R-ELK STACK FOR LOG ANALYSIS USING CUSTOMIZED IDS SIGNATURES

By

Hassan Ishfaq

A thesis submitted to the faculty of Information Security Department, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of MS in Information Security

July 2017

ABSTRACT

Now a days, almost every device including computer, routers, switches, firewalls, software and services generate logs continuously. As number of devices in any large network are large and grow with every new installation, it is administratively becoming less feasible to do monitoring and analysis of each device.

With the growing trend of big data of logs, companies tend to rely more on expensive SIEM solutions for log analysis. However, with the introduction of open source, lightweight and rich featured Search Engine Database models the approaches towards searching data content have become ubiquitous. Proposed System uses open source Generic search engine Elastic Search with other components in order to process large amount of logs and detect attacks via developed IDS signatures through Attack Signature Framework.

ACKNOWLEDGMENTS

All praises to Allah for the strength and His blessing in completing this thesis.

I would like to convey my gratitude to my supervisor for his supervision and constant support. His invaluable help of constructive comments and suggestions throughout the experimental and thesis works are major contributions for the success of this research. Also, I would thank my co-supervisor and committee member for their support and knowledge regarding this topic.

Last, but not the least, I am highly thankful to my mother and brother (Hannan Ishfaq). They have always stood by my dreams and aspirations and have been a great source of inspiration for me. I would like to thank them for all their care, love and support through my times of stress and excitement.

TABLE OF CONTENTS

List of figures

Figure# **Caption** **Page**

LIST OF TABLES

ACRONYMS

APT	Advanced Persistent Threat
CSV	Comma Seperated values
DOC	Document in ElasticSearch
DSL	Domain Specified Language
ELK	Elastic Search with Kibana
IDS	Intrution Detection System
JSON	Javascript Object Notation
LCE	Corelation of Log Enteries
NOC	Network operation Center
PDF	Portable Document Format
PVS	Passive Vulnerability Scanner
QueryDSL	Domain Specified Language Query Language based on Json
REGEX	Regular Expression
R-ELK	Redis with Elastich Search and Kibana
SEM	Security Event Management
SIEM	Security Information and Event Management
SIM	Security Informtion Management
STRIX	Structured threat Information Expression
TAXII	Trusted Automated eXchange of Indicator Information
XML	eXtensibale Markup Language

Chapter 1

INTRODUCTION

1.1 Introduction

Logs are the records of events, and messages generated by the operating systems as well as the software, hardware devices, and services. Any device in a network or that is stand-alone generates logs continuously. In all networks logs grow with every new installation or implementation and it becomes administratively less feasible to monitor and analyze log of every device.

With the growing trend of such big data of logs, companies are relying more and more on expensive SIEM solutions for log analysis. However, with the introduction of open source, lightweight and rich featured Search Engine Database models the approaches towards searching data content have become ubiquitous. The proposed System uses an open source generic search engine, Elastic Search, and other components to process a large number of logs and to detect attacks via developed IDS signatures through Attack Signature Framework.

1.2 Problem Statement

The development of ELK stack that will be using customized IDS signature is required[25] because of the following reasons

- Commercial solutions come in executable format, with no source code available
- Have limited capabilities in terms of customization
- Does not target specific needs of the organization
- Freeware OS SIEM solutions do not process large number of logs

1.3 Research Objective

This research provides solution for problem statement by purposing solution for opensource log analysis and IDS signatures for indexed logs.

This primarily objective is to gain the following goals :-

- Implementing and analysing the R-ELK stack components for using in log processing, indexining and searching
- Developing signatures to be used with R-ELK stack to act as IDS signature
- Developing related tools to facilitate implementation
- Evaluating the results for attacks related to signatures

1.4 Research Methdology

The research is based on implementation and evaluation of implementated system components in different contients of the world. The deployed components were integrated as per their role in order to act as a whole system. Different types of test scanarios were applied to the system in order to test the functionality of the system.

1.5 Purpose and Scope of Proposed System

The proposed system will act like SIEM, to get the logs from all the defined sources, with the following features:

- Using open source and free search engines for centralized log searching, storage, and analyzing compatibility instead of using commercial solutions.
- Developing signatures for search engines to automate the detection of attacks.
- The system should be extensive, flexible and support parallel, distributed architecture.

1.6 Log and Log Types

A log is usually a documented file (or various files) created automatically and conserved by a server that comprises a list of the activities it has performed. Every time an action is done, the server logs the actions of OS, hardware, services, or software. However, based on the configuration, the actions performed are entered from the particular service or software into the log file.

In a quantity context, a log is spontaneously created time-embossed events documentation that is relevant to a specific system. Almost all system applications and software produce log files.

1.6.1 Web Servers

A web server is that program that utilizes Hypertext Transfer Protocol (HTTP) to offer files that display Web pages to its users, as a response to their demand, which is then redirected by the HTTP clients of their computers. Dedicated appliances and computers can also be termed as “web servers”.

Such a process is a typical example of the server/client model. It is essential for every computer that hosts web sites to have web server programs. The leading web servers include the most generally-installed web server (Apache), NGINX (prominent engine X) and IIS (Microsoft's Internet Information Server)[38]. Some other web servers are part of IBM products likeDomino servers, NetWare server of Novell and GWS (Google Web Server).

Web servers mostly come as an integral part of an enormous package of intranet related and Internet programs for the purpose of downloading, serving e-mail, requests for FTP (File Transfer Protocol) files, publishing and building web pages. What factors should be considered when choosing a web server include how it functions with the operating system and all other servers, how it controls security properties, server-side programming languages like php, .net and many others. It also comes with specified search engine and site building tools and publishing apps.

1.6.2 Apache web Server

This is an open source web server product. It was first maintained by a group of software programmers, but is now developed by the Foundation of Apache Software[39]. As of July 2016, Apache serves approximately 46.41percent of all active websites and about 43.18percent of the top one million websites. Apache can create diverse kinds of logs. The two most important kinds are the error log that is made to debug messages and also logs every unusual event that occurs and providing multiple information. The other important log is the access log in which all clients’ requests are observed. The access log is produced and written by the module mod_log_config[40], which is an important module created by community, although this module is not part of the major official releases.

1.6.3 Internet Information Services

Internet Information Server (IIS) is an Internet server group (which includes a File Transfer Protocol server, Hypertext Transfer Protocol server, and the web) with added abilities for Windows 2000 Server and Microsoft's Windows NT operating systems. IIS is accompanied with COM logging modules which help to log site activity in several formats[41]. The IIS may be configured for FTP site or website to document log entries that are derived from the activity of the server and also that of the user.

IIS can store log files to the various file formats for the users[41]. Enabling logging may define the file format that you wish to implement. IIS utilizes the Extended W3C log file format by default. Mostly, the Extended W3C log file format is the preferred choice of log type by many. The log format allows you to configure many extended attributes that help in analyzing security.

1.6.4 Nginx

Nginx (prominent engine X) is a web server. It may act as a converse proxy server for HTTPS, HTTP, POP3, IMAP, and SMTP protocols, an HTTP cache, and a load balancer[42]. Nginx is an HTTP open source web server and converse proxy server. After Microsoft’s IIS and Apache Web Server, Nginx is the third most commonly used web server[44], which currently powers websites like WordPress.com, Pinterest, Zynga, CloudFlare, Hulu, Zappos, and Netflix[43].

Apart from offering of HTTP server abilities, Nginx also operates as a POP3/ IMAP proxy mail server and also functions as HTTP cache server and load balancer. Nginx can run on Mac, Linux, OS X, AIX, Solaris, BSD and HP-UX variants. Nginx does not offer current processes for individual web requests; rather, the number of worker processes to be produced for the major process of Nginx is configured by the administrator. A major thumb rule is having a worker process for every CPU. Each of these processes is one-threaded. Every employee can control many concurrent connections, even in thousands. Instead of utilizing multi-threaded programming, it performs this asynchronously with just a single thread.

Nginx also rotate cache manager and cache loader processes to decode information from a disk and then loads it into the cache and removes it from the cache as directed. It is made of modules that are required at an organized time, meaning that the source code can select whichever of the modules it wants to assemble. There exist modules for linking to the proxy server, load balancing, back-end application servers, and for several other functions. However, there is no module for coding PHP as Nginx can collate PHP code itself.

1.6.5 Syslog

Syslog is the method utilized by network devices for sending event messages to a logging server[45] – normally called the Syslog server. Its protocol is aided by a broad range of tools and may be utilized to log various kinds of events. For instance, a router may forward messages to users by logging on to the console sessions, and a web server may log access-counter events.

Many of the network tools, such as switches and routers, can be used for sending Syslog messages. Also, Unix servers may have the capability to create Syslog data, since many web servers such as Apache, some firewalls, Windows-based servers and even printers do not allow Syslog natively. However a very large amount of third-party equipment makes it very easy to receive IIS data or Windows Event Log and send it to the Syslog server.

Syslog allows the separation of software that creates messages, software that analyzes and reports them, and the system that keeps them. Every message is marked with a facility code depicting the software kind that generates the signal, and allocate a severity name.

The design of computer systems might utilize Syslog for the management of the system, auditing security, the provision of general information, debugging, and analysis of messages. A broad variety of devices, such as signal routers and printer receivers, across most platforms utilize the Syslog. This allows the consolidation of logging data from several kinds of systems in a basic repository. The Syslog Implementations exist for most operating systems. Table 1.1 defines different log levels, its severity and details.

Table 1.1 Syslog Log Level Description

Value	Severity	Keyword	Description	Examples
0	Emergency	Emerg	Unusable, Fatal Error. System Stop	This level is not for the application used.
1	Alert	Alert	The log should be looked properly	Lack of the major ISP connection.
2	Critical	Crit	Critical conditions	A failure in application of the system
3	Error	Err	Error conditions	The file storage limit has been reached and any attempts to write fail.
4	Warning	Warning	May show that an error will happen if the necessary action required is not taken.	Just 2GB is remaining for a non-root file system.
5	Notice	Notice	Unusual events, though not error conditions.	
6	Informational	Info	Normal information messages which need no action.	An application has begun, ended or stopped successfully.
7	Debug	Debug	Data very efficient to developers for application debugging.	

1.7 Conclusion

The various type of logs record the important factors that provide specific information about how they were generated, how they were processed, and also contains other relevant information. This chapter demonstrates why an open source real time log analysis system that can be customized with signatures to find and detect specific attacks is required for the organization’s needs.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This chapter provides a comprehensive view about the work done in field of log based IDS (Intrusion Detection System). U2R (User to Root) and R2L (Remote to Local) attacks and how IDS are used to detect and process that attacks are discussed in detail in this chapter. This chapter also provides comprehensive overview of commercial, free and opensource available SIEM and their integration with the operational envoirnment of other organization. Feature comparison with individual ability of SIEM for enhancement of system according to organizational need is also discussed.

2.2 Log Based IDS for Network / System Intrution Detection

With the advent of technology, especially the internet, activities such as hacking and identity theft have become the norm of the day. There has been escalations in incidents of suspicious activities that may have gone unnoticed. A firewall alone is not enough. Firewall functions by making traffic flow decision by inspecting the data packet headers. Unfortunately, the entire packet content is not inspected. Additionally, firewall, unlike intrusion detection system, cannot identify or stop a malicious code that is embedded within normal traffic making the computer system vulnerable to lethal attacks [12]. A sophisticated attack can, therefore, bypass a firewall undetected. On the other hand, IDS systems scrutinizes the entire content of each packet for possible signs of malicious activity. Besides, the technique of content scaning it also confers intense packet data analysis as compared to a firewall. The most sophisticated attacks are also detected by intrusion detection systems through the integration of familiar protocols such as HTTP.

As such, these events are detected using computer security log analysis by comparing them against known signatures or network behavior analysis. Signatures are set of rules that are used by an IDS to detect suspicious activities such as DoS attacks. For this to be effective real-time monitoring of the events is essential [8]. It enables the maintenance of an updated database system of attack signatures which can greatly enhance the probability of detection of an attack. For that matter, the logging and intrusion detection platforms can serve as a source of vital data that undergoes careful management through adequate filtering and analysis. The quality of data, data rules, and automation are key aspects of this whole process. Usually, there is a lag between the detection of a new threat and the signature for matching and identifying that threat being applied to a particular IDS. The effectiveness of the IDS can be compromised during such a scenario since it is unable to detect the new threat. However, with automation, the impact of the time lag can be bridged. The automation of the analysis process is possible through supporting an event correlation software coupled with centralized logging. This result in a unified platform that minimizes the challenge of monitoring and reporting of events in real-time [13].

Currently, there exist several types of IDS technologies in the contemporary technological world. This is mainly influenced by the existence of various network configurations. From NIDs that monitors all inbound and outbound traffic at all layers of the Open Systems Interconnection (OSI) model and continues to analyze it for any suspicious activity to Host Intrusion Detection Systems (HIDS) that monitor the inbound and outbound packets and alerts the administrator in case of suspicious activity[5]. In addition to analyzing network traffic, HIDS also examines certain system settings such as local security policy, local log audits, and software calls. However, of importance to this discussion is the signature based IDS. Typically, a signature based IDS just like an antivirus software detects malware, monitors the packets in the given network and compares them against the particulare database filled with signatures or patterns of known potential malicious threats. Besides, it has a memory to remember the attributes of the attack for future reference. Remote Dictionary Server (Redis) for instance, is an example of an open-source data storage structure that functions as a database. Over time a database of attack signatures is compiled from the previous intrusions.

For effective log management, the deployment of fully functional subsystems of an IDS system is necessary. Typically an intrusion detection system runs by three primary sub-subsystems, the Detection Engine, the Packet Decoder and the Logging and Alerting System [5]. Detection Engines are algorithmically functioning programs which are part of an IDS which support a collection of related signatures. They are specifically designed to inspect and define a definite set of values or rules that are allowable based on the protocol of a particular network system.

2.3 Generic Searching and Processing

Signature Detection Engines are capable of decision making through log analysis based on the available signature records. By processing, searching and integrating the use of generic search, Elastic Engine searches with the help from defined signatures, can detects the unwanted traffic. This helps the system scan through large amounts of network traffic for any activity that qualifies as an intrusion. It is the recording of these log of events that can be retrieved and matched against a pre-existing database for possible future attacks. Alternatively, the log info could be used as a source of data for policy decision making or damage control.

Normally, the search engine executes intrusion detection in a two-dimensional linked list of rules. As it is, the rules are organized into a series of chain headers and chain options. The rules are usually compressed into a list of common attributes with subsequent incorporation into a single chain header to enhance the detection process. The chain option entity stores the individual detection signatures. Through recursion, the rule chains are searched for each packet in a bi-directional manner. The first rule that matches a decoded packet in the detection engine triggers the specified action in the rule definitions and returns. The alerting and logging subsystems are selected during execution time [4].

Different attacks can be detected by an IDS, such as Probe Attacks, Denial of service (DoS) attacks, R2L (Remote to Local) attacks, and U2R (User to Root) attacks. Each of these attacks targets a specific area. Whereas Probe attacks are those that are aimed at acquiring information about the target network from an external source, DoS attacks function to force the target to stop the service they are executing by flooding it with probes and illegitimate request [7]. The R2L (Remote to Local) attacks that involve the network level and host level, this makes them one of the most difficult to identify. U2R (User to Root) attacks, on the other hand, are often content based and used to work against particular application. A record of the signatures of these threats is important for their detection. However, in the cases of varying known signatures, First Strike Assaults and Denial of Service attacks are a major setback in the identification of these attacks. The “Code Red” virus is an example of how new attacks can bypass a signature based IDS [9].

The Commercial projects related to R-ELK scope and working can be categorized into the followings

- Log Analysis Tools
- Commercial SIEM’s

2.4 Log Analysis Tools

The log analysis tools are used to analyze log files of different formats and produce results related in shape of graph or text according to the given data. These tools came with different features including indexining of data, evaluation of data, reporting in many formats and full text index search.

2.4.1 Grayling 2[47]

This is a totally incorporated log management which is an open source system that allows System Administrators to receive, index, and evaluate disorganized, systematic and framed data from any source systems that is available.

The logging system is extremely pluggable, and permits unify log management from diverse systems. It is incorporated with exterior components like; MongoDB for Elastic search and metadata utilized in keeping log files and used to permit text search.

Graylog 2 possesses the following properties:

- Prepared for enterprise level production
- Can function on data using any log source
- Entails an alerting system and a dashboard
- Permits actual-time log processing
- Very customizable and extensible
- Permits parsing of unstructured data
- Offers an operational data hub

2.4.2 Log Check[48]

Log check is a log management open source system that assists the System Administrators to identify the unfamiliar security violations and problems in log files automatically. It sends messages periodically relating to the result of the analysis to an e-mail address that has been configured.

Log check is made as a cronjob on an hourly interval and all system by default reboots. There are several log file levels where enhancement is done in following ways:

- Workstation: This is for sheltered systems and assists in filtering some of the messages. It also entails rules described under server and paranoid levels.

- Server: this is the filtering default level for checking the log, and its rules are described for several daemons of the system. The rules lay down under the paranoid level are also entailed at this level.
- Paranoid: is intended for high-security systems that are running very few services as possible.

Log check can also be used for sorting out messages which are to be given in three probable layers, and this entails network attack alerts system events and security events. The level of details for reporting system events can be chosen by the System Administrator. However, this depends on the level of filtering although this has no effect on the network attack alerts and security events.

2.4.3 Log Watch[49]

Log watch is a UNIX/ Linux system based log analyzer and reporter which may be customized easily. Also, it allows the addition of extra plug-in and also alows the creation of custom scripts (for particular logging requirements) by the System Administrator.

All it does is reviewing of system log files for a given time frame and then generates on the system areas a report based on receiving of data. One attribute of this logging system is the fact that it can be used easily by a new System Administrator and also function on most UNIX systems and many available Linux distribution.

2.5 SIEM and Their Feature Analysis

2.5.1 Alient Vault[50]

Unified security management is a comprehensive platform that was designed and formatted to ensure the mid-market organization can defend and protect themselves against a modern threat that is evolving. Unified security management reduces greatly the complexity and time required by the user to install and gain insight of the programs within an hour. The prioritization of risk through comparative analysis of threat severity, reputation and asset vulnerability is done by AllienVault.

Major Properties:

- Unified Security Management (USM): There exists five essential security software in a single solution.

- Discovery of Asset: This feature has a built-in software inventory infrastructure, asset inventory and active and passive network asset discovery.

Key Features:

- Vulnerability Assessment: This feature enables organizations to scan assets to identify vulnerabilities that can be exploited by a bad actor.
- Intrusion Detection: A vital part of AlienVault's USM platform is to monitor the network and assets for threats with Network IDS, Host IDS, File Integrity Monitoring, Registry Monitoring, and Rootkit Detection capability.
- Behavioral Monitoring: It has built-in log management, service obtainability monitoring, net flow analysis, and network packet capture.
- Security Intelligence: This feature allows for correlation of data produced by the built-in tools and external data sources, incident response, and reporting to support threat detection and compliance use cases.
- Integrated Threat Intelligence: The ability of the USM to identify the new threat, which cumulates into having a deeper understanding of the threat vector, the technique used by the attacker and providing effective and reliable defense system is driven by the Allen Vault Labs Threat Intelligence. The USM platform is updated for better performance by the AllenVault lab which regularly provides eight different coordinated set of rules to the USM platform.

The ability of Allen Vault to focus on easy to use and speed-to-deployment programs makes it the best choice for organization and company with lesser staff strength and limited security programs in term of cost and effectiveness.

2.5.2 QRadar[51]

The IBM Security's QRadar Platform can be used as a virtual appliance, an appliance, or as infrastructure as a service (IaaS). They have numerous functions such as hybrid option with the deployment of QRadar on-premises analysis on a SaaS solution which was hosted on IBM cloud, which help to perform monitoring remotely freely from their service operation centers. IBM product provide a main platform for integrating log management, event management, anomaly detection, incident forensics, vulnerability, configuration management and security information

Main Features:

- Visibility: It provides a comprehensive surveillance covering the entire IT infrastructure and fast-tracked threat detection capabilities. QRadar can monitor mischievous activity over a long period and reveals advanced threats. The use of QRadarVFlow appliances helps greatly in security monitoring of business activities analysis and detection of an anomaly.
- Prioritization and Reduction of Alerts: Targeted and thorough investigation on the list of suspected incident that are actionable helps in reducing the numerous events into controllable and handy list
- Management of Threat: Data access and detailed user activities log report helps to detect insider fraud with various advanced options in use.
- Activity Log Reports: The integration of network threat protection programs and log management in the same database interface has helped in managing the compliance excellently and efficiently.
- Master Console: IBM QRadar provides a cost effective security intelligence solutions to support managed service providers.

IBM's QRadar is the best solution for large to the midsize enterprise with their requirement supported by SIEM and that enterprise whose cases required network flow, behavior analysis, and packet analysis.

2.5.3 Log Point[52]

The Log Point solution powered by SIEM can extract incidents and events from an existing log report in an IT organization of any size. It also can filter and make a correlation with the result which can be displayed in dashboards which can be in the configuration of responsibilities and specific roles of each user on the log report. Actionable, real-time and perception from the data retrieved from the machine helps to increase and improve the efficiency of the operation and strengthen the security position by compliance with streamline from the regulatory institution. Log Point helps expert to have insight into all events and incident across the organization or infrastructure.

Main properties are as under:-

- Scalability: The report has a centralized based, the analysis, and management enhances the process of easy access the event source.
- Transparent Search and Analysis: The log point gives the users the ability to carry out analysis on the whole infrastructure or organization.

- Big Data Storage: The log point can allow enterprises to make use of any storage system maximally which ensure prioritization of the usage and amount of NAS (critical storage systems) and any interconnection and protocol in the enterprise.
- The flexibility of platform: There are three different ways by which the Log Point can be delivered to meet the user's needs. These are: virtual (make use of current infrastructure which allows for easier and quicker scalability platform); self-contained software package (regarding positioning scenarios or current hardware in the organization permits for flexibility) and appliance (combined hardware and software package).
- Information and Data Generated: The Log Point enabled security devices and network, access, and identity management solutions, wide- enterprise ERP deployments systems and operations management suites.
- Enrichment of Data: The log point allows total data enrichment capabilities, permitting Log Point to create a message relating to an ERP system's critical transaction to carry out an investigation.
- Rapid Processing: There is instant processing of data before storage and achieving an actual-time evaluation of events in short distance.

The log point gives the platform for a small enterprise with a limited budget the opportunity to use SIEM solution in their operation capabilities. They also support large and complex enterprise with an operational solution. Their partnership across the world gives them the capabilities for growth even though they majorly operate in Europe.

2.5.4 Log Rhythm[53]

The log rhythm provides the interface where Log management, network forensic, file integrity monitoring and SIEM are unified in a security intelligence platform. SIEM solutions are majorly used to accommodate the small to large organization. SIEM solutions consist of many unified components which are Advance Intelligence Engine, Event Manager, Console and Log Manager. Log Rhythm combined forensics and management abilities, endpoint monitoring and SIEM to ease with deployment.

- A1 Engine: Log Rhythm achieves visibility by cross-examining all machine data and available log with scientific visibility at the network levels and endpoint. AI Engine helps in leveraging of data which help in analyzing. AI Machine Analytic technology performs real-time analysis of all activities that occurred with the environment. It also supports identification of undetected risks and threat

- Log Rhythm Labs: it also performs function that accelerates the rate of threat detection and responses and this consist of log parsing from over 700 operation systems, devices, and application.
- Detection: The Log Rhythm can discover custom malware attached to zero-day attacks and is formulated to access traditional security solutions which are created to identify known malicious behavior and specific signatures.
- Response: The logs Rhythm instinctively queue up a response or disable an account for validation until a more thorough forensic activity is performed.
- Monitoring of Network: The Log Rhythm's Network Monitor provides accessibility to egress points/ network ingress with Smart Flow data providing deep packet visibility into the application in use and each network session observed. This leveraging the extensive packet metadata delivered and confirm behavioral baselines across observed network activities,
- Smart Capture: The Log Rhythm has the capabilities to do Smart Capture automatically. It captures all packets that are connected with doubtful sessions for complete packet forensics.
- Log Rhythm principally caters to an organization that needs a combination of SIEM, endpoint monitoring, and value ease of function abilities and deployment.

2.5.5 Splunk[54]

Splunk provide the platform for pre-packaged reports, analytic, investigations, correlations to identify, dashboard and respond to the external and internal threat. Splunk also employs querying of language which supports visualization with over 100 statistical commands. It is also possible for Splunk to provide extra supports for major security data source such as asset management system, payload, and malware analysis, wire and network data, asset management and identity system and threat intelligence for speeding up the rate of adoption and deployment.

Main features include:-

- Security and Report Metrics: Splunk provides the platform to leverage many of out-of-the-box reports, metrics, and dashboards. Splunk provides the avenue to transform raw data or tables or graphic into analytic, and the data can be saved into PDF and CSV.

- Classification and Incident Review: Splunk App is used in governance, protection against tampering and in auditing for Security reports on every system and users activities for a comprehensive audit trailing. This support for bulk event rearrangement changes criticality classification and in status, with all the analysis activity procedure available for auditing operation.
- The response, Security Analytics, and Correlation: Splunk enhances responses, containment, prioritization, remediation process and security monitoring by evaluating machine data to comprehend the impact of signals or events.
- Sources of Threat-Intelligence: The sources of threat intelligence are law enforcement, internal and shared data, third-party subscriptions, free threat-intelligence feeds and FS-ISAC Soltra (via STIX/TAXII)[56].

Framework for Threat-Intelligence: Splunk framework allow numerous sources of threat feeds, including a subscription-based feed like a TCP streaming and open-source feeds like a flat files via an API service; Splunk also feeds from law enforcement agent or local environment like that of manual download and share threat feeds like that of Open IOC[57] document by TAXII protocol or STIX

Splunk has acquired Caspida startup[58] and also added a machine that has behavioral analytics to detect threat easily. Many companies are requesting for SIEM customization to support analytic function and log format which will be a great asset to them especially in security and IT support operation.

2.5.6 Tenable[55]

SIEM Tenable leverages the log management abilities of the Correlation of the Log Engine (LCE) to get all the software activity, records, network traffic, and user events. IT evaluates data for events correlation and effect on compliance and security posture. Threat-list intelligence and event context related to any system is offered by Tenable Nessus configuration and vulnerability scans and actual-time surveillance with the Tenable PVS (Passive Vulnerability Scanner).

Key features include:-

- Log Normalization: It normalize, correspond, and evaluates network and user activity from log data derived by any application or device in a central portal across your enterprise
- Event Correlation: Multiple forms of correlation of available event, comprising of associating IDS event, statistical anomalies with vulnerabilities.
- User Monitoring: Tenable monitors associated events like; IDS detection, Net Flow, file access, firewall log activity, login failure or system error, and user activity with particular users for insider threat detection and easy reporting.
- Net Flow Analysis: In every Tenable LCE instances, there exist agents for several platform technologies that may accept Net Flow traffic logs from switches, routers, and other network tools.
- Full Log Search and Indexing: All logs are constricted and kept, and by utilizing full-text search, you can make a search of logs for IP addresses, usernames, keywords, etc. Log searches are kept with a separate checksum which at any time can be re-launched.
- Malware Detection: All the processes running for malware on Windows machines are monitored by the LCE Tenable Windows client, and when malware is detected, it may alert the security team.
- Network Content Analysis: Network traffic in real-time with Tenable PVS can be analyzed by Tenable and can offer a precise vulnerability report and a forensic occured-time log of events system like; social network activity, DNS lookups, and shared files.

2.6 Analysis

All the SIEM solutions which can collect, index and detect attacks are commercial solutions with propriety code. Customizations are only available in provided options and interfaces by the selected solution. The table 2.1 provides a comparison of features among these solutions.

Table 2.1 Comparison between Commercial SIEM

Feature	Splunk	QRadar	Tenable	LogPoint
OpenSource	No	No	No	No
RealTime / Near Realime	Yes	Yes	Yes	Yes
Log Collection Agents	Yes	Yes	Yes	Yes
Customisation	Limited	Limited	Limited	Limited
SourceCode Review	No	No	No	No
DataSharing Policy	Yes	Yes	Yes	Yes

2.7 Conclusion

In conclusion, though firewalls, antiviruses and router-based packet filtering are essential entities of a general network security environment, they are inefficient on their own. It is, therefore, necessary to effectively manage these security systems such that they complement each other in the efforts to thwart any malicious attacks. IDS tools are being integrated to inside and outside firewalls and are quickly becoming the gold standard in the maintenance of secure network systems. Commercial SIEM's are made for commercial purposes and not provided specific purposes for the organization. Usually commercially available SIEM and log analysis solutions that provide collection and searching applications target a large audience. So difficult to be managed, mould in specific organizational needs. Specific organizational needs some processes to be moulded and managed accordingly. While Opensource solutions in this market are largely frameworks and provides a access methods to be built upon them. Some tools that have been built upon them, their framework cannot be integrated into operations without further development. So the need for log search and collect and a customized without limitations search app like ElasticSearch is needed.

Chapter 3

DESIGN AND METHDOLOGY

3.1 Introduction

In our data-driven world, there are many challenges associated with log analysis. Companies produce a lot of data, yet there is no easy way to log and analyze this data. The data which is generated every minute is used to analyze business and develop appropriate business strategies. There should be mechanisms to enable easy logging and analysis of such huge amounts more effortlessly.

The ELK stack is a complete, integrated logging platform built on a combination of three open-source technologies; Elastic search, Logstash, and Kibana. Design is based on open source experiment and being done in collection of logs. The ELK stack attempts to address problems in log analysis, including the existence of non-consistent log formats and decentralized logs. Also, expert system functions can be built into the logging systems [16]

3.2 R-ELK Stack

The R-ELK stack is a complete, integrated logging platform built on a combination of three open-source technologies; Redis, Elasticsearch, Logstash, and Kibana. The ELK stack attempts to address problems in log analysis, including the existence of non-consistent log formats and decentralized logs.

The R-ELK open-source stack includes:

- Redis: Redis is used to store and manage buffer to handle large amount of data[23]
- Elasticsearch: This is used for deep search and data analytics.
- Logstash: This is used for managing centralized logging, which includes shipping of logs and forwarding them from multiple servers, log parsing, and log enrichment
- Kibana: This provides for powerful data visualizations.

The application ELK stack enables various use cases, for example, free and structured search, use for data analytics, log and event analysis, as well as visualization and visual exploration using Kibana [18]

3.3 System Description and Components

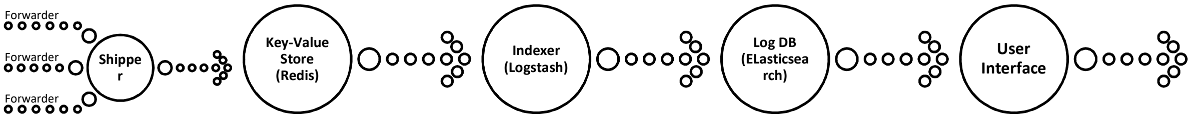


Figure 3.1: A typical ELK stack data pipeline.

In a typical ELK Stack pipeline, logs from multiple application servers are shipped through the Logstash shipper to a central Logstash indexer. The Logstash indexer outputs data to an Elasticsearch cluster. This can be queried by Kibana to display visualizations and build dashboards on the log data.

3.3.1 Shipper – Logstash Forwarder / FileBeat

Logstash forwarder and/or File beats is an open source file content fetcher, usually used to get logs files and send them to logstash and are installed on the Client Server and used to forward logs from the installed system to the main server. These softwares supports self signed certificate for authentication and encryption. It supports lumberjack output formats.

3.3.2 Broker – Redis

Redis is a data structure server. This means that it provides a set of commands to access the variable data structure, which is sent using a client-server model with TCP sockets and a simple protocol. Therefore, individual processes can get, fetch and update required data whether it is same structure in a colloborative style.

The implementation of the data structure has some special properties for Redis[19]:

- Redis has also the option to store on disk whether configured to use both disk and server memory. This means that Redis is fast but not nonvolatile.

- Data structure stress on the realization of memory efficiency, so the data structure inside redis will manage to work in less memory than the same data structure, using a high level programming language modeling.
- Redis offers many features such as copy, adjustable level of durability, clustering, high availability.

3.3.3 Indexer – LogStash

Logstash on the data pipeline helps to collect, parse, and analyze a variety of structured and unstructured data as well as events which are generated from systems. Logstash provides plugins that enable connecting to various input sources and platforms. It is designed to process logs efficiently, to process events, and to for distributing unstructured data sources into various outputs using output plugins such as file, stdout or Elastic search. Logstash has the following features [16]:

- Centralized data processing: Logstash helps to build a data pipeline that centralizes processing of data. By using various plugins for input or output, log stash converts various input sources to a common format
- Custom log formats support: Logs from different applications vary in format from application to application. Logstash helps to parse and to process custom formats in scale. Logstash provides support for writing unique filters for tokenization and provides easily usable filters.
- Plugin development: Custom Logstash plugins can be developed and published. Already, a large variety of custom plugins is available

3.3.4 Searching – ElasticSearch

Elasticsearch is a distributed highly scaleable and flexible ALv2 licensed (apache opensource) search engine that is multitenant-capable and was released in 2010 by Shay Banon [18]. Elasticsearch is a project that is fairly new. It is built on top of Lucene which is quite a mature Java-based search and indexing technology that is open-source. Elasticsearch runs in a Java application server. However, applications do not have to be written in Java to work with Elasticsearch. This is because it can send and receive data over HTTP in JSON to search, to index and to manage an Elasticsearch cluster.

Elasticsearch is best for applications which are built to handle real-time or similar data that has to be processed and analyzed fast such as analytics applications. Many internet businesses and organizations have been using Elasticsearch. These include Netflix, GitHub, Stack Overflow, and Facebook. These use Elasticsearch to handle requirements of agile data processing and data storage. [14].

The main focus of Elasticsearch is to provide powerful and fast search. It also aims to explicitly address the issues of availability and scalability. Elasticsearch is also built for Big Data search as well as a performance which relational databases were not designed to support. Elasticsearch provides structured search, aggregations, highlighting of hit word and more. These features enable developers and users to extract information that is valuable from their data without regarding the form.

The main features of Elasticsearch include[16]:

- It is open source distributed, highly available and scalable as well as a real-time document store.
- Elasticsearch provides real-time capabilities for search and analysis
- Elasticsearch provides a RESTful API to do lookup, and various other features including geolocation, multilingual search, autocomplete contextual suggestions, and snippets of results

Elasticsearch is horizontally scalable and provides easy integration with cloud infrastructure.

3.3.5 Visualize – Kibana

Kibana is an open source platform for data visualization platform that is Apache 2.0 licensed. It helps to visualize all kinds of structured and unstructured data that has been stored using Elasticsearch indexes. Kibana is written in HTML and JavaScript. Kibana uses the powerful indexing and search features of Elasticsearch which are exposed through its RESTful API to present powerful graphics to users [11]. Kibana exposes data using beautiful graphs, geo maps, pie charts, histograms, tables and more. Kibana simplifies understanding large data volumes. A simple browser interface enables quick creation and sharing of dynamic dashboards which display queries real time.

Some of the key features of Kibana are as follows:-

- Kibana provides a flexible visualization and analytics platform for use in business intelligence.
- Real-time analysis, charting, summaries and debugging

- An intuitive, user-friendly interface. The user interface is highly customizable through drag and drop when needed
- Allows saving a dashboard, as well as managing several dashboards
- Dashboards are easily shared and embedded within other systems

3.3.6 SIEMIDS

SIEMIDS is a self-developed system in PHP / CSS /HTML /js to interact and act as dashboard for the R-ELK System.

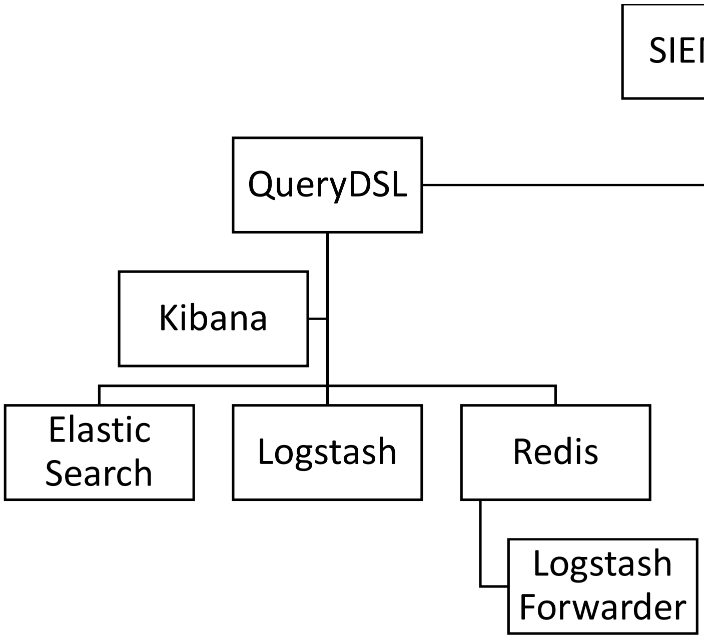
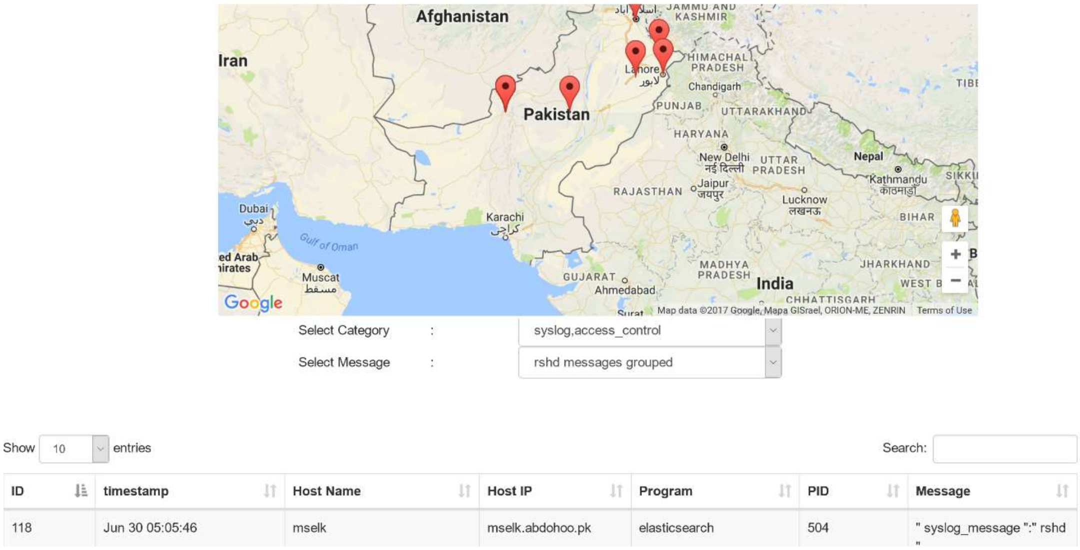


Figure 3.2 SIEMIDS Integration with R-ELK System

SIEMIDS reterives data from the system, data is further categorized into syslog and web signatures on a php page. The ELK application protocol interface compatible signature is loaded from the database and applied/searched via QueryDSL. The result is processed in SIEMIDS and displayed on map and also in table format as shown in figure3.2.



Select Category : syslog,access_control

Select Message : rshd messages grouped

Show 10 entries

Search:

ID	timestamp	Host Name	Host IP	Program	PID	Message
118	Jun 30 05:05:46	mselk	mselk.abdohoo.pk	elasticsearch	504	" syslog_message ":" rshd

Figure 3.3: SIEMIDS Simple Dashboard

3.4 Attack Signature Framework

An attack signature is a exclusive procedure of information that can be used to detect an attacker's attempt to exploit a known operating system or application weakness. The developed signatures work on SIEMIDS and are generated by Signature Conversion Tool. This framework for attack signatures require following to be described in signature

In XML OSSEC format[37]:

1. Validated XML signature format
2. Nested Signature XML format not supported in Conversion tool

In plain text/Regex Format

- Attack Signature regular expressions or simple text

Attack Signature framework is implemented in Signature Conversion Framework

3.4.1 QueryDSL

Elasticsearch supports json based QueryDSL language that is used to put and get information in json based queries. Abstract syntax tree of json based queries , consisting on two types of clauses:

- Leaf query clauses

Leaf query clauses look for a specific value in a specific field, it used terms like match, term or range. These queries can be taken into action by themselves.

- Compound query clauses

Compound query clauses consists on wrapping of other complex queries and are used to associate multiple queries in a logical way (like the bool or dis_max query), or to change their behaviour (like constant_score query).

3.4.2 Signature Conversion Framework / Tool

The signature conversion tool for R-ELK stack converts the strings, and regular expressions (that are used to detect attacks in logs converts) to the signature used in QueryDSL for Elastic Search.This signature Development framework mostly uses leaf query clause of elastic search and converts the signature through the following features. Attack Signature are categorized into two main types, and their signature generation process is described as below:-

1. In Ossec XML Format
    a. Attacks signature in standard XML format is given by the user
    b. Attack Framework parses the user input and checks the user input is validated
    c. Attack framework converts each XML nodes into objects for the languages, and each object is calibrated and embedded according to QueryDSL Signature
    d. Prepared ready to use Syntax for Signature is displayed on the screen.
2. In Plain Text

In the case of Regular expression or string, the entered regular expression is embedded into QueryDSL specified string.

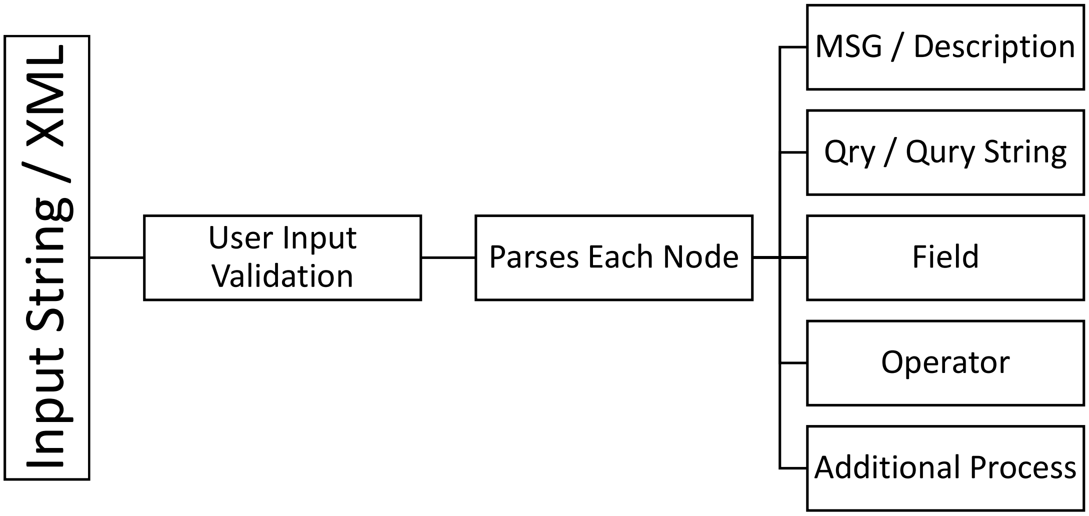


Figure 3.4 Singnature Conversion Parsing

3.5 Conclusion

This chapter gives comprehensive details about how different opensource and self developed components can be integrated with each others to form a comprehensive, enterprise level ready system that can be used, molded accordingly to the organizational needs to detect attacks and can act as analyser of logs. Attack signature generation and conversion tool sets to generate logs for the system and embedding it in order to use it.

Chapter 4

IMPLEMENTATION AND TESTING

4.1 Introduction

The chapter summarize implementation steps that are taken in order to completely build the system. In view of implementation, this chapter is about the development envoirnment, implementation requirements, languages and systems used in implementation, virtual private servers hired in different data centers and its real world implementation along with different testing scanarios that includes unit testing and acceptance testing related to implementation of the system.

4.2 General Storage Requirements

The general storage requirements depend on many features in case of R-ELK stack. However, table 4.1 describes how much data /storage is required in different scenarios.

Table 4.1 Storage Requirements according to Fields and Indexes

Test number	string fields	_all	doc_values	index size (in bytes)	Expansion ratio (index size / raw size)
1	Analyzed	enabled	enabled	7.6E+07	1.118
2	Not analyzed	enable	enable	7.6E+07	1.118
3	Analyzed	disabled	enabled	5.9E+07	0.869
4	not_analyzed	disabled	enabled	4.8E+07	0.709
5	not_analyzed, except (‘agent' field is analyzed)	disabled	enabled	5.1E+07	0.754
6					
7	analyzed	enabled	disabled	6.6E+07	0.97
8	Not Analyzed	Enable	Disable	6.6E+07	0.97
9	Analyzed	disabled	disabled	4.9E+07	0.721
10	not_analyzed	disabled	disabled	3.7E+07	0.553
11	not_analyzed(agent field analyzed)	disabled	disabled	4.1E+07	0.613

4.3 Preliminary Development

4.3.1 Storage Schema and Result Source

Elastic Search Storage schema terms are likes relation of traditional database schema, but does not have integrity checks and its traditional NO-SQL db.

The following gives overview about table terms used in elastic search.

Table 4.2 Elastic search in comparisonwith relational database

Relation Database	Elastic search
Database	Index
Table	Type
Row	Document
Column	Fields
Schema	Mapping

The Complete mapping with fields and type is defined in the table 4.4.

Table 4.3 Elastic Search Fields, Field Types and their usages

Field	Field type	Remarks
syslog_program	string	The program which generated log
received_from.raw	string	The full host name from where log is received in raw
syslog_hostname.raw	string	The hostname from where log is received
received_at	date	The date/time of l entry receiving time
_source	_source	The main json object from which contains multiple information
syslog_message.raw	string	The syslog / original log message in raw
type	string	Type of received log

Field	Field type	Remarks
@version	string	Version indication of received log version usually 1
received_from	string	The system from which log is received usually hostname
syslog_program.raw	string	Log generated from program like ssh
file	string	From which log entry is read at client server
host.raw	string	Host from log is received
syslog_pid.raw	string	The pid – program id at received from system in raw
syslog_pid	string	The pid – program id at received from system in raw
offset.raw	string	The offset at log file – raw
offset	string	The offset at log file
file.raw	string	File path – raw
host	string	
syslog_severity	string	Priority defined by the program or system config at client server
syslog_severity_code	number	
syslog_hostname	string	
_index	string	
type.raw	string	
message	string	Original message of the log entry
@timestamp	date	Time stamp
syslog_facility_code	number	
syslog_facility	string	

Field	Field type	Remarks
syslog_timestamp	string	
syslog_message	string	
syslog_timestamp.raw	string	
syslog_facility.raw	string	
syslog_severity.raw	string	Priority set by the host program
geoip.location	geo_point	Geo ip location if defined
_id	string	Log id
_type	string	Log type
_score	number	Usually 1 if not defined

Each field results is stored in elastic search index Schema, when viewing individually will be displayed in following JSON structure.

{
"_index": "logstash-2017.02.26",

- "_type": "syslog",
- "_id": "AVqDM-YPmxWrqVoGtd4u",
- "_version": 1,
- "_score": 1,
- "_source": {
    - "message": "Feb 26 17:07:46 mselk sshd[11772]: Failed password for root from 202.109.143.115 port 2661 ssh2",
    - "@version": "1",
    - "@timestamp": "2017-02-26T17:07:59.000Z",
    - "type": "syslog",
    - "file": "/var/log/auth.log",
    - "host": "mselk.abdohoo.pk",
    - "offset": "1288477",
    - "syslog_timestamp": "Feb 26 17:07:46",
    - "syslog_hostname": "mselk",
    - "syslog_program": "sshd",
    - "syslog_pid": "11772",
    - "syslog_message": "Failed password for root from 202.109.143.115 port 2661 ssh2",

- "received_at": "2017-02-28T05:31:39.933Z",
- "received_from": "mselk.abdohoo.pk",
- "syslog_severity_code": 5,
- "syslog_facility_code": 1,
- "syslog_facility": "user-level",
- "syslog_severity": "notice"

} }

4.4 Choice of Infrastructure and Programming Languages

4.4.1 PHP – Hypertext PreProcessor

PHP stands for Hypertext Preprocessor and is a server-side web based programming language.

PHP in R-ELK Stack for implementation signature works as a primary language in both of following systems.

- Signature Conversion Tool
- Signature implementation framework.

Selection of PHP was made due to its following features

- PHP can be bundled with variety of databases and can interact easily with may databases like MySQL, MariaDB, Oracle and others. PHP databases MySql and its variants are open source and is campatible with many servers like apache, nginx and others. PHP is available Windows, Linux and UNIX systems.
- Due to all these languages being free it is cheap and easy to setup and creates a system/script using PHP.
- There is valuable study material available on the internet for PHP, because of this it is very comfortable to learn PHP and find a lot of support on different resources in the internet.
- PHP is very cost effective and most of its software are opensource and free therefore most of IT experts use this technology and it is easy to find a professional who can set up the organization's structure on the bases of PHP infrastructure.

4.4.2 QueryDSL

Querydsl is a huge software pattern developed for Java language. Its programming instructions are just the structred querry langauge. QueryDSL is the main programming language that is used in Elastic Serach. Following are the key features of QueryDSL:-
It is has the feature of suggesting code wether it be the methods or variables.

- QueryDSL is strictly typed programming language and the programmer has to follow its syntax other wise the instructions will not work.
- The Domain Objects and the Properties Objects are used in QueryDSL and can be referenced.
- The Domain Objects can be updated and manipulated easily.
- Any sort of processing queries can be easily used.

4.4.3 RegularExpressions – REGEX

Regular expressions are an important part of any system that needs pattern matching, and advanced find functions. Moreover, regular expressions are supported by many platforms and can easily interact and integrate between many different languages. Along with these features, regular expression also have following features:-

- Regular expressions are portable
- Regular expressions help to write short code
- Regular expressions save time

Regular expressions can match just about anything

4.4.4 Apache Web Server

Apache Web Server is the most used web server and it live time is very vast IT industry is using this server since 1995. Nowadays more than 50 percent of web applications are hosted on Apache Web Server. It is very popular because of following distinguished features:-

- It is an old software therefore a lot of bugs have been fixed because of this it is very stable and crashes are very rare in it.
- As soo many IT experts are using this server therefore there is a vast amount of help and study material that as been published on internet about this server.
- It is an opensource and free of cost software
- All of the major operating systems including Microsoft Windows, Mac OS and Linux distros support this web server.
- Apache Web Server is continuously supported by its developers and all of its bugs are fixed as soon as possible.
- Although it is opensource and free yet it provides so much configurable options to meet all the needs of an organization.

4.5 Implementtion Details

The whole R-ELK system is implemented and integrated with three systems separately on internet infrastructure using public IP.

The system implementation is divided into following category:

Phase 1: Main R-ELK Stack Components are placed and integrated into each other in one server

Phase 2: Log Stash forwarder with valid cert installed on the system where system log is to be analyzed

Phase 3: Signature conversion tool and SIEMIDS is Developed.

Phase 4: Signature made thorough development is stored in SIEMIDS MySQL Database development to use.

Phase5: Data Collections is made through launching a simulated attack on the server, All Logs were transferred to the R-ELK server.

Phase 6: Data Result is collected and showed success and limitations.

4.5.1 SIEMIDS Development

SIEMIDS act as middleware between user and QueryDSL. The system is developed with the following languages

- PHP
- JavaScript / js / jQuery
- HTML / CSS
- QueryDSL API functions
- Maps

4.5.2 Embedded Signatures

The signature generated through Attack signature framework are embedded in PHP framework that is being utilized by SIEMIDS are to be embedded in PHP signature files. The system SIEMIDS loads all the function being used when the query is performed; it is loaded via sending a request to elastic search QueryDSL.

4.5.3 Attack Detection

Attack detection process works on the signature thorough the generated signature. It follows following steps

- Logstash forwarder sent data to the log stash which in indexed and stored accordingly after receiving Redis broker
- Attack signatures, generated attack signature framework are applied via PHP framework for elastic search Query DSL[33][34]
- SIEMIDS outputs the results.

Figure 4.1 outlines the complete process in detail.

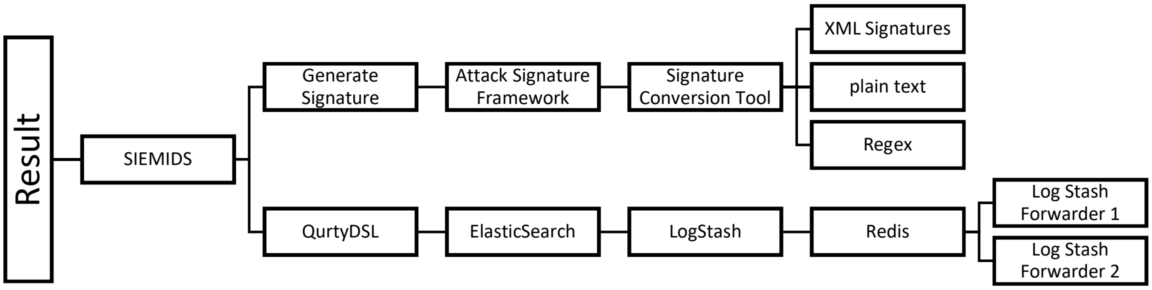
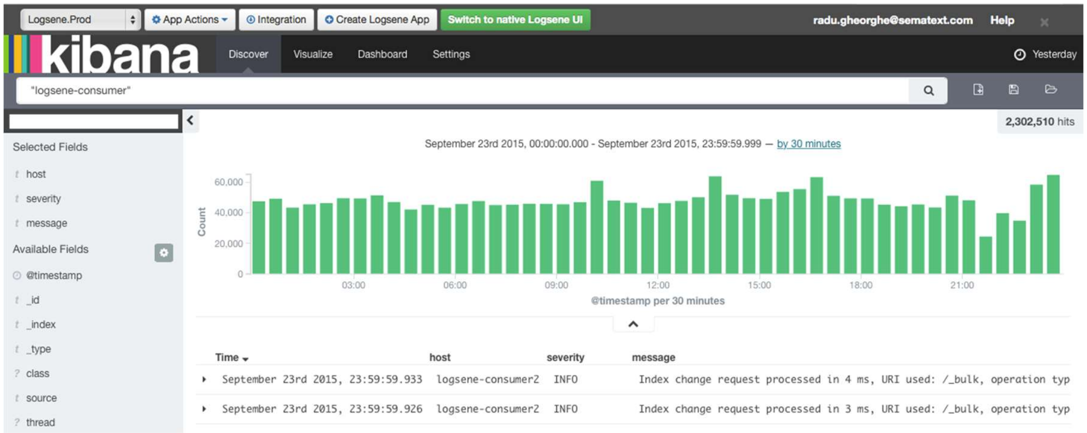


Figure 4.1 Complete System Integration using in Attack Detection

4.5.4 Kibana

The customized interface is built upon the the open source default interface "Kibana" which has built in Web server to serve the result pages and view index.



```
Time                           host                severity   message
September 23rd 2015, 23:59:59.933   logsene-consumer2   INFO   Index change request processed in 4 ms, URI used: /_bulk, operation typ
September 23rd 2015, 23:59:59.926   logsene-consumer2   INFO   Index change request processed in 3 ms, URI used: /_bulk, operation typ
```

Figure 4.2 Kibana Dashboard

4.5.5 Realworld Implementation

In realworld implementation, Servers – VPS were deployed in different parts of the world with different operating system in order to test sytem in its true functionality and properties like flexability and accessability.

The figure 4.3 express how it is implemented in realworld scenario.

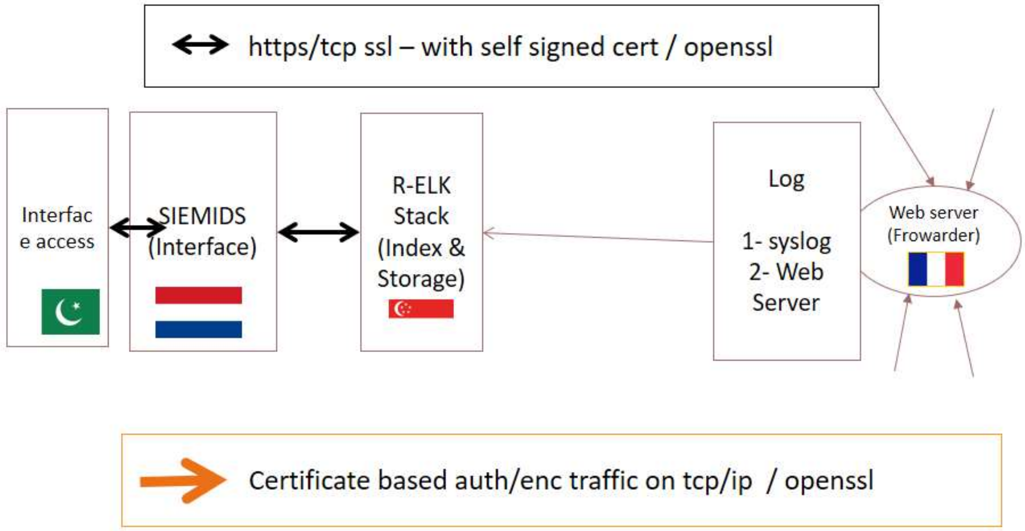

https/tcp ssl – with self signed cert / openssl
Interfac e access
SIEMIDS (Interface)
R-ELK Stack (Index & Storage)
Log
1- syslog
2- Web Server
Web server (Frowarder)
Certificate based auth/enc traffic on tcp/ip / openssl

Figure 4.3 Realworld Implementation Flow

The servers given in table 4.4 were deployed accordingly in final implementation scenario

Table 4.4 VPS Details for System Implementation

Sr.	Server Ip	Hostname	OS	Location	App
1	185.92.223.140	msids	CentOS 7 x64	Amsterdam	SIEMIDS
2	104.238.189.246	mslamp	CentOS 6.8 x64	Paris	Logstash Forwarder/ PHP Application
3	45.32.109.224	mselk	Debian 8 x64	Singapore	R-ELK + Kibana
5	45.32.163.155	mssigntool	Ubuntu Server	Miami	Signature Tool

Logstash forwarder located in Paris, France is a client server with DVWA php vulnerable application, here logstash forwarder is configured with grok and filters in such a way that it reports syslog and web application attack log to R-ELK server located in Singapore. The SIEMIDS sends queries with IDS signature to detect attacks and log analysis for available signature. These signature generated from Server#5 (table 4.4) locally hosted server and embedded in SIEMIDS. SIEMIDS is access via browser to show its result with map. Server#4 (table 4.4) Server location in Miami used to test and attack specifically at mslamp server in order to test signature. The Signature tested are given in Appendix . A

4.6 Functionality Testing and Testing Levels

4.6.1 Test Objective

Test objective is to ensure that system components are performing and giving result as they should be and designed to do. The whole testing phase start from the creation of test envoirnemt till the ensured and quality assurance on the system and integrity of the system components.

4.6.2 Testing Levels

The following testing was performed on the application

- Unit Testing
- Integration testing
- System testing
- Regression testing
- Acceptance testing

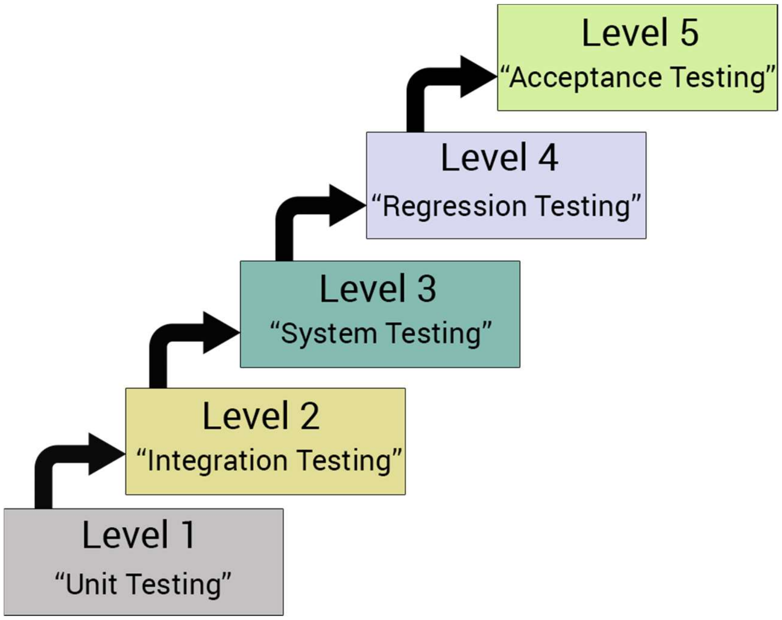


Figure 4.4 Testing Level used while functionality testing

Unit Testing

Unit testing is the first level of testing and focuses on testing small units of individual codes. Unit testing is always conducted before integration testing. Unit testing is done by using different testing frameworks and tools.

Integration Testing

Integration testing is one of the crucial testing levels. This testing is conducted to test the combined parts of the application to check that they function correctly together. This test performed to ensure components are working after integration into the system combined.

System to Acceptance Testing

Different scenarios were tested from system testing in order to verify and validate system implementation, regression testing was used in limited fashion as system was implemented in a virtual environment and in the last acceptance testing was done in order to test system against given requirements.

In all the system, white box testing method was used internally. Blackbox testing was done externally.

4.7 Security and Safeguards

The Following steps were introduced into the system in order to ensure the system security and safety

1. All VPS used to setup system used Key based ssh login
2. Log data shared with ELK Server near real time encrypted with openssl self signed certificate
3. IP Binding for usage of API dashboard
4. System implemented across the globe via VPS presented in different geographical locations on different IP address space.

4.8 Conclusion

The R-ELK – SIEMIDS/Kibana system was conceived and implementation details was explored in this chapter. The system implemented in different components and aligned in different virtual systems located across the globe. Security verification and validation was system implement and controls were places accordingly.

Chapter 5

CONCLUSION AND FUTURE DIRECTION

This chapter includes conclusions from dissertation, The possible future work and related enhancements of the work also have been specified.

5.1 Conclusions

Logs of any system of any level provides significant level of information about the process. Controlling, analyzing, searching and collecting logs in one place has always been a difficult and expensive task both in terms of computer resources and financially. This dissertation not only provides a framework by using open source process and applications for the complete life cycle from collecting logs till analyzing but also an analyzing system is built on that open source components in order to further detect attacks, attack signature framework for the built system and a dashboard for the results with an application protocol interface for further enhancements of the system.

During the research of this dissertation , following contributions have been gained

- Logs collected from different system available in different regions in different continent with different IP space.
- Logs from different systems were transferred to the main system via encrypted system
- An API for further enhancement and development of the system
- A complete customized IDS system applicable to detect attacks and display basic information

5.2 Limitations and Future Work

System has some limitation that effect the system performance as IDS. Some aspects of the signature framework and cluster process that can be enhanced is not implemented due to lack of time. Some of the more key features that can enhance system functionality and performance includes:-

- Algorithm to detect advance attacks and develop co-relation between different attacks
- Powerful application protocol interface development for external export function for other opensource and commercial projects and development of new applications
- Dynamic dashboard building option can improve reporting and easy to use
- SIEMIDS Dashboard if built on Kibana can also enhance Kibana and make default options to select and use system by utilizing in built Dashboard and API system

Appendix "A" – Attack Signature Codes

These attack signatures codes shows signature id Custom number with description about that specific error, group relates it to software or service from where this error will be generated while severity level given number shows its importance level

Signature ID	Description	Group	Severity / Level
31401	PHP Warning message.	apache	0
PHP Warning			
31402	PHP Fatal error.	apache	0
PHP Fatal error			
31403	PHP Parse error.	apache	0
PHP Parse error			
31404	PHP Warning message.	apache	0
PHP Warning			
31405	PHP Fatal error.	apache	0
PHP Fatal error			
31406	PHP Parse error.	apache	0
PHP Parse error			
31411	PHP web attack.	apache attack	6
expects parameter 1 to be string, array given in			
31412	PHP internal error (missing file).	apache	5

Failed opening|failed to open stream			
31421	PHP internal error (missing file or function).	apache	5
Failed opening required |Call to undefined function			
1001	File missing. Root access unrestricted.	syslog,errors	2
Couldn't open /etc/securetty			
1002	Unknown problem somewhere in the system.	syslog,errors	2
$BAD_WORDS			
1004	Syslogd exiting (logging stopped).	syslog,errors	5
exiting on signal			
1005	Syslogd restarted.	syslog,errors	5
restart			
1007	File system full.	syslog,errors low_diskspace	7
file system full|No space left on device			
1008	Process exiting (killed).	syslog,errors service_availability	5
killed by SIGTERM			
2101	Unable to mount the NFS share.	syslog,nfs	4
nfs: mount failure			
2102	Unable to mount the NFS directory.	syslog,nfs	4

<table>
<tr><td colspan="4">reason given by server: Permission denied</td></tr>
<tr><td>2103</td><td>Unable to mount the NFS directory.</td><td>syslog,nfs</td><td>2</td></tr>
<tr><td colspan="4">rpc.mountd: refused mount request from</td></tr>
<tr><td>2301</td><td>Excessive number connections to a service.</td><td>syslog,xinetd</td><td>10</td></tr>
<tr><td colspan="4">Deactivating service</td></tr>
<tr><td>2501</td><td>User authentication failure.</td><td>syslog,access_control,
authentication_failed</td><td>5</td></tr>
<tr><td colspan="4">FAILED LOGIN |authentication failure|Authentication failed for|invalid password for|LOGIN FAILURE|auth failure: |authentication error|authinternal failed|Failed to authorize|Wrong password given for|login failed|Auth: Login incorrect</td></tr>
<tr><td>2504</td><td>Illegal root login.</td><td>syslog,access_control
invalid_login</td><td>9</td></tr>
<tr><td colspan="4">ILLEGAL ROOT LOGIN|ROOT LOGIN REFUSED</td></tr>
<tr><td>2505</td><td>Physical root login.</td><td>syslog,access_control</td><td>3</td></tr>
<tr><td colspan="4">^ROOT LOGIN on</td></tr>
<tr><td>2506</td><td>Pop3 Authentication passed.</td><td>syslog,access_control</td><td>3</td></tr>
<tr><td colspan="4">^Authentication passed</td></tr>
<tr><td>2801</td><td>Smartd Started but not configured</td><td>syslog,smartd</td><td>0</td></tr>
<tr><td colspan="4">No configuration file /etc/smartd.conf found</td></tr>
<tr><td>2802</td><td>Smartd configuration problem</td><td>syslog,smartd</td><td>0</td></tr>
</table>

<table>
<tr><td colspan="4">Unable to register ATA device</td></tr>
<tr><td>2803</td><td>Device configured but not available to Smartd</td><td>syslog,smartd</td><td>0</td></tr>
<tr><td colspan="4">No such device or address</td></tr>
<tr><td>5101</td><td>Informative message from the kernel.</td><td>syslog,linuxkernel</td><td>0</td></tr>
<tr><td colspan="4">PCI: if you experience problems, try using option</td></tr>
<tr><td>5102</td><td>Informative message from the kernel</td><td>syslog,linuxkernel</td><td>0</td></tr>
<tr><td colspan="4">modprobe: Can't locate module sound</td></tr>
<tr><td>5103</td><td>Error message from the kernel. Ping of death attack.</td><td>syslog,linuxkernel</td><td>9</td></tr>
<tr><td colspan="4">Oversized packet received from</td></tr>
<tr><td>5105</td><td>Invalid request to /dev/fd0 (bug on the kernel).</td><td>syslog,linuxkernel</td><td>0</td></tr>
<tr><td colspan="4">end_request: I/O error, dev fd0, sector 0|
Buffer I/O error on device fd0, logical block 0</td></tr>
<tr><td>5107</td><td>NFS incompability between Linux and Solaris.</td><td>syslog,linuxkernel</td><td>0</td></tr>
<tr><td colspan="4">svc: bad direction</td></tr>
<tr><td>5108</td><td>System running out of memory. Availability of the system is in risk.</td><td>syslog,linuxkernel
service_availability</td><td>12</td></tr>
</table>

<table>
<tr><td colspan="4">Out of Memory:</td></tr>
<tr><td>5109</td><td>Kernel Input/Output error</td><td>syslog,linuxkernel</td><td>4</td></tr>
<tr><td colspan="4">I/O error: dev |end_request: I/O error, dev</td></tr>
<tr><td>5110</td><td>IRC misconfiguration</td><td>syslog,linuxkernel</td><td>4</td></tr>
<tr><td colspan="4">Forged DCC command from</td></tr>
<tr><td>5111</td><td>Kernel device error.</td><td>syslog,linuxkernel</td><td>0</td></tr>
<tr><td colspan="4">ipw2200: Firmware error detected.</td></tr>
<tr><td>5112</td><td>Kernel usbhid probe error (ignored).</td><td>syslog,linuxkernel</td><td>0</td></tr>
<tr><td colspan="4">usbhid: probe of</td></tr>
<tr><td>5113</td><td>System is shutting down.</td><td>syslog,linuxkernel system_shutdown</td><td>7</td></tr>
<tr><td colspan="4">Kernel log daemon terminating</td></tr>
<tr><td>5130</td><td>Monitor ADSL line is down.</td><td>syslog,linuxkernel</td><td>7</td></tr>
<tr><td colspan="4">ADSL line is down</td></tr>
<tr><td>5131</td><td>Monitor ADSL line is up.</td><td>syslog,linuxkernel</td><td>3</td></tr>
<tr><td colspan="4">ADSL line is up</td></tr>
<tr><td>5200</td><td>Ignoring hpiod for producing useless logs.</td><td>syslog,linuxkernel</td><td>0</td></tr>
<tr><td colspan="4">^hpiod: unable to ParDevice</td></tr>
<tr><td>2831</td><td>Wrong crond configuration</td><td>syslog,cron</td><td>0</td></tr>
</table>

<table>
<tr><td colspan="4">^unable to exec</td></tr>
<tr><td>2834</td><td>Crontab opened for editing.</td><td>syslog,cron</td><td>5</td></tr>
<tr><td colspan="4">BEGIN EDIT</td></tr>
<tr><td>2832</td><td>Crontab entry changed.</td><td>syslog,cron</td><td>5</td></tr>
<tr><td colspan="4">REPLACE</td></tr>
<tr><td>2833</td><td>Root's crontab entry changed.</td><td>syslog,cron</td><td>8</td></tr>
<tr><td colspan="4">^(root)</td></tr>
<tr><td>5301</td><td>User missed the password to change UID (user id).</td><td>syslog, su authentication_failed</td><td>5</td></tr>
<tr><td colspan="4">authentication failure; |failed|BAD su|^-| -</td></tr>
<tr><td>7101</td><td>Problems with the tripwire checking</td><td>syslog,tripwire</td><td>8</td></tr>
<tr><td colspan="4">Integrity Check failed: File could not</td></tr>
<tr><td>5901</td><td>New group added to the system</td><td>syslog,adduser</td><td>8</td></tr>
<tr><td colspan="4">^new group</td></tr>
<tr><td>5902</td><td>^new user|^new account added</td><td>syslog,adduser</td><td>8</td></tr>
<tr><td colspan="4">^new user|^new account added</td></tr>
<tr><td>5903</td><td>Group (or user) deleted from the system</td><td>syslog,adduser</td><td>2</td></tr>
<tr><td colspan="4">^delete user|^account deleted|^remove group</td></tr>
</table>

5904	Information from the user was changed	syslog,adduser	8
^changed user			
5401	Three failed attempts to run sudo	syslog,sudo	10
3 incorrect password attempts			
5402	Successful sudo to ROOT executed	syslog,sudo	3
; USER=root ; COMMAND=			
9102	PPTPD communication error	syslog,pptp	0
^tcflush failed: Bad file descriptor			
9201	Squid debug message	syslog,pptp	0
^ctx: enter level|^sslRead|^urlParse: Illegal | ^httpReadReply: Request not yet |^httpReadReply: Excess data			
2931	Yum logs.	syslog,yum	0
^Installed|^Updated|^Erased			
2932	New Yum package installed.	syslog,yum config_changed	7
^Installed			
2933	Yum package updated.	syslog,yum config_changed	7
^Updated			
2934	Yum package deleted.	syslog,yum config_changed	7

^Erased

Signature ID	Description	Group	Severity / Level
1001	File missing. Root access unrestricted.	Syslog	Error
^Couldn't open /etc/securetty			
1010	Process segfaulted.	Syslog	5
segfault at			
2502	User missed the password more than one time	Syslog/ACM	10
more authentication failures;|REPEATED login failures			
2504	Illegal root login	Syslog / acm	9
ILLEGAL ROOT LOGIN|ROOT LOGIN REFUSED			

Appendix “B” – QueryDSL API Function List with Access Methods

Category	Type	Function	Api Method
Document	Single	New	PUT

/index_name/name_type/1
{
  "my_field" : "my_value"
}			

Document	Single	Get Existing	GET
/index_name/name_type/0			
Document	Single	Delete	Delete
/index_name/name_type/0			
Document	Single	Update	PUT

/index_name/name_type/1
{
  …
}			

Document	Multi	Fetch	GET

GET /_mget
{
  "docs" : [
    {
      "_index" : “index_name”,
      "_type" : “name_type”,
      "_id" : "1"
    }
  ]
}
 	
GET /index_name/_mget
{
  "docs" : [
    {
      "_type" : “name_type”,
      "_id" : "1"
    }
  ]
}
 	
GET /index_name/name_type/_mget
{
  "docs" : [
    {
      "_id" : "1"
    }
  ]
}			

<table>
<tr><td>Document</td><td>Multi</td><td>Delete By Match Query</td><td>POST</td></tr>
<tr><td colspan="4">/index_name/_delete_by_query
{
  "query": {
    "match": {
      …
    }
  }
}</td></tr>
<tr><td>Document</td><td>Multi</td><td>Change By Query</td><td>POST</td></tr>
<tr><td colspan="4">/index_name/_update_by_query?conflicts=proceed
{
  "query": {
    "term": {
      "my_field": "my_value"
    }
  }
}

POST /my_index1,my_index2/my_type1,my_type2/_update_by_query</td></tr>
<tr><td>Document</td><td>Multi</td><td>ReIndex</td><td>POST</td></tr>
<tr><td colspan="4">/_reindex
{
  "source": {
    "index": "old_index"
  },
  "dest": {
    "index": "new_index"
  }
}</td></tr>
<tr><td>Search</td><td>URL</td><td>URL PARM</td><td>GET</td></tr>
<tr><td colspan="4">/index_name/name_type/_search?q=my_field:my_value
/index_name/name_type/_search
{
  "query" : {
    "term" : { "my_field" : "my_value" }
  }
}</td></tr>
<tr><td>Search</td><td>Shards</td><td>Get Shards/Indicies</td><td>GET</td></tr>
<tr><td colspan="4">/index_name/_search_shards</td></tr>
<tr><td>Search</td><td>Count</td><td>Count Query</td><td>GET</td></tr>
<tr><td colspan="4">/index_name/name_type/_count?q=my_field:my_value
/index_name/name_type/_count</td></tr>
</table>

{
  "query" : {
    "term" : { "my_field" : "my_value" }
  }
}			

Search	Validate	Validate Query	GET

/index_name/name_type/_validate?q=my_field:my_value
 	
/index_name/name_type/_validate
{
  "query" : {
    "term" : { "my_field" : "my_value" }
  }
}			

Search	Explain	Computaton Search feedback	GET

/index_name/name_type/0/_explain
{
  "query" : {
    "match" : { "message" : "elasticsearch" }
  }
}
/index_name/name_type/0/_explain?q=message:elasticsearch			

Search	Profile	Timing info by individual components	GET

/_search
{
  "profile": true,
  "query" : {
    …
  }
}			

Search	Field Stats	Statical Properties wthout Query	GET
/_field_stats?fields=my_field /index_name/_field_stats?fields=my_field /my_index1,my_index2/_field_stats?fields=my_field			
Indicies	Index	New	PUT
/my_index			

```
{
  "settings" : {
    …
  }
}
```

Indices	Index	Delete Existing	DELETE

```
 /my_index
/my_index1,my_index2
/my_index*
/_all
```

Indices	Index	Get Information	GET

```
/my_index
/my_index*
my_index/_settings,_mappings
```

Indices	Index	Check if exists	HEAD

```
/my_index
```

Indices	Index	Open/Close	POST

```
/index_name/_open
/index_name/_close
```

Indices	Index	Rollover	POST

```
/index_name/_rollover
{
  "conditions": {
    …
  }
}
```

Indices	Mapping	New To Existing Index	PUT

```
PUT /index_name/_mapping/new_type
{
  "properties": {
    "my_field": {
      "type": "text"
    }
  }
}
```

Indices	Mapping	Information By Field	GET

```
/index_name/_mapping/name_type/field/my_field
/my_index1,my_index2/_mapping/name_type/field/my_field
```

```
/_all/_mapping/my_type1,my_type2/field/my_field1,my_field2
/_all/_mapping/my_type1*/field/my_field*
```

Indices	Mapping	Check if Exists	HEAD

```
/index_name/_mapping/my_type
```

Indices	Alias	Creat on index	POST

```
/_aliases
{
  "actions" : [
    { "add" :
      { "index" : “index_name”, "alias" : "my_alias" }
    }
  ]
}

/_aliases
{
  "actions" : [
    { "add" :
      { "index" : ["index1", "index2"] , "alias" : "another_alias" }
    }
  ]
}
```

Indices	Alias	Remove/Delete	POST

```
/_aliases
{
  "actions" : [
    { "remove" :
      { "index" : “index_name”, "alias" : "my_old_alias" }
    }
  ]
}
```

Indices	Index Settings	Update	PUT

```
PUT /index_name/_settings
{
  …
}
```

Indices	Index Settings	Reterive	GET

```
/index_name/_settings
```

Indices	Index Settings	Analyze and Get Tokens	GET

```
GET /_analyze
{
  "analyzer" : "standard",
  "text" : "this is a test"
}
```

BIBLIOGRAPHY

[1] A. Singh and H. GonzÂ´alezâ€"VÂ´elez, "Hierarchical Multi-Log Cloud-Based Search Engine", *International Conference on Complex, Intelligent and Software Intensive and Systems*, 2016.

[2] B. Moharil, C. Gokhale, V. Ghadge, P. Tambvekar, S. Pundlik and G. Rai, "Real Time Generalized Log File Management and Analysis using Pattern Matching and Dynamic Clustering", *International Journal of Computer Applications*, vol. 91, no. 16, pp. 1-6, 2014.

[3] "Use of Elastic Search for Intelligent Algorithms to Ease the Healthcare Industry", *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 3, no. 6, 2014.

[4] S. Gupta, "Logging and Monitoring to Detect Network Intrusions and Compliance Violations in the Environment", *SANS Institute InfoSec Reading Room*, 2012

[5] M. Morshedur Hassan, "Current Studies On Intrusion Detection System, Genetic Algorithm And Fuzzy Logic", *International Journal of Distributed and Parallel systems*, vol. 4, no. 2, pp. 35-47, 2013.

[6] J. Bai, "Feasibility analysis of big log data real time search based on Hbase and ElasticSearch", 2013.

[7] S. S. Kaushik, "Detection of Attacks in an Intrusion Detection System", *International Journal of Computer Science and Information Technologies*, vol. 23, no. 982-986, 2011.

[8] P. Sinai Kenkre, A. Pai and L. Colaco, "Real Time Intrusion Detection and Prevention System", *In Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications*, pp. 405-411, 2014.

[9] S. Sheenam and S. Dhiman, "Comprehensive Review: Intrusion Detection System and Techniques", *IOSR Journal of Computer Engineering*, vol. 18, no. 04, pp. 20-25, 2016.

[10] R. Appleyard and J. Adams, "Using the ELK Stack for CASTOR Application Logging at RAL", *International Symposium on Grids and Clouds*, 2015.

[11] M. A. Iversen, "When Logs Become Big Data", Master's thesis, University of Oslo, 2015

[12] A. Nadeem and M. Howarth, "A Survey of MANET Intrusion Detection &amp; Prevention Approaches for Network Layer Attacks", *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2027-2045, 2013.

[13] S. Naik, "A Multi-Fusion Pattern Matching Algorithm for Signature-based Network Intrusion Detection", *International Journal of Research in Engineering, IT and Social Sciences*, vol. 6, no. 8, pp. 36-41, 2016.

[14] O. Kononenko, O. Baysal, R. Holmes and M. W. Godfrey, "Mining modern repositories with elasticsearch", *School of Computer Science University of Waterloo*, 2014.

[15] Rob Appleyard and James Adams. Using the elk stack for castor application

[16] Saurabh Chhajed. Learning ELK Stack. Packt Publishing, 2015

[17] P. ., "INTRUSION-DETECTION SYSTEM FOR MANETS: A SECURE EAACK", *International Journal of Research in Engineering and Technology*, vol. 03, no. 15, pp. 333-337, 2014.

[18] Taipei, Taiwan, 2015. Proceedings of Science. Kurt Hurtado and Tal Levy. Going beyond the needle in a haystack: Elasticsearch and the elk stack.https://speakerd.s3.amazonaws.com/presentations/70d4390b68504e02b4a40a1aa3754532/strata15-elk-stack-needle-haystack.pdf. Accessed: 2016-10-31.

[19] Redis Source Code Repoisitory hosted on Github - https://github.com/antirez/redis

[20] Tuning Logstash Garbage Collection for High Throughput in a Monitoring PlatformDong Nguyen Doan; Gabriel Iuhasz 2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) Year: 2016 - Pages: 359 - 365, DOI: 10.1109/SYNASC.2016.063

[21] P. P. I. Langi and W. Najib, "An evaluation of Twitter river and Logstash performances as elasticsearch inputs for social media analysis of Twitter", *2015 International Conference on Information & Communication Technology and Systems*, 2015.

[22] T. Prakash, "Geo-identification of web users through logs using ELK stack", *Patel2016 6th International Conference - Cloud System and Big Data Engineering*, 2016. DOI: 10.1109/CONFLUENCE.2016.7508191

[23] S. Chen, X. Tang and H. Wang, "Towards Scalable and Reliable In-Memory Storage System", *A Case Study with Redis*, pp. 1660 - 1667, 2016. DOI: 10.1109/TrustCom.2016.0255

[24] R. Lubis and A. Sagala, "Multi-thread performance on a single thread in-memory database", *7th International Conference on Information Technology and Electrical Engineering*, 2015.

[25] J. Edwards, *[26] Security Information and Event Management Buyers Guide 2016*. Massachusetts: Solutions Review, 2016, pp. 1-32. DOI: 10.1109/IACC.2016.141

[26] J. Edwards, *[26] Security Information and Event Management Buyers Guide 2016*. Massachusetts: Solutions Review, 2016, pp. 1-32.

[27] C Geijer, " Log-Based Anomaly Detection for SystemSurveillance ", Master's thesis in Computer Systems and Networks, University of TechnologyGothenburg, Sweden 2015.

[28] A. Fry, "A FORENSIC WEB LOG ANALYSIS TOOL: TECHNIQUES AND IMPLEMENTATION", Master of Information Systems Security, Concordia University Montreal, Quebec, Canada, 2011.

[29] C. Maimbo, "Exploring the Applicability of SIEM Technology in I T Security", Master of Computer and Information Sciences (MCIS), Auckland University of Technology, 2014.

[30] R. Vaarandi, "Tools and Techniques for Event Log Analysis", Doctor of Philosophy in Engineering, Institute of Computer Science, University of Tartu, Estonia, 2005.

[31] "How to collect and visualize your logs with the ELK stack (Elasticsearch Logstash Kibana)", *Scaleway*, 2017. [Online]. Available: https://www.scaleway.com/docs/how-to-use-the-elk-stack-instant-apps/. [Accessed: 22- Aug- 2017].

[32] Linux Notwest [Online]. Available https://www.linuxfestnorthwest.org/2015/sessions/log-analysis-elk-stack-elasticsearch-logstash-kibana [Accessed: 22- Aug- 2017].

[33] A. Gotovskis, S. Å erlinskas and M. MarcinkeviÄ ius, "Query DSL library for Elasticsearch", *GitHub*, 2017. [Online]. Available: https://github.com/ongr-io/ElasticsearchDSL. [Accessed: 22- Aug- 2017].

[34] R. Julin and B. BÃ¶sel, "[34] Simple PHP client for ElasticSearch", *GitHub*, 2017. [Online]. Available: https://github.com/nervetattoo/elasticsearch. [Accessed: 22- Aug- 2017].

[35] "search-documents - Elastica", *Elastica.io*, 2017. [Online]. Available: http://elastica.io/getting-started/search-documents.html. [Accessed: 22- Aug- 2017].

[36] "Home “OSSEC”, *Ossec.github.io*, 2017. [Online]. Available: http://ossec.github.io. [Accessed: 22- Aug- 2017].

[37] D. Parriott, A. Widdersheim and J. Rossi, "OSSEC HIDS Code", *GitHub*, 2017. [Online]. Available: https://github.com/ossec/ossec-hids. [Accessed: 22- Aug- 2017].

[38] " January 2017 Web Server Survey | Netcraft", NetCraft, 2017. [Online]. Available: https://news.netcraft.com/archives/2017/01/12/january-2017-web-server-survey.html. [Accessed: 22- Aug- 2017].

[39] "The Apache http Server Project", Apache Foundation, 2017. [Online]. Available: https://httpd.apache.org/. [Accessed: 22- Aug- 2017].

[40] "The Apache http Server Log Module", Apache Foundation, 2017. [Online]. Available: https://httpd.apache.org/docs/current/mod/mod_log_config.html. [Accessed: 22- Aug- 2017].

[41] "IIS MSDN", MSDN Microsoft, 2017. [Online]. Available: https://msdn.microsoft.com/en-us/library/ms525807(v=vs.90).aspx. [Accessed: 22- Aug- 2017].

[42] "Nginx Wiki", Nginx, 2017. [Online]. Available: https://www.nginx.com/resources/wiki/. [Accessed: 22- Aug- 2017].

[43] "search-documents - Elastica", *Elastica.io*, 2017. [Online]. Available: http://elastica.io/getting-started/search-documents.html. [Accessed: 22- Aug- 2017].

[44] "Nginx Home Clients", Nginx, 2017. [Online]. Available: https://www.nginx.com/. [Accessed: 22- Aug- 2017].

[45] " January 2017 Web Server Survey | Netcraft", NetCraft, 2017. [Online]. Available: https://news.netcraft.com/archives/2017/01/12/january-2017-web-server-survey.html. [Accessed: 22- Aug- 2017].

[46] "Syslog Overview", Labtech, 2017. [Online]. Available: https://docs.labtechsoftware.com/LabTech10.5/Content/Configuration/Syslog/SyslogOverview.htm. [Accessed: 22- Aug- 2017].

[47] "grayling - search ", Sourceforge. [Online]. Available: https://sourceforge.net. [Accessed: 12- Aug- 2017].

[48] "Logcheck -- Logfile Scanner ", Logcheck, 2017. [Online]. Available: http://logcheck.org/. [Accessed: 22- sep- 2017].

[49] "Logwatch - Community Help Wiki ", Ubuntu, 2017. [Online]. Available: https://help.ubuntu.com/community/Logwatch. [Accessed: 10- Aug- 2017].

[50] "AlienVault Unified Security Management & Threat Intelligence ", Allienvault, 2017. [Online]. Available: https://www.alienvault.com/. [Accessed: 22- Aug- 2017].

[51] "IBM® QRadar - Security Analytics Platform ", IBM, 2017. [Online]. Available: https:// www.ibm.com/QRadar/Security. [Accessed: 22- Aug- 2017].

[52] "SIEM & Log Management ", LogPoint, 2017. [Online]. Available: www.logpoint.com/en/. [Accessed: 22- Aug- 2017].

[53] "LogRhythm, The Security Intelligence Company ", LogRhythm, 2017. [Online]. Available: https://logrhythm.com/. [Accessed: 22- Aug- 2017].

[54] "Splunk: Operational Intelligence, Log Management, Application ", Splunk, 2017. [Online]. Available: https://www.splunk.com/. [Accessed: 22- Aug- 2017].

[55] "Tenable™ - The Cyber Exposure Company ", Tensble, 2017. [Online]. Available: https:// www.tenable.com/ /. [Accessed: 22- Aug- 2017].

[56] "Soltra | Cyber Threat Intelligence & Data | Cyber Defense Platform ", Soltra. [Online]. Available: https://www.soltra.com/en/. [Accessed: 22- Aug- 2017].

[57] "Open IC Framework", OpenIOC, 2017. [Online]. Available: http://www.openioc.org/. [Accessed: 22- Aug- 2017].

[58] "Splunk acquires cybersecurity startup Caspida", VentureBeat. [Online]. Available: https://venturebeat.com/2015/07/09/splunk-acquires-cybersecurity-startup-caspida-for-190m/. [Accessed: 22- Aug- 2017].