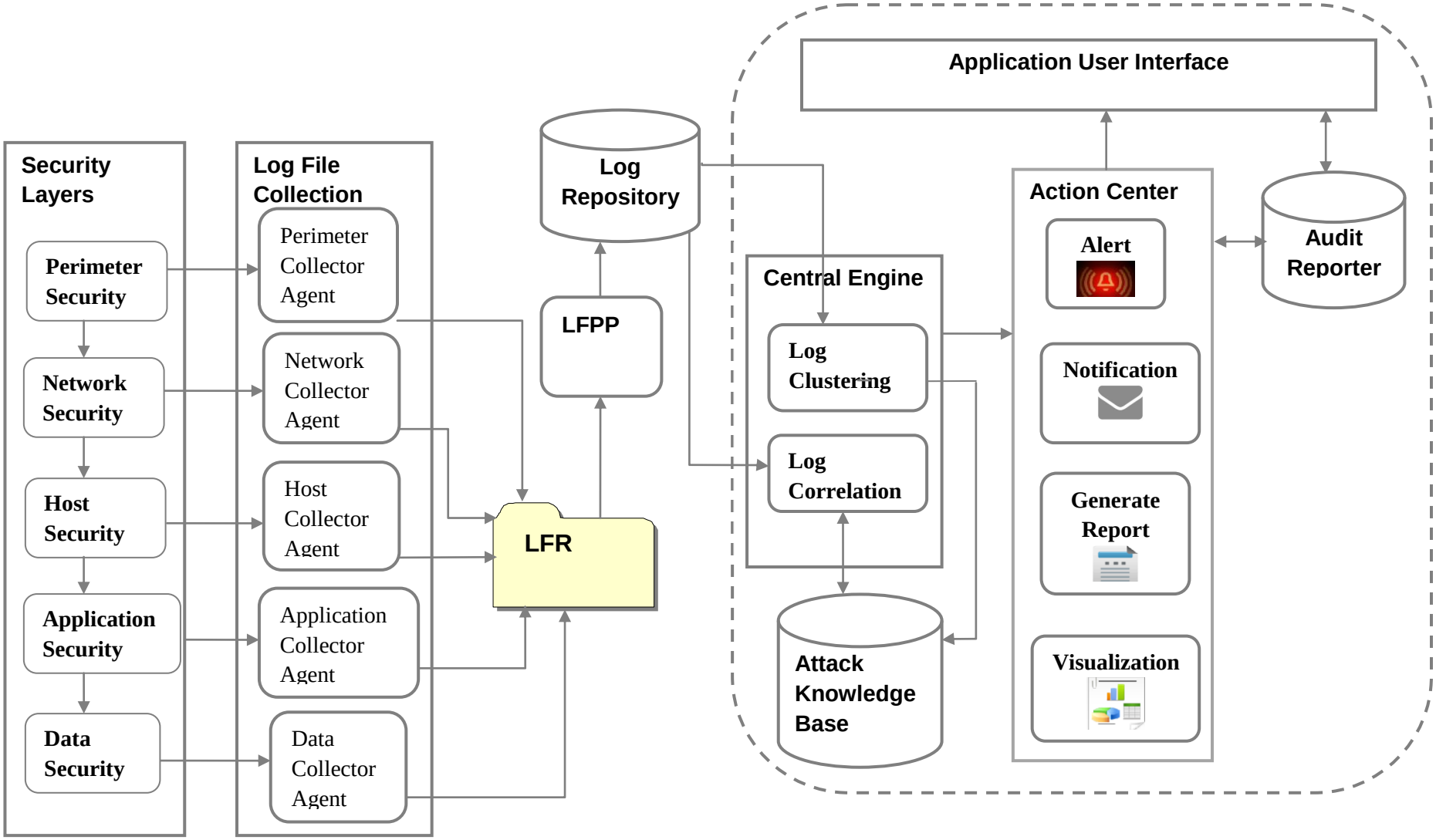


Figure 4.1 Architecture of the Proposed Log File Analyzer using Layered Approach

4.4 Log File Repository

Log File Repository (LFR) is a central storage for the entire collected log files gathered from heterogeneous devices layered in the data center network of an organization. Our proposed system relies on this component since the collection of log record is important for conducting log analysis task. The repository simply consists of raw log data without modification of its content. Log file preprocessing takes those log records from this component.

4.5 Log File Pre-Processor

Log File Pre-Processing (LFPP) is a planned activity applied on the raw collected log records taken from central store called log file repository before proceeding to any further step. It provides analysts to obtain great advantage in terms of accuracy during identifying incidents from log processing. This task is a ground task for discovering necessary information out of the entire data. It enables to have the most refined log data which can have a direct relationship with the performance of the log analyzer.

Today's data center security products are different in kind and emerge from various manufacturers for the purpose of managing its infrastructural elements security, and thus, the data will be collected from multi source. In addition to the heterogeneity of log files sources, types, their format is distinct depending on the type of device. Thus, all collected log files must be integrated together and preprocessed, which helps to analyze and respond for administrators in an intelligent manner.

The log file preprocessing component is responsible to make log data ready and take it as an input for further processing by applying specific operations such as log parsing, log cleaning, log normalization, and log aggregation. The flow of log file pre-processing module is shown below in Figure 4.2.

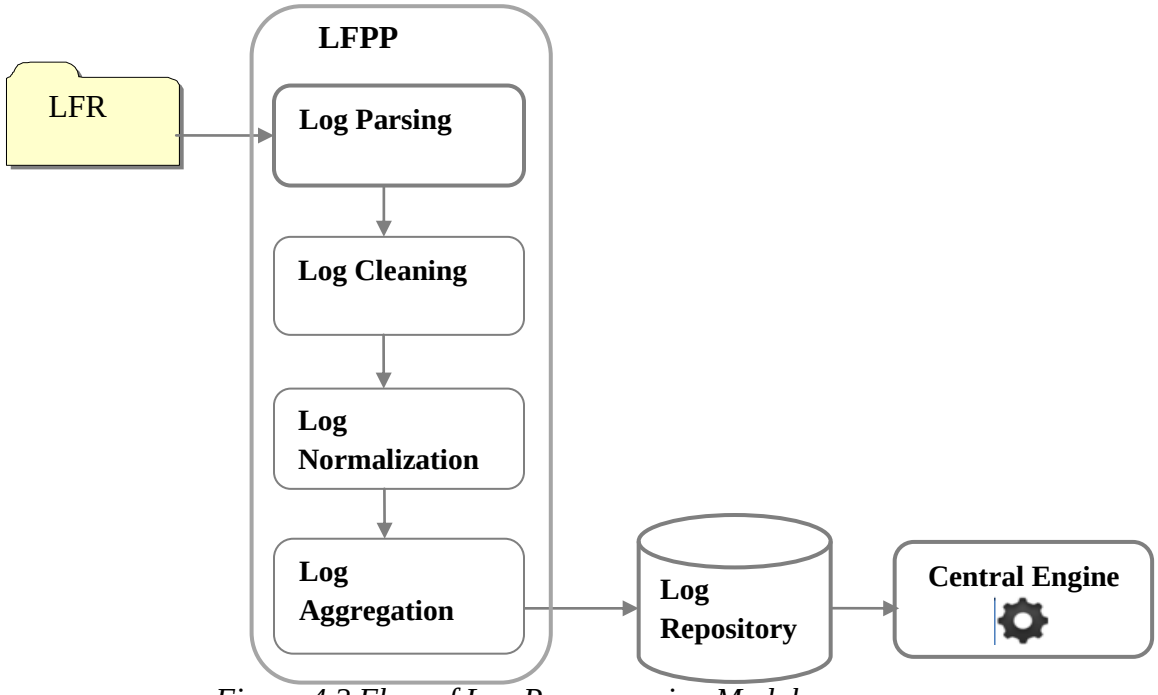


Figure 4.2 Flow of Log Preprocessing Module

The following part will briefly describes each component of log pre-processing module shown in the above figure as follows.

4.5.1 Log Parsing

Log parsing is first component of LFPP which uses a method to separate or segment each of log entries based on predefined rule (such as using regular expression) into known category through defining fields or attributes so that, each parsed value lay under those defined fields. This stage can help to easily extract features and for searching each of log records based on some parameter (argument). Feature extraction is part of log parsing that extracts each feature from the log files. The algorithm shown in Algorithm 4.1 presents how the process of feature extraction is conducted to have feature values that aids log parser.

Most log files have some sort of well defined structure or format. For instance, a web access log specifies URI, user agent, referrer, IP address, and other fields for each log. Hence, for example, when we feed a collected web access log to the parsing component it can simply put the values into those valid fields accordingly. Algorithm 4.1 shows an algorithm of a log parser module used to fragment log records into known fields.

```
//Log Parsing
1. Input: log records//
2. output: parsed log records
3. Begin
   For each log_i in log file
     For each feature_i in log_i
      If log_i.pattern is valid then
             LR.Add(log_i)
          If (feature_i.value == valid)
             LR.Add(feature_i)
          End if
        End if
       Next
   Next
   End
```

Algorithm 4.1:Log Parsing

4.5.2 Log Cleaning

All of collected and parsed log data entries may not be equally important and have similar characteristics. Therefore, this component aimed to identify such a log record with null value, erroneous value, missing value, and resolve inconsistencies as required. It also deals with omitting unnecessary entries and placing data fields of interest for long term purpose. In fact, the production of log data from devices brings different formats, and log data may be incomplete, mutually contradictory, and very large in volume. For example, log files in a large enterprise grow by terabyte per day and filled with non specific or spurious event records.

Although, this collected information is useful and sometimes the only information at hand, at the same time it might be dirty. Hence, proper cleaning method is applied for the collected log data. In general, this component involves the task of removing unwanted features in the log data that might be insignificant for the purpose of log analysis. Algorithm 4.2 shows an algorithm of

log cleaning component used to prepare a cleaned log data.

```
// Log Cleaning
1. Input: parsed log file// from algorithm 4.1
2. Output: cleaned log records
3. Variable: default value
4. Begin
    For each feature_i in log file
      If (feature_i.value not null)
         If (feature_i.value valid)
            LR.Add(value)
         Else
            LR.Add(default value)
         End If
      End If
    Next
   End
```

Algorithm 4.2: Log Cleaning

4.5.3 Log Normalization

In this stage once we obtained a parsed and cleaned log data then, log normalization deals with the process in which each log record is converted to a particular data representation (format) and categorized consistently. In our context, this is a place where heterogeneous event log data from dissimilar systems are converted into a common event exchange language to produce common format. It enables to take log events record from all devices and present them in a commonly known log format for the analysis sub system. Algorithm 4.3 shows an algorithm of a log normalization module used to create similar data representations.

```
//Log Normalization
1. Input: cleaned log records// from algorithm 4.3
2. Output: normalized log data//
3. Begin
     For each feature_i in log file
        If feature_i.format==valid
            LR.add(feature_i)
        Else
             Change (space,comma);
             LR.add(feature_i)
        End If
     Next
   End
```

Algorithm 4.3: Log Normalization

4.5.4 Log Aggregation

It is a task where similar entries are identified and consolidated in terms of the corresponding log entry associated with the number of occurrences of the event. In our system identifying such record can minimize the burden of central engine to process similar log entries. It is the last stage in the LFPP module and provides fine tuned data set to our central engine for log analysis. Algorithm 4.4 shows an algorithm of a log aggregation component used to find similar occurrences of log file entries.

```
//Log Aggregation

1.  Input: Normalized log records// from algorithm 4.4
2.  Output: Aggregated log data
3.  Begin
       For each feature_i in log file
          For each feature_j in log file
            If similarity (feature_i≠feature_j)
                LR.Add(feature_i)
            End IF
          Next
        Next
    End
```

Algorithm 4.4: Log Aggregation

4.6 Log Repository

In this work, Log Repository (LR) is a component which provides centralized repository for preprocessed log data after applying the preprocessing tasks on log files stored in log file repository. The issue of supplying well prepared log data to the central engine is mainly dealt with LR as well as makes the process of log analysis to be easy.

LR always keeps the log records for long term purpose. It is considered as a weapon for many organizations to comply with their security policy, standards and procedures pertaining to security of data center and the latter can be helpful in cases of network forensics. It can also, help to some extent with network diagnostics.

4.7 Central Engine

This component is the backbone of our entire system in which the collected and preprocessed log files will be analyzed. The reason for storing logs can be in case of forensics, auditing, reviewing and understanding network related information that provides further necessities of central engine module. Routine task of log analysis is conducted in this part which is beneficial

for identifying security incidents, policy violations, fraudulent activities, and operational failures of network problems.

It is mainly concerned with processing tasks through applying techniques for example, using selected data mining algorithms. First, the engine takes a well prepared log files with their features that defines each of devices events from LR as they stream into the system. It is able to process them through using techniques such as clustering and correlation to produce usable information. Later, the administrator is able to infer information about the analysis from the user interface.

Once, the central engine performs analysis and comes up with different kinds of events finally sends the detail result to the response system (remediation center) component to react properly against security attacks. Clustering aimed to form various cluster and find attacks based on filtering threshold. On the other hands, correlation process finds relationship among log events to identify attacks and register in Attack Knowledge Base (ANB).

Therefore, in this work log analysis is done based on attack knowledge base which includes attacks signatures; attacks types; ip address, port number and other information. In our work, clustering and correlation modules are taken as part of central engine which will be discussed in the next subsections in which both deal with log analysis. Table 4.1 shows list of data mining algorithms with their corresponding function which can be used for log analysis.

Table 4.1 Various Data Mining Techniques

Techniques Name	**Function**
Association	Pattern is discovered based on a relationship of a particular item on other items in the same transaction
Classification	Classify each item in a set of data into one of pre defined set of classes or groups. Classification method makes use of mathematical techniques such as decision trees, linear programming, neural network and statistics
Clustering	Makes meaningful or useful cluster of objects that have similar characteristic using automatic technique

It has great advantage to analyze large amounts of log data through different approaches to obtain valuable information to assure data center security. Detailed description of each component in the central engine is stated below.

4.7.1 Log Correlation

Log correlation (association) deals with finding relationship between log events to come up with new attack definition. Understanding the occurrence of attack from single log file often leads to false detection accuracy of attacks (malicious events) from the log file. The assumption is all of log records provide relevant background about the detail of an event occurrence. Figure 4.3 shows log correlation process which performs the following operations.

In our proposed system log correlation begins with accepting preprocessed log data stored in LR. Then, search query will be sent to attack knowledge base to check the presence of attack definition. Attack knowledge base processes the query and matches result found to create an attack scenario showing details of attack and the result will be shown in the action center module. If the search result did not match, then the process of registering the attack information as a new entry to the attack knowledge base will be conducted. The flow of processes in log correlation is presented in Algorithm 4.5.

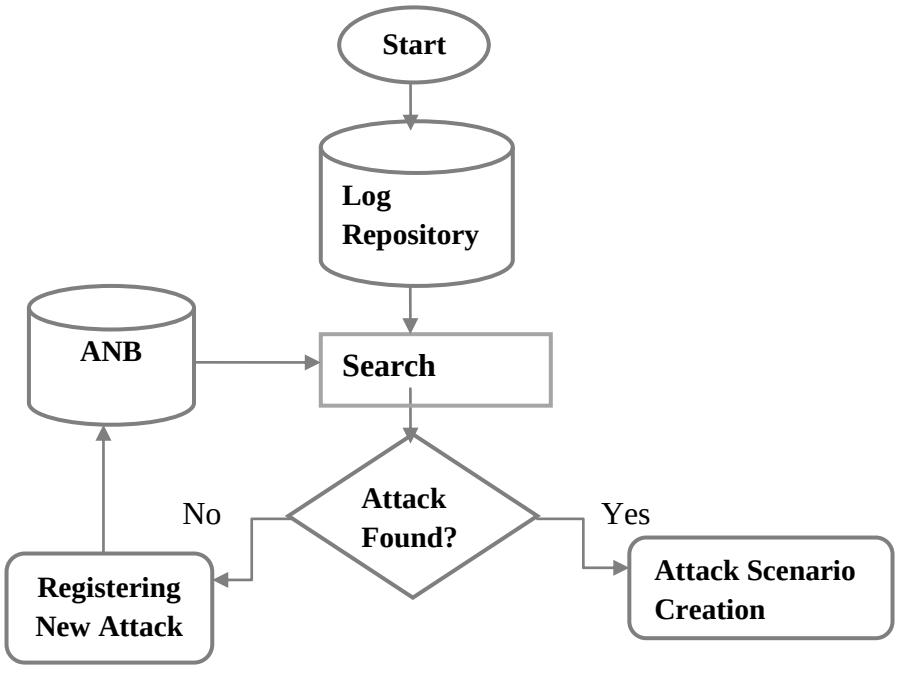


Figure 4.3 Log Correlation Process

```
//Log correlation
1. Input: Log data of LR
        :Attack Knowledege Base(ANB)
2. Output: Attack Scenario
3. Begin
        read attack in ANB
      For each log_i in LR
        If log_i.value equals Attack
             create attack scenario
         Else
             display(Attack search found)
             register(Attack)//newly found attack
         End If
       Next
   End
```

Algorithm 4.5: Log Correlation

4.7.2 Log Clustering

Clustering is a technique employed to cluster and filter log entries based on their features to facilitate the process of log analysis to identify attacks or abnormal events. In our work, using clustering algorithms we construct an equivalence clustering analysis. Clustering analysis method reduces the number of similar log events and to emphasize the similarity of log events. Figure 4.4 illustrates log clustering module for clustering log events.

In our proposed system, LR module provides well prepared log data to log clustering module to obtain an interesting pattern from the log analysis. It begins with extracting important features out of the entire categorized log records into distinctive category and isolates log entries if necessary. Next, it groups each of log entries before proceeding to the filtering stage by estimating the ideal number of clusters (K) for each log data set using Expectation Maximization (EM) clustering algorithm.

Then, filter the clustered events using filtering threshold. Finally, combine filtered events in which the attribute values are exactly alike through aggregation. Attack scenario will be developed and sent to either action center or ANB. To achieve the above, Algorithm 4.6 and Algorithm 4.7 describes log clustering and log filtering, respectively which is shown below.

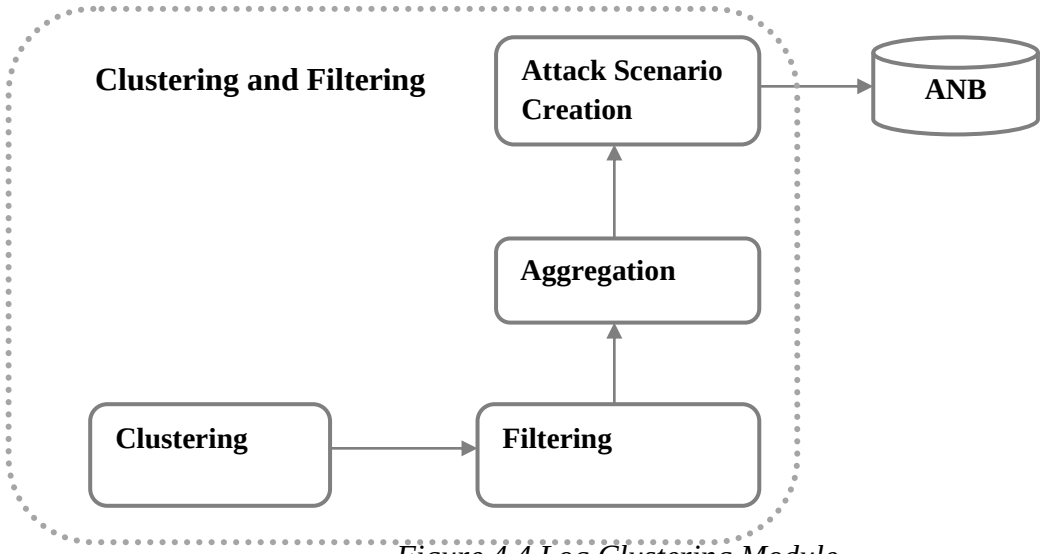


Figure 4.4 Log Clustering Module

```
//Log clustering
1. Input: Log data from LR//Aggregated log from algorithm 5
2. output: Clustered log records
3. Begin
         extract features// see algorithm 4.1
           For each log_i in log file
              If feature.similar(log_i,log_i+1)
                 LR.add(log_i,log_i+1)
               Else
                 group(logi,logi+1)
               End If
            Next
   End
```

Algorithm 4.6:Log Clustering

```
// Log Filtering
   1. Input: clustered events//see algorithm 4.6
   2. output: filtered events
   3. Begin
        For each log_i in cluster
         If log_i.value equals filtering threshold
              LR.Add(log_i)
             cluster(log_i less than or equal to total
             number of cluster events)
         End If
        Next
      End
```

Algorithm 4.7:Log Filtering

4.8 Attack Knowledge Base

In our proposed work Attack Knowledge Base is composed by a set of definitions of atomic attacks collected from the analyzed logs and also it can be extended for using for further log analysis through applying correlation function with events in the log file. Each atomic attack represents a well defined intruder action and the resulting events registered in the log files along with a series of attributes as specified below.

Attack= {

[$type_1$, $signature_1$, $exists_1$, $timestamp_1$, $ipaddress_1$, $logfile_1$, $port_1$]

[$type_2$, $signature_2$, $exists_2$, $timestamp_2$, $ipaddress_2$, $logfile_2$, $port_2$]

.....

[$type_n$, $signature_n$, $exists_n$, $timestamp_n$, $ipaddress_n$, $logfile_n$, $port_n$] } or simply

Attack= $\bigcup_{i=1}^{n}(type_i, signature_i, exists_i, timestamp_i, ipaddress_i, logfile_i, port_i)$

Where each of atomic attack is composed as follows:

type: is a description of the event in order to understand what the event is about.

signature: specifies the event information recorded on the log file and is used to find the event during the correlation process.

exists: is a boolean value or flag used to mark that the event was found and that it is part of the atomic attack. In this part comparison with the registered atomic attack will be made.

timestamp: is the time attribute for occurrence for an event.

ipaddress: is the IP of the intruder and extracted from the event attributes which is important for analysis.

log file: is the source log file where the event is located.

port: is the port number of the intruder in which they accomplish their action extracted from event attribute if exists.

The signature of log records leads to the search process, while the rest of the variables save the event related information. In other words, when the signature is found in the log file, exists flag value is set to be true, and the timestamp, IP address and port attributes if exists are saved in their respective fields.

4.9 Action Center

If the system detects an incident it will react through its action center or remediation area (act as incident management) channel which include alerting and even initializing counter measures. There are various means in which an administrator will be informed. These are: alert, SMS for notification generate report in different format and provide visualization (using bar, pie chart, statistical detail and others).

4.10 Audit Repository

It is the final process of log analysis which provides precise and detailed log information. There are different ways to put event logs. Logs are saved for long duration as it helps to determine forensic analysis. As events stream in to central engine they are written in database in normalized schema. This storage helps us to do forensic analysis on historic data. Stored events are used in different activities such as forensic, monitoring and auditing, requisites in response and other activities. We selected and used MySQL the most popular open source RDBMS.

Chapter Five - Implementation and Discussion

In this Chapter the implementation details of our proposed log file analyzer using layered data center security is presented. To evaluate the capacity of this framework in detecting anomalies, we extracted and combined events from various sources. The proposed system was implemented in detecting various anomalies were evaluated based on the challenge results of SOTM#34[7]. In addition, log files gathered from Addis Ababa University data center which is the case of our work were used this purpose.

5.1 Data collection

In order to provide our prototype we have to find for well organized log files corpus. Thus, to achieve this we used various collected log records such as: SOTM#34 and AAU as an input and evaluate the performance of the system accordingly. The evidence files of SOTM#34 are divided into four subdirectories according to its source: http, iptables, snort, and syslog, as shown in Table 5.1. We collected various kinds of log files which allow conducting an extensive log analysis task.

Public security log sharing site[8] contains various free shareable log samples collected from various systems, security and network devices, applications, etc. The logs are collected from real systems; some contain evidence of compromise and other malicious activity. Wherever possible, the logs are not sanitized, anonymized or modified in any way (just as they came from the logging systems). In the site there are nine bundles of log data. From those SOTM#34 is one of the collected data with the following property:

- **Size**: 3.0MB compressed; about 52.7MB uncompressed.
- **Date collected:** 2005
- **Source system:** Linux RedHat Fedora
- **Format:** tar gzipped
- **Type:** inux */var/log/messages*, Apache */var/log/httpd/access_log/var/log/httpd/error_log*, */var/log/httpd/ssl_error*, IPTABLES firewall log and Snord NIDS logs*/var/log/snortsyslog*

[7] http://old.honeynet.org/scans/scan34/
[8] http://log-sharing.dreamhosters.com/

Table 5.1 Log File Property of SOTM#34

Directory	File	Registers	Entries	Collected time ranges
Http	access_log	Request to the webserver	3554	Jan 30 04:34:59- Mar 17 11:38:27
	error_log	Errors and results of the requests	3692	Jan 30 04:33:18- Mar 17 11:38:27
	ssl_error_log	Errors is ssl connections	374	Jan 30 04:33:18- Mar 16 01:01:43
Iptables	Iptablesyslog	Connection data flowing through the gateway	179752	Feb 25 12:11:24- Mar 31 23:49:38
Snort	Snortsyslog	Alarms triggered by the NIDS snort	69039	Feb 25 12:21:33- Mar 31 23:49:38
Syslog	Maillog	Mail traffic(smtp,pop3)	1172	Jan 30 04:19:27- Mar 17 04:14:33
	Messages	General system messages	1166	Jan 30 04:09:22- Mar 17 13:06:36
	Secure	Login information(sshd,pop3)	1587	Jan 31 04:19:27- Mar 17 04:14:33

Similarly to SOTM#34 log data we also collected and used http web server log data of AAU for the purpose of validating detection capability of our system as shown in Table 5.2.

Table 5.2 Log File Property of AAU HTTP Web Server

Dir	File	Description	Entries	Collected time ranges
Http	access_log	Request to the web server	15868324	07 Dec 2014 06:25:24 28 Aug 2014 09:24:18
	error_log	Errors and results of the requests	4194304	04 sep 2014 22:05:17 07 Dec 2014 06:25:24

The list of sample identified attacks or intrusions in SOTM#34 is shown below in Table 5.3. Those recorded events help us to come up with more attacks detection through clustering and correlation analysis. In addition, they can be used as a reference for administrators to bring their focus in limited devices.

Table 5.3 Some Identified Intrusions from SOTM#34

Date	**Time**	**Source IP**	**Destination IP**	**Source Port**	**Destination Port**	**Protocol**	**Message**
26-Feb 2005	18:57:37	213.135.2.227	-	-	-	-	GET /cgi-bin/awstats.pl HTTP/1.0
26-Feb-2005	18:57:37	213.135.2.227	11.11.79.89	49727	80	TCP	SYN
26-Feb-2005	19:00:43	213.135.2.227	11.11.79.89	50860	80	TCP	BLEEDING-EDGE EXPLOIT Awstats Remote Code Execution Attempt
26-Feb-2005	19:01:44	82.55.78.243	11.11.79.67	24934	80	TCP	SYN
26-Feb-2005	19:01:47	82.55.78.243	11.11.79.67	24934	80	TCP	WEB-ATTACKS rm-command attempt
26-Feb-2005	19:02:00	82.55.78.243	-	-	-	-	http://www.shady.go.ro/awt.Gz
27-Feb-2005	2:55:37	212.203.66.69	11.11.79.67	33847	80	TCP	SYN
27-Feb-2005	2:55:40	212.203.66.69	11.11.79.67	33847	80	TCP	BLEEDING-EDGE EXPLOIT Awstats Remote Code Execution Attempt

5.2 Development Environment and Tools

Several tools and technologies were utilized for the purpose of developing the prototype implementation. The following is list of programming, analysis, database management, and operating environment which are used in the prototype implementation.

- Java programming language is used to write the implementation of the prototype. The Java™ Programming Language is a general-purpose, concurrent, strongly typed, class-based object-oriented language[9]. It is normally compiled to the byte code instruction set and binary format defined in the Java Virtual Machine Specification. It is designed to be platform independent with additional key principles such as usability, reliability and security.
- NetBeans Integrated Development Environment (IDE) version 8 is used for developing the prototype with Java the selected programming language. This version of NetBeans supports ava Platform Standard Edition (Java SE) specification version 7.4 with Java Development Kit (JDK) 8 and Java SE Runtime Environment (JRE) 8[10].
- MySQL database is used to store service related data in the prototype development. It is a widely used open source relational database management system[11].
- Weka is a popular suite of machine learning software written in Java used for analysis which contains a collection of visualization tools and algorithms for data analysis and predictive modeling together with graphical user interfaces for easy access[12].

5.3 Prototype Implementation

The implementation plan follows the process flow of proposed design for providing its demonstration. It tends to provide the proof of concepts in log analysis as security aid and its use for assuring data center security and other related areas. The flow starts from collection of raw log data from various log generating devices of the data center to the final visualization in remediation center.

[9] http://en.wikipedia.org/wiki/Java_%28programming_language%29
[10] https://netbeans.org/
[11] http://www.mysql.com/
[12] http://en.wikipedia.org/wiki/Weka_%28machine_learning%29

Log data provides rich source of information exhibiting the action performed during the usage of computer in the network in our daily work. Log collection considers a collection of such log files from various layered devices with the help of agents to capture the records and feed to the next component for managing centrally and for pre-processing. Log pre-processing is provided with collection of log files from log data source for preparing them for analysis in the central engine component.

Central engine is core component of the system responsible for the overall processing of log files provided by LR. Log Repository (LR) is a component works as a central repository for the preprocessed raw of log data and maintains its availability, confidentiality and integrity. Attack Knowledge Base (ANB) is repository for atomic attack definition obtained from the central engine after analysis.

5.4 Preprocessing Tasks

a. Parsing

It is a first step in which an entire collection of log records in log data source will be segmented(isolated) with defined regular expression to obtain required features of log file. The identification of each feature enables to easily manage each log file and proceed to the higher level log analysis stage.

In order to achieve our objective of identifying holes in the data center infrastructure processing heterogeneous log files plays significant role for administrators to have multidimensional view about the incident. Proper parsing of log files has direct relationship with detection accuracy of abnormal log events from the whole log records. Figure 5.1 shows sample access log parser screenshot and the sample code shown in annex I.

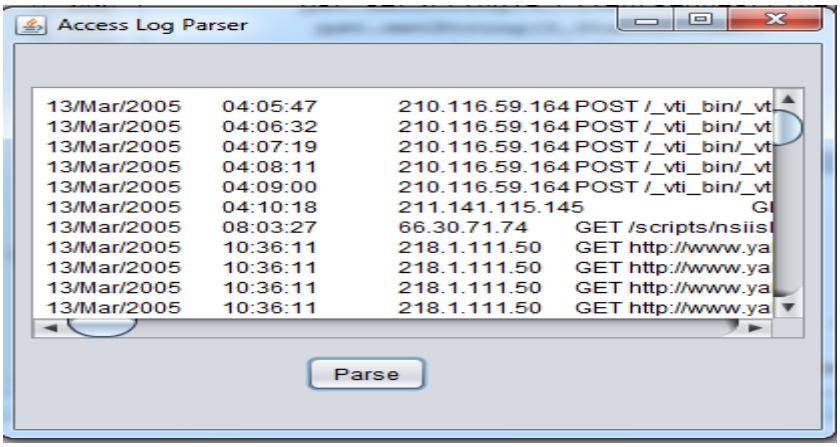


Figure 5.1 Sample Screenshot for Access Log Parser

b. Cleaning

The fact parsed log records contain basically unwanted feature values which have impact on the overall attack identification of the system. In this stage our system can identify a log record with erroneous, missing, incomplete, null values and resolve them accordingly. Cleaned log record aimed to prepare fine tuned data source. Figure 5.2 shows a sample screenshot of cleaned access log records.

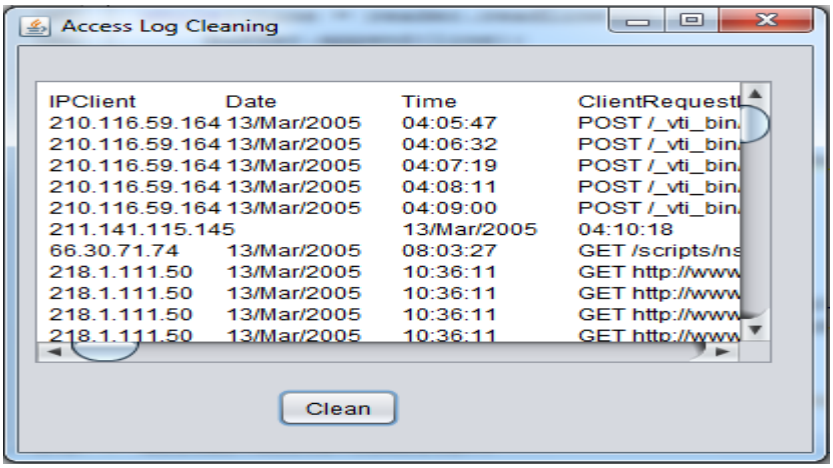


Figure 5.2 Sample Screenshot for Access Log Cleaner

c. Normalization

For this stage cleaned log is taken as an input from previous and then created common log representation for each of log files. To achieve this task we have created a comma separated list

of values which shows uniform occurrence of log records. Figure 5.3 shows sample screenshot of normalized access log file.

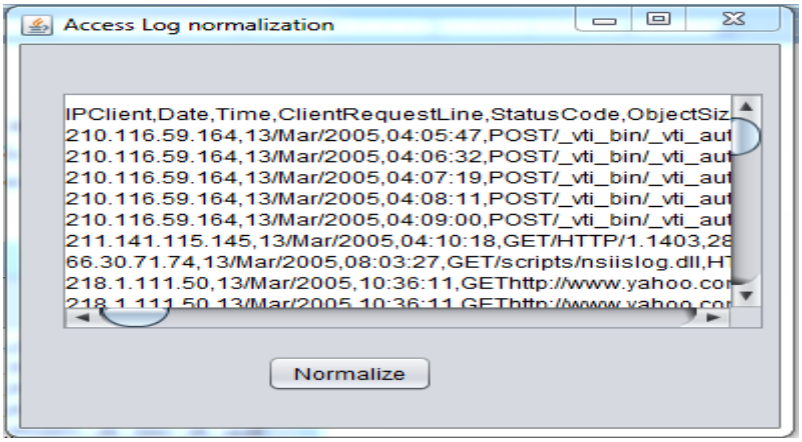


Figure 5.3 Sample Screenshot for Access Log Normalization

d. Aggregation

In the log file repository, we might get log records with similar record values. Processing of similar instances is burden for the system. Aggregating (consolidate) of those various normalized instance is important to alleviate such case. Our system can identify such a record and merge it. The appearance of too many similar log records provides a type of information about characteristics of log records. One of the methods to reduce the complexity of processing an entire collection of log can be aggregating in this manner. Figure 5.4 shows a sample screen shot of aggregated access log file.

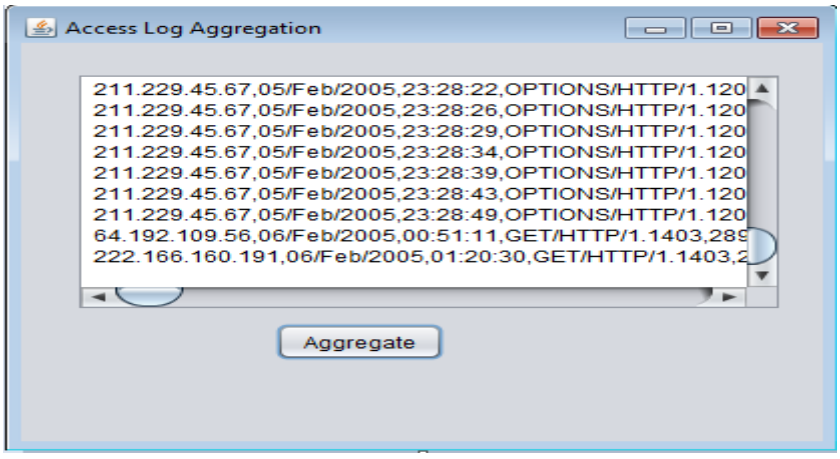


Figure 5.4 Sample Screenshot for Access Log Aggregation

In general, in order to preprocess the whole log data we develop a custom written program used to extract the relevant value from the logs which is incorporated with additional functions, i.e., Isolator. For further processing, the isolator separated the events of iptables firewall log according to connections (TCP, UDP and ICMP) in separate files, since the number of features in the events of this log was not even.

The process of feature selection was used to eliminate insignificant features and boost the precision in predicting and identifying abnormal events so weka's both *CfsSubsetEval* and *BestFirst* search method was used to implement this step. Feature selection was performed by setting every feature in the log as class attributes to know the relationship of the set feature with other features which will assist in precisely identifying the important features. The features selected from each log used for further processing is provided in Table 5.4.

Table 5.4 Features Selected by CfsSubsetEval

Log Type	**Selected Features**
Access	Date, Time, IPClient, ClientRequestLine, StatusCode, ObjectSize, Agent
Error	Date, Time, Severity, ClientIP, Msg
SSL_Error	Date, Time, Severity, Msg
TCP	Date, Time, Direction, PHYSIN, PHYSOUT, LEN, TOS, PREC, TTL, ID, DF, SPT, DPT, WINDOW, RES, STATUS, URGP, SRC, DST
UDP	Date, Time, Direction, PHYSIN, PHYSOUT, LEN, TOS, PREC, TTL, ID, DF, SPT, DPT, LEN, SRC, DST
ICMP	Date, Time, LEN, TOS, PREC, TTL, ID, DF, id, seq, SRC, DST
Message	Date, Time, Daemon, PID, Operation, User, Tty, UID, EUID, Remotehost, Systemmessage
Mail	Date, Time, From, To, Daemon, Mailer, Stat, Priority, Protocol, Message_ID, Relay, Control_address, DSN, Queue_ID, Messages_queued, Messages_delivered, Bytes_queued, Bytes_delivered
Security	Date, Time, Daemon, PID, Operation, User, Source, Systemmessage
Snort IDS	Date, Time, Rulenumber, Rule, Classification, Priority, Protocol, SourceIP, Sourceport, DestinationIP, Destinationport

Common Log Format

Obviously, keeping information in one universal (common) format through bringing all diverse logs events together helps the analysis process of identifying most anomalies. However, including every feature in the logs have an impact to increase the schema size. In order to analyze all of the log events collectively, Common Log Format (CLF) was framed with common and important features which are available in the logs such as timestamp, source IP, destination IP, source port, destination port, protocol and others.

Additionally, message was also included in the log format as it expresses the action performed by the event and their patterns were entirely different for benign and malignant events. The framed format not only specifies the features for CLF, but also the features from various logs that can fit into it. The CLF features and the corresponding features chosen from various logs that fit in GFL were tabulated in Table 5.5. Note that the selection of features for CLF depends on the feature that the log foile contains and the relevance of features for analysis which may vary with different logs.

Table 5.5 Common Log Format Features

Common Log Formats	**Date**	**Time**	**Source IP**	**Destination IP**	**Source Port**	**Destination Port**	**Protocol**	**Message**
Access	Date	Time	IP Client					Client Request Line
Error	Date	Time	Client IP					Message
SSL_Error	Date	Time						Message
TCP	Date	Time	SRC	DST	SPT	DPT	Protocol	Status
UDP	Date	Time	SRC	DST	SPT	DPT	Protocol	
ICMP	Date	Time	SRC	DST			Protocol	
Message	Date	Time						
Mail	Date	Time	Remote Host					System Message
Security	Date	Time	Source		Port			System Message
Snort IDS	Date	Time	Source IP	Destination IP	Source Port	Destination Port	Protocol	Rule

5.5 Central Engine

This component is the core section of our proposed system used to analyze both in terms of learning and predicting the log data using clustering and correlation respectively which is received from the LR so as to distinguish abnormal events. After analysis attack scenario will be built (i.e attack is identified). The following section briefly discuss on functionality of each component in detail.

5.5.1 Clustering

a. Clustering log events

Clustering separates abnormal events from the normal ones taking various log files as input. In our case, the normal events must be identified and reduced prior to analysis to decrease processing overhead. It requires the number of clusters (K) to group the entire log events. As we are using multiple logs, clustering every log with different clusters (K) to choose the best cluster is time consuming. Since we want to separate different event patterns in separate clusters, K was calculated based on the patterns in the log. An arrangement consisting of two steps was used for clustering which are: (i) predicting the best number of clusters K_{ij} for the given log event E_{ij}, and (2) clustering E_{ij} using the manipulated K_{ij}. Applying clustering algorithm separately for different services improves the detection quality, and therefore every single log was treated separately for this step.

In our system we use Expectation Maximization (EM) to find the ideal number of clusters using cross-validation, and predict the best number of clusters in step (i) above, as every cluster relates to a pattern. First, we applied with the default values for the parameters (i.e. seed = 100, K=-1) on the selected features of the individual logs (E_{ij}) to estimate the best number of clusters (K_{ij}). For example, the number of clusters for Apache Access log, EM initially selects eight clusters by cross validation, but ends up with forming only five clusters (0, 1, 2, 6, 7) shown in annex II.

So, the ideal number of clusters chosen for this log was five, and the similar strategy was followed for all the other logs to calculate the ideal number of clusters. The cluster generated by the EM reveals the fact that the number of clusters generated was not influenced by the number

of features and events in a log, but by the patterns of the events. The time taken to predict the ideal number of clusters by EM for each log varied from seconds to minutes depending on the number of features and volume of events. Using the clusters (K_{ij}) manipulated, the logs were clustered and the clustered events (E_{ij}) with different parameter settings were evaluated.

The clusters generated by the algorithms for various settings were verified to substantiate the ability of the algorithm in classifying the abnormal events in separate clusters. In Apache Access log, there were 16 successful intrusive events and 42 unsuccessful events. Under the context of our research, both were considered as anomalies.

b. Filtered events

In our work we filtered log events to remove the normal events at the same time as retaining the abnormal events for further processing. The threshold (E_t) for identifying sparse clusters was calculated based on the number of events and clusters for a particular log, i.e., $E_t = E_{ij} / k_{ij}$. Hence, the characteristics of anomalies were not required for filtering events. The entire step was automated using the script written in Perl which receives a clustered file in ARFF format as input and produces the filtered-in events in CSV format. The performance of this step was evaluated by the number of events reduced and the number of abnormal events retained for further processing.

c. Aggregating filtered-in events

This component combines redundant events by reducing the events in the filtered log. Aggregation merges redundant records into a single record. A set of two or more events was combined, if all the features in the events were exactly similar to the following event(s), and the aggregated event serves as the representative.

Since all intrusive events filtered-in were distinctive, aggregation did not remove any of these events. Less number of filtered-in events was reduced in logs, but further process was not affected by this reduction. Apache access and error log were trimmed to an average of 12.75%, whereas an average of 2% in Linux syslog.

Nearly 75% of the filtered-in events were reduced in SSL_Error log was because of similar event patterns recorded in this log with the same timestamp. Aggregation reduced an average of 20% filtered-in events in Snort IDS log. A similar percentage of reduction was achieved by aggregation with the other three subsets as well. All the events that were retained by the respective logs after it has been filtered and aggregated were abnormal of SOTM#34 kind.

d. Transferring to Common Log Events

The purpose of this component is to extract common features from various aggregated logs as stated in CLF. Unavailability of a particular feature in a log was buffered with a hyphen (-) during transfer. The events in Apache server, Linux syslog and Snort IDS logs were transferred to CLF, excluding Apache SSL_Error and Linux mail log, since many of the CLF features were not available in these logs; but they were maintained separately to be used in the analysis. As iptables firewall log was also used only during analysis, these events were stored separately as per the features specified in CLF.

e. Clustering

Even though log records were clustered and filtered before, there are chances of having less number of normal or insignificant events; due to the inaccuracy in clustering and the subsequent retention by the filtering threshold. To discover such events and also to find the relationship between the events in CLF which contains events from various logs will be re clustered. The accuracy of clustering was evaluated using the different metrics.

f. Detecting anomalous events by analyzing features

The clustered events were analyzed based on features to detect anomalies. We performed two types of analysis: IP address analysis and port number analysis.

IP address analysis: identifies the relation between the IP address of the events from various logs to discover the presence of intrusive events in multiple logs. Therefore, the IP address existing in Linux and Apache events that also exist in Snort IDS events was identified. Not every IP address in Apache and Linux log has a match with Snort IDS log, because SOTM of the abnormal activities may have been overlooked by Snort IDS, but the anomalous patterns

must have joined together in the same cluster during clustering. Therefore, to cover also those abnormal activities which were missed by the Snort IDS log, all events in those clusters to which the identified IP address belongs to were extracted and considered as anomalous.

Port number analysis: identifies anomalous events based on port numbers. This was not an alternative to IP Address analysis, but to capture those anomalous events which were available in those clusters that were not captured by it. Most anomalous activities were launched from a host by utilizing the unassigned and dynamic ports, since no predetermined service was running on these ports. In the case of inbound connections the source port number of the events was checked against the listing of the dynamic and unassigned port numbers as per Internet Assigned Numbers Authority (IANA)[13].

5.5.2 Correlation

The process of correlation is defined by finding the relationship among log events to produce a series of abnormal events (simply attacks) observed in the log files. In our system we tried to perform this by preparing a search query and send it to the ANB as shown in Figure 5.5 and if the intended result is found then it will create scenario which is shown in Figure 5.6 and if it is not found then it will register as a new record which is illustrated in Figure 5.7. Figure 5.5 shows a log correlation process to search attack pattern.

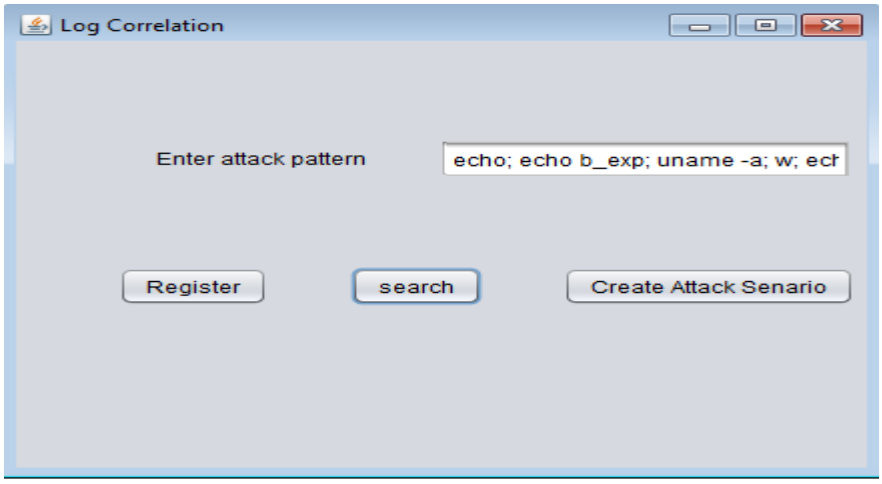


Figure 5.5 Sample Screenshot for Log Correlation Process

[13] http://www.ietf.org/assignments/port-numbers

The following figure shows attack scenario created for search result of submitted attack pattern.

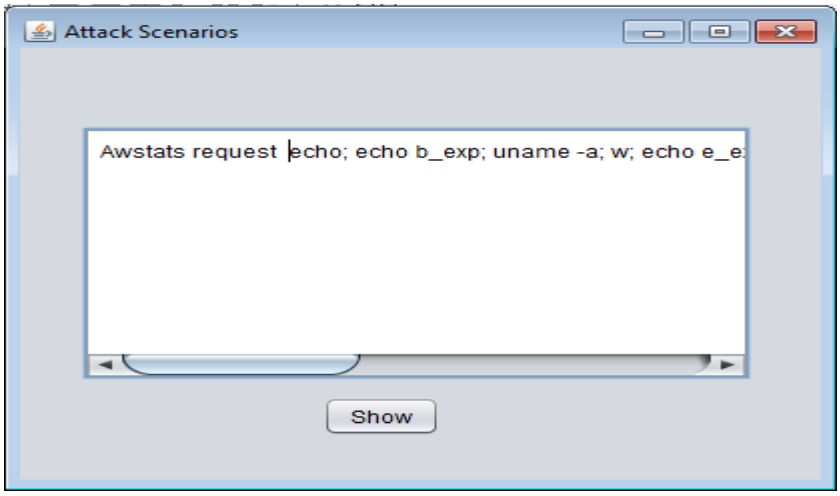


Figure 5.6 Sample Screenshot for Attack Scenario

The following figure shows attack registration form for new attacks which are not found in the ANB.

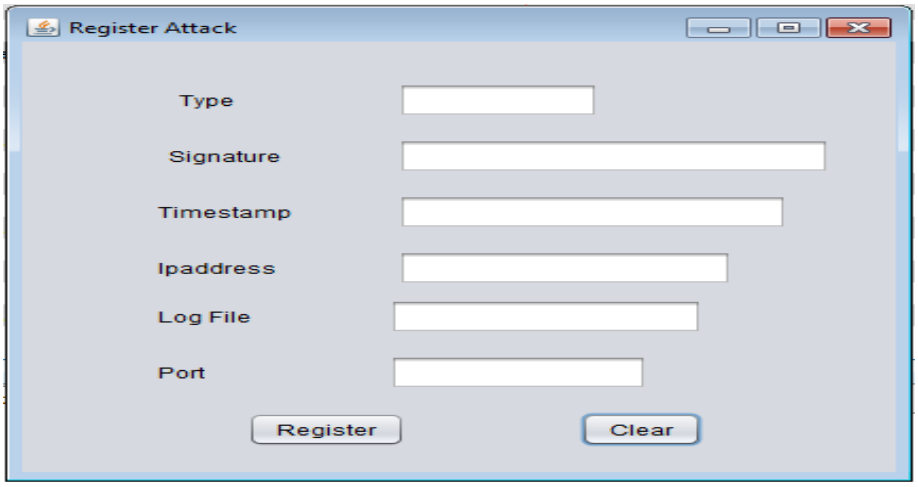


Figure 5.7 Sample Screenshot for Registering Attack

5.6 Result and Discussion

The quality of clusters produced by clustering algorithms was calculated using Weka Experimenter with 10 fold cross validation and 10 iterations to allow every part of the log to be tested. True Positives (TP), True Negatives (TN), False Positives (FP) and False, Negatives (FN) were measured to calculate the precision and False Positive Rate (FPR) of clustering.

Precision can be calculated by dividing true identified log events by total number of clustered logs (i.e. precision=TP/Total number of clustered log events).

In addition, False Positive Rate is calculated by dividing incorrectly identified log events to a given cluster by the sum of incorrectly identified log events to a given cluster and correctly clustered log events in to other cluster (i.e FPR = FP / (FP + TN)). Apart from that, the clusters generated were examined to evaluate the precision of the algorithms in separating abnormal events from the normal once. The evaluation of clustered log events for SOTM#34 and our case AAU is summarized and shown in Table 5.6 and Table 5.7 respectively.

Table 5.6 Evaluation of Clustered Log Events for SOTM#34

File	**Log File**	**Clustered log records**	**Evaluation**				
			TP	**FP**	**TN**	**Precision (%)**	**FPR (%)**
Http	access_log	3554	3281	73	97	92.31	42.94
	error_log	3692	3450	192	44	93.44	81.35
	ssl_error_log	374	298	76	21	79.6	78.35
Iptables	Iptablesyslog	179752	168269	11034	1102	93.61	90.91
Snort	Snortsyslog	69039	58480	10559	527	84.70	95.24
Syslog	Maillog	1172	794	378	80	67.74	82.53
	Messages	1166	935	531	204	80.18	72.24
	Secure	1587	1449	138	36	91.30	79.31

Table 5.7 Evaluation of Clustered Log Events for AAU

File	**Log File**	**Clustered log records**	**Evaluation**				
			TP	**FP**	**TN**	**Precision (%)**	**FPR (%)**
Http	access_log	15868324	14676231	892093	9578	92.48	98.93
	error_log	4194204	3672201	322003	5697	87.55	98.26

Chapter Six - Conclusions and Future Works

6.1 Conclusions

Following results of this research the process of finding abnormal events (attacks) out of the whole log entries of those critical log generating devices can be easily conducted and possible solution can be determined. An overview of data center, data center security, trends for data center security, log file analysis and its approaches is provided. Data center security is a continual, planned and managed means of assuring immunity for the infrastructure critical and other elements. Through reviewing various related researches we obtained a concept of different approaches of log analysis that provides security for an organization data center. The drawbacks of those researches are the gaps in which this research work addressed.

In our study, we presented an automated log analyzer system architecture organized in a layer for enhancing the security of enterprises data center. The system analyzes a set of heterogeneous log files produced by those layered devices appearing in the infrastructure for achieving the goal of this study by collecting log data, preprocess them, build central engine for analysis and taking remedial measure through action center are the main processes of the proposed system.

Log data collection concerned with gathering various types of log files and put all of them into a container centrally. In pre-processing phase the data is well prepared and adjusted to come up with better detection result by feeding to our system. The central engine phase is to twofold:- first, the clustering component groups the log entries and filter anomalous events based on filtering threshold. Depending on the pattern of log events they will be grouped into well defined clusters.

Once clustering is completed then the correlation finds a relationship among the log record events within two consecutive steps. Firstly, an atomic attack definition is extracted from attack knowledge base used in order to search specific attack action with pre defined attack argument (parameter) and secondly it computes that, if the attack definitions found then the correlation process construct attack scenarios describing the attacks detail. On the other hands, if the attack definition is not found then, it will be registered in the ANB as a new entry. Finally, to show

results available the action center represents those scenarios through displaying various visualizations (like graph, reports, messages and others) to facilitate the interpretation for the administrator.

The central engine module in our work is a core of the entire system performing log processing or analysis task. It is used for finding security holes in bidirectional fashion using both clustering and association techniques. The efficiency of this component is mainly determined by its preceding stages. Its effectiveness determines the performance of the log file analyzer system.

In order to validate the usability of our system we used a real network traffic based log data of both SOTM#34 and Addis Ababa University data center devices generated were used. Depending on those inputs we successfully found several attack actions which leads to the construction of an attack scenarios.

The result found encourages us to advance the work and test various clustering algorithms with different parameters and use heterogeneous log data set based on the proposed solution. Since the goal of our work is to reduce the possibility of obtaining false positive and false negative the result shows that the direction that clustering can help to reduce them. In general, our study have achieved its objective in terms of bringing new dimension for enhancing security of enterprises data center by analyzing critical devices generated log files with reasonable result.

6.2 Contributions

The contribution of this thesis work is summarized as follows:

- Provide log file analyzer system architecture with capabilities of identifying anomalous events from normal events having heterogeneous log files.
- Perform tasks such as log collection, log preprocessing, log analysis with central engine and show events detail.
- Determine security of the data center through analyzing heterogamous collection of log files and identify breaches (holes).
- Build attack knowledge base containing atomic attack definitions.

- Support administrators to adjust their security setting for the prevention and monitoring of security attacks which ensures a continual means of guaranteeing the health of data center environment.
- Develop a prototype system.

6.3 Future Works

The following are some of the potential future works to the continuation of our work.

- Develop an online log analysis framework for real time attack detection.
- Apply more approaches or techniques to obtain useful knowledge and reduce false positive and false negative results.
- Consider various parameter settings to obtain accurate detection result.
- Take into account for more log records to make the work generic.
- Understands users' behavior from analyzed logs.
- Use attack knowledge base for the sake of forensics, auditing and others.
- Securing log data sources from intruders' corruption.
- Develop a log file compression strategy for limited storage reason.

References

[1] Knapp Ken, Gary D. Denney, and Mark E. Barner, "Data Center Security: Analysis of Two Audit Reports", *In Proceedings of the Southern Association for Information Systems Conference*, Charleston, pp. 5-10, March 12 -14, 2009.

[2] Li Qian, "Study of Information System Security of Government Data Center Based on the Classified Protection", *Computer Science and Education (ICCSE) 8th International Conference*, IEEE, pp. 1315-1319, 26-28 April 2013.

[3] Carrie Roberts, "Discovering Security Events of Interest Using Splunk", Retrieved from: *http://www.sans.org/reading-room/whitepapers/logging/rss/-34272*, Accessed on: July 05 2014.

[4] Krishna Kant,"Data Center Evolution: A Tutorial on State of the Art, Issues, and Challenges", International Journal of *Computer Networks*, Vol. 53, No. 17, pp. 2939-2965, 2009.

[5] Theophilus Benson, Aditya Akella, and David A. Maltz, "Network Traffic Characteristics of Data Centers in the Wild", *In Proceedings of the 10th annual conference on Internet measurement, ACM*, Microsoft Research, New York, USA, pp. 267–280, 2010.

[6] Manoj Patel, Manish Shrivastava, and Kavita Deshmukh, "Simulation and Assessment of Network Security Based on System Dynamics", *International Journal of Scientific Engineering and Technology*, Vol. 1, Issue No. 2, pp. 17-21, 01 April 2012.

[7] Khasnabish B., Dong W., Karavettil S., and Ning S., "Security Framework for Virtualized Data Center Services", *Internet Engineering Task Force (IETF) Internet Draft*, Vol. 5, December 26, 2012.

[8] Yigezu Balcha Jorro, "Information System Security Audit Readiness Case study: Ethiopian Government Organizations", *Unpublished Master's Thesis*, Royal Institute of Technology, KTH, Department of Computer and Systems Sciences, 2011.

[9] Onwubiko Cyril, "A Security Audit Framework for Security Management in the Enterprise, "*5th International Conference in Global Security, Safety, and Sustainability*", London, UK, Vol. 45, pp. 9-17, 2009.

[10] Justin Kapp, "How to Conduct a Security Audit", *International Journal of PC-Network Advisor*, pp. 3-8, Issue 120, 2000.

[11] Jalal Frihati, Florica Moldoveanu, and Alin Moldoveanu, "General Guidelines for the Security of a Large Scale Data Centre Design", *University Politehnica of Bucharest Sci. Bull.*, Series C, Vol. 71, Issue 3, 2009.

[12] Krishna Kant, Meixing Le, and Sushil Jajodia, "Security Considerations in Data Center Configuration Management", *In Configuration Analytics and Automation (SAFECONFIG), 4th Symposium*, pp. 1-9, IEEE, Oct 31-Nov 1, 2011.

[13] Ho-Yu Lam, Song Zhao, Kang Xi, and Chao H.J., "Hybrid Security Architecture for Data Center Networks", *IEEE International Conference in Communications (ICC)*, pp. 2939 2944, 10-15 June 2012.

[14] Udeze Chidiebele C., Okafor Kennedy C., Inyiama H.C., and Okezie C.C., "Effective Security Architecture for Virtualized Data Center Networks", *International Journal of Advanced Computer Science Application*, Vol. 3, Issue 1, pp. 196-200, Jan 2012.

[15] Nath, Kapil Kumar Gupta Baikunth, and Kotagiri Ramamohanarao, "Network Security Framework", *International Journal of Computer Science and Network Security (IJCSNS)*, Issue 120, 2006.

[16] James H. Andrews, "Testing Using Log File Analysis: Tools, Methods, and Issues", *In Proceedings of the 13th IEEE International Conference on Advanced Software Engineering*, pp. 157-166, Honolulu, Hawaii, October 1998.

[17] Sewale Belachew, "A Cloud Computing Framework for Ethiopian Higher Education Institutions", *unpublished Master's Thesis*, School of Information Science, Addis Ababa University, June 2012.

[18] Jacques Saraydaryan, Fatiha Benali, Stéphane Ubeda and Véronique Legrand "Comprehensive Security Framework for Global Threats Analysis*", International Journal of Computer Science Issues (IJCSI)*, pp. 18-32, Vol. 2, 2009.

[19] Nashaat el-Khameesy, Hossam Abdel, and Rahman Mohamed, "A Proposed Model for Datacenter in Depth Defense to Enhance Continual Security", *International Journal of Information Technology and Computer Science*, pp. 55-67, March 2013.

[20] Kuipers D., and Fabro M., "Control Systems Cyber Security: Defense in Depth Strategies", *unpublished Master's Thesis*, United States, Department of Energy, 2006.

[21] Kazimierz Kowalski and Mohsen Beheshti, "Improving Security through Analysis of Log Files Intersections", *International Journal of Network Security*, Vol. 7, No. 1, pp. 24 - 30, July 2008.

[22] Pingchuan Ma, "Log Analysis-Based Intrusion Detection via Unsupervised Learning", *Unpublished Master's Thesis*, Master of Science in School of Informatics, University of Edinburgh, UK, 2003.

[23] Ahmed Youssef and Ahmed Emam, "Network Intrusion Detection Using Data Mining and Network Behaviour Analysis", *International Journal of Computer Science and Information Technology*, Vol. 3, pp. 87-98, 2012.

[24] Deepak Upadhyaya and Shubha Jain, "Model for Intrusion Detection System with Data Mining", *International Journal of Advanced Research in Computer Engineering and Technology*, pp. 145-148, Volume 1, Issue 4, June 2012.

[25] Anish Gupta, Vimal Bibhu, and Rashid Hussain, "Security Measures in Data Mining", *International Journal of Information Engineering and Electronic Business*, Vol. 3, pp. 34-39, 2012.

[26] Prashant Achari, Susanta Adhikary, Jayashree Madugundu, and Mungara Jitendranath "Knowledge and Rule Based Learning Engine to Analyze the Logs for Troubleshooting", *International Journal for Scientific Research and Development*, Vol. 2, Issue 04, 2014.

[27] Kallol Bagchi and Godwin Udo, "An Analysis of the Growth of Computer and Internet Security Breaches", *International Journal of Communications of the Association for Information Systems*, Vol. 12, Issue 1, pp. 684-700, 2003.

[28] Jorge Herrer´ıas and Roberto G´omez, "Log Analysis Towards an Automated Forensic Diagnosis System", *International Conference on Availability, Reliability and Security*, pp. 659-664, 15-18 Feb 2010.

[29] Sorot Panichprecha, "Abstracting and Correlating Heterogeneous Events to Detect Complex Scenarios", *unpublished PhD Thesis*, Queensland University of Technology, Brisbane, Australia, March 2009.

[30] Fageeri, Sallam Osman, and Rohiza Ahmad, "An Efficient Log File Analysis Algorithm Using Binary Based Data Structure", *Procedia-Social and Behavioral Sciences*, Vol. 129, pp. 518-526, 2014.

[31] Lawrence A. Gordon, Martin P. Loeb, William Lucyshyn and Robert Richardson, "CSI/FBI Computer Crime and Security Survey", *Book*, Technical Report, 2006.

[32] Frederick M. Avolio, Steve Fallin, and D.Scott Pinzon, "Producing Your Network Security policy", *Book*, Available at: Watchguard.com, July 2007.

[33] Lance Cleghorn, "Network Defense Methodology: A Comparison of Defense in Depth and Defense in Breadth", *International Journal of Information Security*, Vol. 4, No. 3, pp. 144-149, 2013.

[34] Xin Yue, Wei Chen, and Yantao Wang, "The Research of Firewall Technology in Computer Network Security", *International Journal of Computational Intelligence and Industrial Applications*, IEEE, Asia-Pacific Conference, vol. 2, pp. 421-424, 28-29 Nov. 2009.

[35] Choi, Young B., Chrisopher Sershon, John Briggs, and Chad Clukey, "Survey of Layered Defense, Defense in Depth and Testing of Network Security", *International Journal of Computer and Information Technology*, Vol. 03, Issue 05, September 2014.

[36] Gyan Prakash Pal, and Sadhana Pal, "Virtual Local Area Network (VLAN)", *International Journal of Scientific Research Engineering and Technology (IJSRET)*, Volume 1, Issue 10, pp. 006-010, January 2013.

[37] Ritika kajal, Deepshikha Saini, and Kusum Grewal, "Virtual Private Network", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 2, Issue 10, pp. 428-432, October 2012.

[38] Nen-Fu Huang, Chia-Nan Kao, Hsien-Wei Hun, Gin-Yuan Jai, and Chia-Lin Lin, "Apply Data Mining to Defense-in-Depth Network Security System", *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (ICAINA)*, 2005.

[39] Choi Young B., Chrisopher Sershon, John Briggs, and Chad Clukey, "Survey of Layered Defense, Defense in Depth and Testing of Network Security", *International Journal of Computer and Information Technology*, Vol. 03, Issue 05, September 2014.

[40] Perrin C., "Understanding Layered Security and Defense in Depth", Retrieved from: *http://www.techrepublic.com/blog/it-security/understanding-layered-security-and-defense-in-depth/*, *Book*, Tech Republic, Accessed on: September 23, 2014.

[41] Veda A., KReSIT I. I. T., Kalekar P., and Bodhankar A., "Intrusion Detection Using Data Mining Techniques", Report IIT Bombay, 2006.

[42] TechNet, "Common Types of Network Attacks" Microsoft, Retrieved from: *http://technet.microsoft.com/en-us/library/cc959354.aspx*, Accessed on December 16, 2014.

[43] Nikhil Kumar Singh, Deepak Singh Tomar, and Bhola Nath Roy, "An approach to understand the end user behavior through log analysis", *International Journal of Computer Applications*, Volume 5, No. 11, pp. 27-34, August 2010.

[44] Nathaphon Kiatwonghong and Songrit Maneewongvatana, "Intelli-log: A Real-time Log Analyzer", *2nd International Conference on Education Technology and Computer (ICETC)*, pp. 22-24, June 2010.

[45] Jan Valdman, "Log File Analysis", Department of Computer Science and Engineering, *Unpublished PhD Thesis***,** Tech. Rep. DCSE/TR-2001-04, 2001.

[46] Sakha A., "Cyber-Forensic Log Analysis", *Unpublished Doctoral dissertation*, Concordia University, 2008.

[47] Adam Oliner, UC Berkeley, Archana Ganapathi, and Splunk Wei Xu, "Advances and Challenges in Log Analysis", December 20, 2011.

[48] Blask, Chris, S. Harris, A. A. Harper, D. Miller, and S. Van Dyke, "Security Information and Event Management (SIEM) Implementation", *Book*, 2010.

[49] Muhammad Kamran Ahmed, Mukhtar Hussain and Asad Raza, "An Automated User Transparent Approach to log Web URLs for Forensic Analysis", *Fifth International Conference on IT Security Incident Management and IT Forensics*, 2009.

[50] Leite Jorge Pinto, "Analysis of Log Files as a Security Aid", *In Information Systems and Technologies (CISTI), 6th Iberian Conference*, IEEE, pp. 1-6, 2011.

[51] Asif-Iqbal H., Nur Izura Udzir, Ramlan Mahmod, and Abdul Azim Abd Ghani, "Filtering Events Using Clustering in Heterogeneous Security Logs", *Information Technology Journal*, Vol.10, No. 4, pp. 798-806, 2011.

[52] Ghani Abdul, "An Unsupervised Heterogeneous Log Based Framework for Anomaly Detection", *Unpublished Master's Thesis*, Faculty of Computer Science and Information Technology, Universiti Putra, Malaysia, 2005.

[53] Shengyan Shi, Shen Xiaoliu, Zhao Jianbao, and Ma Xinke, "Research on System Logs Collection and Analysis Model of the Network and Information Security System by Using

Multi-agent Technology", *Fourth International Conference In Multimedia Information Networking and Security (MINES)*, pp. 23-26, IEEE, 2012.

[54] Söderström, Olof and Esmiralda Moradian, "Secure Audit Log Management", *Procedia Computer Science*, pp. 1249-1258, 2013.

[55] Herrerias Jorge and Roberto Gomez, "A Log Correlation Model to Support the Evidence Search Process in a Forensic Investigation", *Second International Workshop in Systematic Approaches to Digital Forensic Engineering (SADFE 2007)*, pp. 31-42, IEEE, 2007.

[56] Madani Afsaneh, Saed Rezayi, and Hossein Gharaee, "Log Management Comprehensive Architecture in Security Operation Center (SOC)", *International Conference in Computational Aspects of Social Networks (CASoN)*, IEEE, pp. 284-289, 2011.

[57] Vaarandi Risto, "Mining event logs with slct and loghound*", Network Operations and Management Symposium (NOMS)*, IEEE, pp. 1071-1074, 2008.

[58] Sayed Omid Azarkasb and Saeed Shiry Ghidary, "Logs Correlation: Current Approaches, Promising Directions, and Future Policies", *Journal of Basic and Applied Scientific Research*, Vol. 2(5), pp. 4413-4322, 2012.

[59] Kowalski K. and Beheshti M., "Analysis of Log Files Intersections for Security Enhancement", *Third International Conference in Information Technology: New Generations (ITNG)*, pp. 452-457, 10-12 April 2006.

[60] J.H. Andrews, Yingjun Zhang, "General Test Result Checking with Log File Analysis*", In IEEE Transactions on Software Engineering*, Vol. 29(7), pp. 634–648, July 2003.

[61] J.H. Andrews, Yingjun Zhang, "Broad Spectrum Studies of Log File Analysis", *In Proceedings of the 22nd International Conference on Software Engineering (ICSE)*, pp. 105-114, Limerick, Ireland, June 2000.

Annexes

I. Sample Code for Access Log Parser

```
//Access log parser
package parsing;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Scanner;
import java.util.regex.*;
import java.sql.*;

interface Log {


   public static final int NUM_FIELDS = 9;

   /**
    * The sample log entry to be parsed.
    */
}

/**
 * Parse an Apache log file with Regular Expressions
 */
public class ApacheLogParser implements Log {

   public static void main(String argv[]) throws FileNotFoundException, IOException,
SQLException  {

    Scanner sc = new Scanner(new File("D:\\log data
collected\\SotM34\\merged\\http\\access\\ACCESSLOG.txt"));
Connection conn = null ;
```

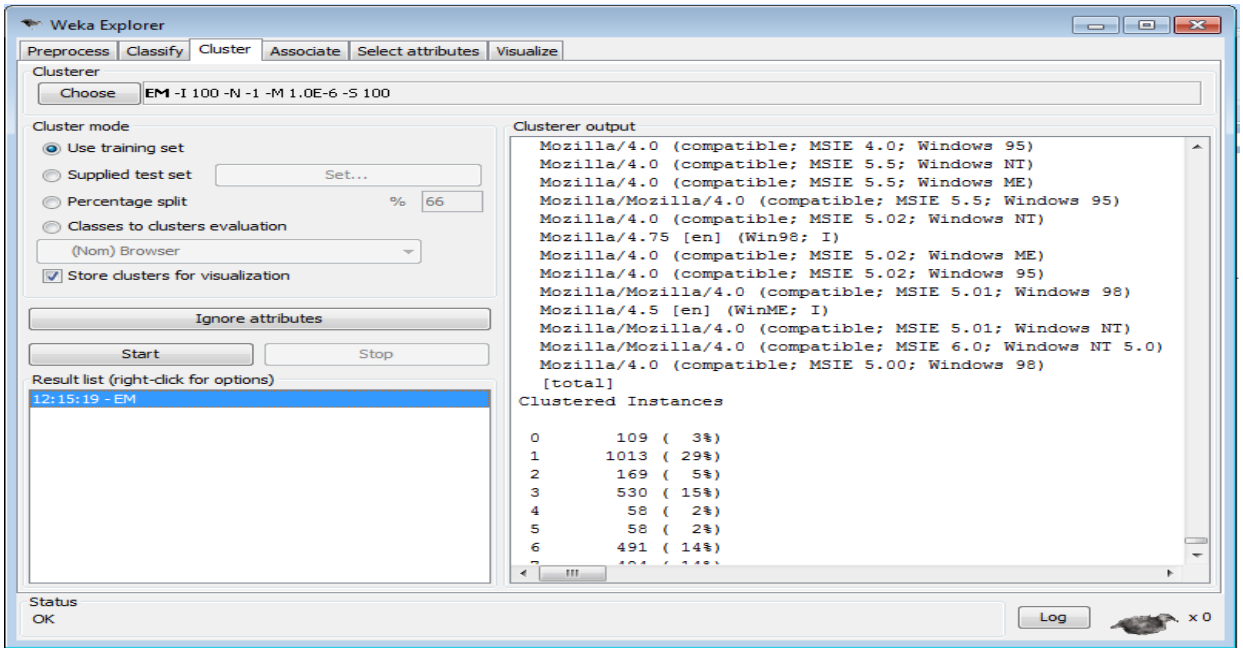
```
    Statement  st = null;
    ResultSet rs = null;
      PreparedStatement pst = null;

       String line = "";
        String com=",";

File logfile = new File("D:\\Accesslog.csv");

       String arr[] = new String[9];
       arr[0] = "IPClient";
       arr[1] = "Date";
       arr[2] = "Time";
       arr[3] = "ClientRequestLine";
       arr[4] = "StatusCode";
       arr[5] = "ObjectSize";
       arr[6] = "Referer";
       arr[7] = "Browser";

       String name = ""+ arr[0] + com + arr[1] + com + arr[2] + com + arr[3] + com + arr[4] +
com + arr[5] + com + arr[6] + com + arr[7] + "\t";

       // BufferedWriter wr = new BufferedWriter(new FileWriter(logfile));
       BufferedWriter writer = new BufferedWriter(new FileWriter(logfile));

          String logEntryPattern = "^([\\d.]+) (\\S+) (\\S+) \\[([\\w:/]+\\s[+\\-]\\d{4})\\] \"(.+?)\"
(\\d{3}) (\\d+) \"([^\"]+)\" \"([^\"]+)\"";


          writer.write(name+ "\n");

          int count = 0;
          conn = condb.ConnecrDbc();
          while (sc.hasNext()) {
             //System.out.println("line number " + ++count);
             line = sc.nextLine();
             String l = line;
```

```
        //String
a="matcher.group(1)+matcher.group(4)+(matcher.group(5))+(matcher.group(6))+(matcher.grou
p(7))+(matcher.group(8))(matcher.group(9))" ;
        Pattern p = Pattern.compile(logEntryPattern);
        Matcher matcher = p.matcher(line);

        if (!matcher.matches()
            || NUM_FIELDS != matcher.groupCount()) {
          System.err.println("Bad log entry (or problem with RE?):");
          System.err.println(line);
          //writer.write(line + "\n");
          return;
        } else {
          String IPClient = "" + matcher.group(1);
          String DateTime = "" + matcher.group(4);
          DateTime = DateTime.replace("-0500", "");
          String[] dt = DateTime.split(":", 2);
          dt[1]=dt[1].trim();
          dt[1]=dt[1].replaceAll("\\s+","");
          String ClientRequestLine = "" + matcher.group(5);

          String StatusCode = "" + matcher.group(6);
          String ObjectSize = "" + matcher.group(7);
          String Referer = "" + matcher.group(8);
          String Browser = "" + matcher.group(9);

           System.out.println("ID: " + count++);
          System.out.println("Date: " + dt[0]);
          System.out.println("Time: " + dt[1]);
          System.out.println("IPClient:" + IPClient);
          System.out.println("ClientRequestLine: " + ClientRequestLine);
          System.out.println("StatusCode: " + StatusCode);
          System.out.println("ObjectSize: " + ObjectSize);
          if (matcher.group(8).equals("-")) {
            System.out.println("Referer: " + Referer);
            Referer="";

          }

          if (matcher.group(9).equals("-")) {
```

```
                System.out.println("Browser: " + Browser);
                Browser="";

            }
             System.out.println("Browser: " + Browser);


      try
      {

       String sql= "insert into access
(IPclient,Date,Time,ClientRequestLine,StatusCode,ObjectSize,Referer,Browser) values
(?,?,?,?,?,?,?,?)";
//       st =con.prepareStatement(sql
       String date=dt[0];
       String time=dt[1];
      pst = conn.prepareStatement(sql);
      pst.setString(1,IPClient );
      pst.setString(2,date);
      pst.setString(3,time);
      pst.setString(4,ClientRequestLine);
       pst.setString(5,StatusCode);
        pst.setString(6,ObjectSize);
        pst.setString(7,Referer);
        pst.setString(8,Browser);
      pst.execute();

      }catch(Exception ex){
   System.out.println(ex);

}

            String writeLine = IPClient + com + dt[0]+ com+ dt[1] + com+ClientRequestLine+
com + StatusCode + com+ ObjectSize + com + Referer + com + Browser + "\n";
            writer.flush();
            writer.write(writeLine);
   }
}
         conn.close();
   }}
```

II. Demonstration Sample for Clustering Access Log

The following figure shows sample for clustering access log using weka data mining tool which takes preprocessed log data from LR to cluster into defined group of clusters.



Declaration

I, the undersigned, declare that this research is my original work and has not been presented for degree in any other university, and that all sources of materials used for the research have been acknowledged.

Declared by:

Name: Samuel Getachew

Signature: _____

Date: _____

Confirmed by advisor:

Name: Dejene Ejigu (PhD)

Signature: _____

Date: _____