Customized ELK Stack for Network Data Monitoring and Analytics

by

Henghui Li and Zhiyuan Li

A dissertation submitted to The Johns Hopkins University

in conformity with the requirements for the degree of

Master of Science in Security Informatics

Baltimore, Maryland

April, 2019

Abstract

Nowadays, with the prevalence of big data, the number of logs generated by the system continues to increase, traditional ways of reading or viewing are no longer suitable. To deal with such large-scale log data, ELK stack is an excellent choice for Network data monitoring and log analytics. Considering this, we choose ELK stack as the key point to implement a system which is able to perform different network attack detection. In our capstone project, we first reviewed the literature and existed solution related to this field. Based on this, we set goals for the systems we are going to design and choose Blueliv Data Set, Malware Domain Lists and Spamhaus Spam Database as the data set. Then we give out the design of our system, which is able to handle a ton of log files. To explore how ELK Stack can be utilized on the real-life production environment, we set the Filebeat Nginx Module for log management. And with the logs collected, we focus on the detection for following fields: malicious and botnet IPs visiting, crimeserver IPs Visiting, DNS blocklist IPs visiting, DDOS attack. At last, we use Kibana to visualize and analyze the results.

Acknowledgments

First of all, we would like to express our sincere heartfelt to our advisor , Dr Lei Ding, for her invaluable advice, constant encouragement and precise modification, and we admire her knowledge and personality.

Then we would also like to express our gratitude to the faculties who work in the library because they provide us with great convenience in writing our researches.

Finally, we would like to thank classmates who friendly encourage us in writing the thesis and overcoming the difficulties.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Idea

As the number of logs generated by the system continues to increase, a large number of logs are no longer suitable for manual reading or viewing(Clinton Gormley, 2014). Also when we deploy a clustered server, the log files are scattered across multiple servers("Elastic Website"). View log information needs to be fetched and viewed on each server. Therefore, some techniques are often used to analyze existing logs and display the log information in a graph or other easy-to-read manner. Based on this, the ELK stack comes into our sight(Dean J., 2008). ELK is the abbreviation of three popular open-source projects: Elasticsearch, Logstash, and Kibana. For the analysis log, it is generally divided into three steps: log collection, log storage, and data display. Corresponding to ElasticStack: L(ogstash), E(lasticsearch), K(ibana)(Lu, 2019). Elasticsearch is an open-source, restful, distributed search and analytics engine built on Apache Lucene. Support for various languages, high performance, and schema-free JSON documents makes Elasticsearch an ideal choice for

various log analytics and search use cases.

Logstash is an open-source data ingestion tool that allows you to collect data from a variety of sources, transform it, and send it to your desired destination. With pre-built filters and support for over 200 plugins, Logstash allows users to easily ingest data regardless of the data source or type.

Kibana is an open-source data visualization and exploration tool for reviewing logs and events. Kibana offers easy-to-use, interactive charts, pre-built aggregations and filters, and geospatial support and making it the preferred choice for visualizing data stored in Elasticsearch.

With the help of he ELK stack, we are able to aggregate logs from all our systems and applications, analyze these logs, and create visualizations for application and infrastructure monitoring, faster troubleshooting, security analytics, and more.

In our capstone project, we are aiming to analyze intrusion from the logs, and our system is able to find out the attacks in real time or from the logs stored off-line. Following the log analysis process, we plan to collect and store the logs from different systems using Logstash, and then analyze the logs with the help of Elastic search. After that, we use Kibana to visualize the result, which makes it easier for us to have an understanding of how the data is distributed and conduct security analytics.

1.2 Systems Objects

With the prevalence of big data, it is inevitable there are more and more malicious programs and attacks targeting the server. Elasticsearch, Logstash,

and Kibana are the core suites that makeup ELK, but not all suites. Depending on the application scenario, ELK can implement different architectures with different kits. Based on this, our system is designed to provide service that has a higher accuracy on detecting the intrusion. In our system, there are several features we want to reach:

1. Detect DDOS attack: As DDOS attack is one of the most popular attacks and will cause catastrophic consequences to the system, we need to pay much of our attention to this part.

2. Detect botnet IP: Find out whether there is an IP address belongs to the botnet trying to visiting our server and alert for it.

3. IP DNS Blocklist: Check the IP address of the sending mail server against a public list of mail servers known to send spam.

4. Detect abnormal behavior: Try to find out unusually dangerous actions, for example, is there anyone in my organization visiting a known malware URL/domain

1.3 Assumption

In order to describe system functionality, we would make some definitions and assumptions. In the current stage, these assumptions would simplify the system situation. And we would try to reach the assumptions in the next stage.

1. All normal traffic and intrusion can be recorded by the system. And

from the records, we can find out the IP address, time, events and all the other necessary information.

2. The IP addresses or the IP addresses controlled by the attacker are in the dataset we are going to use.

3. DNS blocklist and the malware domain list contains all the upcoming attacks

Chapter 2

Literature Review and Problem Definition

2.1 Log

The definition of logs are time-series based machine data, including IT system information (servers, network devices, operating systems, application software), and various sensor information for the Internet of Things. The log reflects user behavior and in fact data.

The log processing system has undergone a long period of development and has three main phases(Jayathilaka H., 2017):

1. Log processing v1.0: The log is not processed centrally; the log is only traced after the event, and the log is not detected after the hacker is invaded; using the database to store the log is not suitable for complex event processing.

2. Log processing v2.0: Using Hadoop platform to realize log offline batch processing, the disadvantage is poor real-time performance; using Storm

stream processing framework, Spark memory computing framework to process logs, but Hadoop/Storm/Spark are programming frameworks, not a real-time system

3. Log processing v3.0: Use log real-time search engine to analyze logs. The first feature is fast. The log is delayed from the generation to the search analysis. The second is large, and the amount of TB logs is processed every day. The third is flexible. Search for any logs. The solutions represented are Splunk, ELK, SILK.

2.2 Traditional Methods and Current Problems

All information system platforms generate a large number of logs every day, usually based on streaming data, including user access records, database operation records, etc. When the amount of data reaches a certain order of magnitude, the traditional single-node system cannot complete the retrieval and analysis tasks. (Siregar, 2017)(A.Abdulraheem, 2007)They must be processed using a distributed logging system. In general, these systems need to have the following characteristics:

1. build a bridge between application systems and analysis systems
2. support quasi-real-time online analysis systems and off-line analysis systems
3. have high scalability and reliability, that is, when the number of data increases, the horizontal performance can be extended by adding nodes;

when one or some nodes fail, only the performance of the system is affected, and the system functionality is not affected

In a large-scale cluster system, logs are distributed and stored on different devices. The traditional method uses filtering tools such as cat, tail, sed, awk, grep to filter and output the analysis method because the efficiency is low and no longer applicable and also increase workload.

The most urgent task is to use a centralized log management platform to collect and collect logs from all servers for monitoring and analysis. The excellent system operation and maintenance platform can not only realize the centralized management of the components of the data platform, facilitate the daily monitoring of the system operation and maintenance personnel, improve the operation and maintenance efficiency, but also feedback the system operation status to the system developers.

2.3 Related Works

As network data monitoring and log analysis is a really critical topic, many researchers have conducted their research on it. Literature(Chen Fumei, 2017) studied various subsystems of distributed data stream processing system in a big data environment, including data collection subsystem, message queue management subsystem, streaming data processing subsystem and data storage subsystem, for 4 subsystems. The key technologies involved were detailed and compared from an application perspective.

Literature (Liao Xiangke, 2016) systematically reviewed the progress of log

research and summarized the research fields, research methods, research directions and future challenges of the log from three aspects: log feature analysis, log fault diagnosis and log analysis. Strong guiding significance.

Literature (Zhao Yining, 2017) introduced the optimization method of log pattern refining algorithm for large-scale system log using MapReduce mechanism in large system and pointed out that the research and use of ELK combination developed by Elastic have shown a significant growth trend in recent years. ELK The real-time data analysis framework has certain advantages for log stream processing, and other auxiliary analysis programs are still needed for the analysis requirements of specific systems or environments.

Literature (Bai Ju, 2014) uses Flume to collect logs, uses plug-ins to provide log events for Hbase, uses Elasticsearch for log search, and proposes a real-time big data log search integration scheme, which has some reference significance, but the transmission and storage of logs are too complicated. ELK technology has relevant applications in big data systems, e-commerce platforms, astronomical systems, and power systems, and has good effects in system monitoring and data analysis. Based on this, our capstone project further explores and summarize Elastic to implement log security analysis.

Literature (Merve Astekin, 2018) designed LogEnhancer to reduce the number of software uncertain branches by enhancing the log content, thus reducing the difficulty of branch critical condition judgment and error reproduction, and improving diagnostic efficiency. Specifically, a section is assumed, code with multiple branches, there are more branch statements make the program difficult to debug. At this time, the condition variable is written into the log information, and the judgment result of the previous conditional statement

can be judged by the value of the condition variable in the log information. To eliminate an indeterminate branch, it should be noted that even with LogEnhancer, most of the logging behavior needs to be decided by the developer. At present, the large-scale software log is large, but the amount of information contained in the log is Insufficient, developers usually need to add more dynamic information of the program runtime to understand the cause of the failure, that is, frequently update the log, and continuously add new program variables to the log message. Study the method of automatically enhancing the log information. Can greatly reduce the developer's workload and improve the efficiency of code debugging(Xu W., 2009).

2.4 Term of Explanation

1. DDOS attack: Distributed Denial of Service (DDoS) attacks refer to the use of client/server technology to combine multiple computers as an attack platform to launch DDoS attacks on one or more targets, thereby multiplying denial of service attacks by power.

   There are many ways to attack DDoS. The most basic DoS attack is to use a reasonable service request to occupy too many service resources(Gu X., 2009), so that legitimate users can not get the response of the service. A single DoS attack generally adopts a one-to-one approach. When the target CPU speed is low, the memory is small, or the network bandwidth is small, and the performance is not high, the effect is obvious.

2. Malware Domains: The Malware Domains page lists domains that are

known to generate spam, host botnets, create DDoS attacks, and generally contain malware.

3. Botnet: A botnet is a network that can be used to control a bot program virus by using one or more means of communication to form a one-to-many control network between the controller and the infected host. The attacker spreads bots through various channels to infect a large number of hosts on the Internet, and the infected host will receive an attacker's instructions through a control channel to form a botnet.

   Launching DDos attacks using Botnet is one of the most important threats at present. Attackers can send instructions to all bots they control, allowing them to simultaneously access specific network targets at a specific time, thus achieving the goal of DDos. Because Botnet can form a huge scale, and it can be better synchronized by using DDos attack, it can make DDos more harmful and prevent more difficult when issuing control commands(Zaharia M., 2012).

4. DNS blocklist: IP DNS Blocklist checks the IP address of the sending mail server against a public list of mail servers known to send spam. The DNS blocklist is actually a list of IP addresses that can be queried. The DNS query method is used to find out whether an A record of an IP address exists to determine whether it is included in the DNS blocklist. The IP address in the list indicates that the spam has been posted externally. Therefore, it is used by the spam filter to filter spam sent by the list like a virus definition file.

2.5 Dataset Description

In our capstone project, in order to detect the DDOS attacks and other intrusions, we mainly use the following datasets to conduct our experiments:

1. Blueliv Data Set

   This is a dataset created and maintained by Blueliv, that allows accessing Blueliv's Cyber-Threat Intelligence feeds. All the information in the dataset is the real data collected form enterprises. Depending on this dataset, we are able to get a huge amount of log records including normal traffic and intrusion data. These data are made up of various kinds of attacks, and due to the limit of time, we mainly focus on the data of Crime Servers and Bot IPs.

   The crime servers data set contains the information of the malicious servers previously tried to launch an attack to other servers, which means it is controlled by the attackers or it was successfully attacked by the malicious servers and then became another source of the attack.

   The Bot IPs dataset contains the information of the botnet. Combining the Bot IPs dataset with the Crime Server Set, we can have a more complicated and complete data set to implement our experiment.

2. Malware Domain Lists

   This data set can be obtained from the malware domain lists website https://www.malwaredomainlist.com/mdl.php, and it contains the details of various malicious domain lists. This data set is widely used, we can use this data set to analyze the user's abnormal behavior. More

specifically, we can try to find out is there anyone in our organization try to visit a known malware domain recorded in the dataset. And if we can find out the abnormal actions, we can know the visit to the malware domain is by mistake or is on purpose, and give out alerts and countermeasures if it is already controlled by the attacker.

3. Spamhaus Spam Database

   Spamhaus is an international non-profit organization whose main task is to track Internet spam gangs, real-time blacklisting technology. Spamhaus has released a large number of spam organization databases, including SBL, XBL, and PBL.

   (a) XBL (Exploits Block List)

   It is a real-time blacklist IP list for hijacking machines (such as zombies) or worms/viruses with built-in spam engines and other types of Trojans.

   (b) SBL (The Spamhaus Block List)

   It is a verified spam source and a real-time blacklist of spam messages. It is also one of the main projects of spamhaus, with new records and records deleted 24 hours a day, 7 days a week, in 9 countries around the world.

   (c) PBL (The Policy Block List)

   It is primarily an IP address segment that contains dynamic IP and which SMTP servers are allowed to send mail without authentication.

In our capstone project, we use ZEN, it is the collection of the above three, that is, the data including XBL, SBL, and PBL.

Chapter 3

System Solution, Design, Implementation, Result and Analysis

3.1 System Design and Key Components

The traditional way to do log management and analytics is to log in to systems (most likely the Linux systems) and use command line tools such as cat, tail, sed, awk, grep, etc. to filter and output the data for analytics. This method could only process a small amount of data, and more often, sometimes developers who need to look at log files are not allowed to have certain server privileges. Now Elk Stack provides developers an efficient way to manipulate log data. Our design is trying to make good use of this amazing framework to get the information we need in a handy way.

Since in this project our goal is to conduct security and log analytics on web applications, we design the whole system and select the ELK stack components based on our requirement and use cases. Our solution makes use of several most popular components of ELK stack, which have complete functionality

that covered on a ton of log and security analytics use cases.

Generally speaking, Logstash is mainly used to collect the logs sent by each Shipper, and then do some filtering and parsing functionality, and then send the processed data to Elasticsearch to store. Elasticsearch servers as a database, it is not similar to any relational database or non-relational database, it has its own way to store data, indexing data for further query and for the most important aggregation and analytics on data. Kibana is used as the user interface of Elasticsearch and the visualization tool for our input data. It provides configurable user-friendly interface for visualization data that stored and indexed on Elasticseach. The shippers, in this case, are also Elk Stack components. Elastic provides a lot of different kind of shippers called Beats for different use cases, for exmaple, Filebeat for log files, Packetbeat for network data, Winlogbeat for Windows Event Logs.

We have considered several different designs of our solution and there are all popular designs in the industry. The easiest design would be shown as in Figure 3.1. Filebeat installed on each host to collect and ship the data to Elasticseach. The Elasticsearch endpoint could be deployed either on the Cloud, or on the local environment.

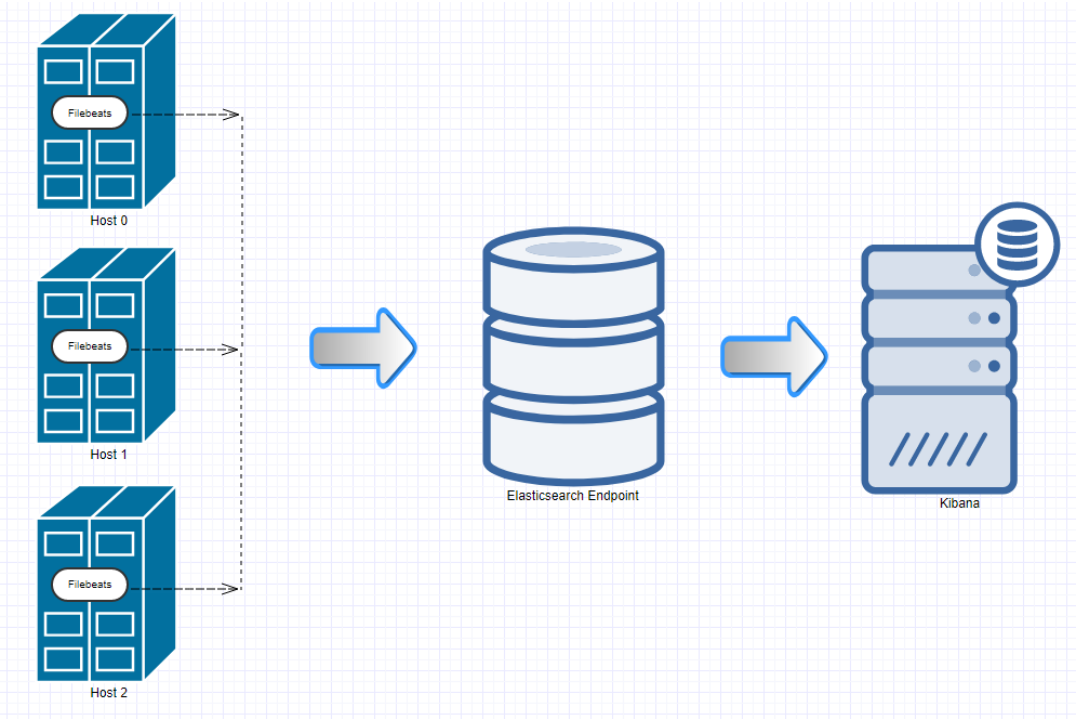

Figure 3.1: System Design 1

In the production environment, there are a ton of log files need to be shipped and transfer. A data-intensive application could use a message queue to ensure the data is shipped correctly to Elasticsearch endpoint. This system design would be shown as in Figure 3.2.

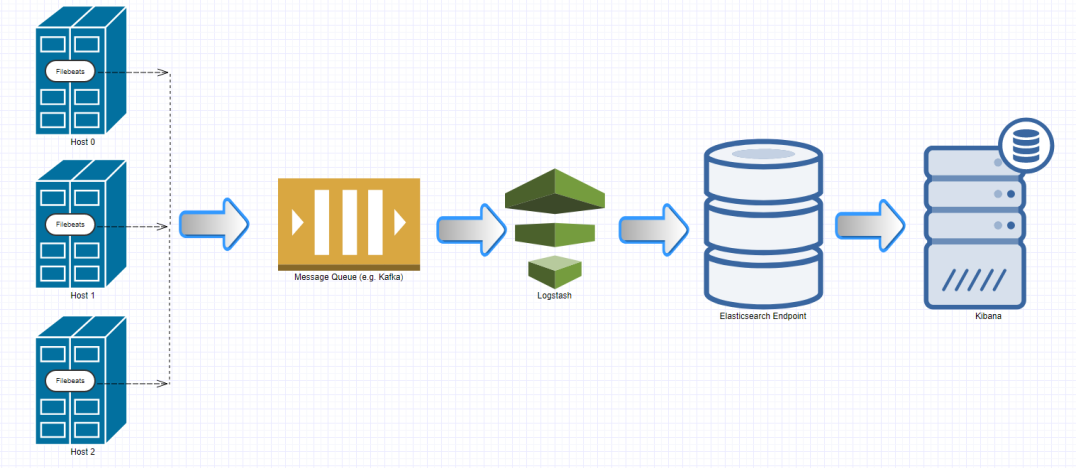

Figure 3.2: System Design 2

Since we don't have very intensive data and all the data are generated from our self-deployed server on AWS Cloud, based on the use cases and requirement of our project, eventually, our solution built on the design as shown in Figure 3.3.

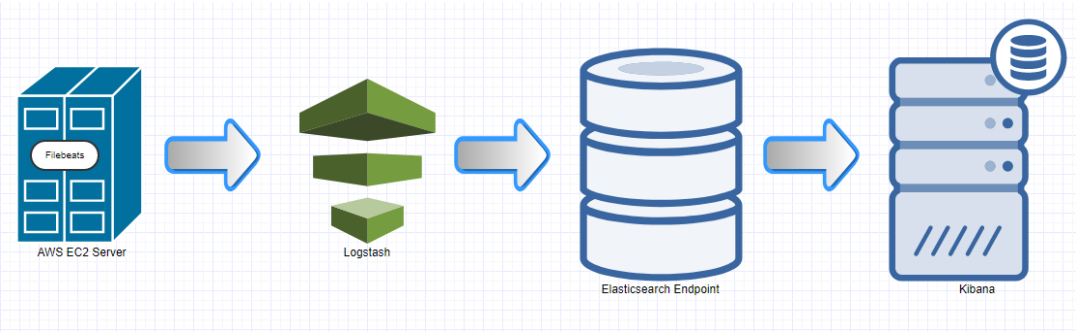

Figure 3.3: System Design 3

Based on this design, we use AWS EC2 server as a host to deploy our web application server. We deployed a simple node.js web application server and utilize Nginx as a load balancer. Nginx also servers as web application gathering tool to collect our application log data. The technical detail would

be recorded in the following part of this report.

3.2 Log Analytics on Open-source Server Log files

3.2.1 Open-source Server Log files

Since the goals of our project mainly focus on utilizing the ELK Stack framework for log management and security analytics, the best way to get started is making good use of the public open-source log files. Typically, the web server applications generate different types of logs, including access logs, error logs, agent logs, and etc. Currently, the popular web servers or HTTP servers in the industry such as Tomcat, Nginx, and Apache are all automatically generating these multiple types of logs for the application deployed above it.
An HTTP server or web server is essentially a Layer 5 application of OSI network layers. It usually runs on the host server, binding server's IP address and listening on a TCP port to receive and process HTTP requests, so the clients such as Chrome, IE, Firefox can obtain web pages (HTML format), documents (PDF format), audio (MP4 format), video (MOV format) and many other things on the server through HTTP protocol. Typically, an access log message of an HTTP server can contain information as follows.

1. A time-frame indicating response time
2. An IP address indicating the client's IP
3. A HTTP Status Code indicating the status of the requesting resource
4. An URL of requesting resource

5. An user agent string that can be used as a identifier such as a computer's hostname or browser's version

The open source Apache HTTP server we used are from a web blog. A typical message looks like as shown in Figure 3.4.

```
83.149.9.216 - - [28/May/2014:16:13:42 -0500] "GET /presentations/logstash-monitorama-2013/images/kibana-search.png HTTP/1.1" 200 203023
"http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
83.149.9.216 - - [28/May/2014:16:13:42 -0500] "GET /presentations/logstash-monitorama-2013/images/kibana-dashboard3.png HTTP/1.1" 200
171717 "http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
```

Figure 3.4: Log Message Sample

3.2.2 ELK Stack Component Configuration

We use Filebeat as a log files collector and a light-weight shipper. Therefore, we configure Filebeat to read the log files from a certain location. The configuration details are specified in the filebeat.yml file. Specifically, we enable Fillebeat inputs to read the certain log files we wish to be processed, configuring the Logstash as outputs and specifying the host and port number that is running the Logstash application. The log data will be fetched from the local directory and shipped to Logstash for further parsing and analyzing.

The three main functions in the Logstash instance are event input, event data filtering, and event output. These three functions of Logstash are performed based on configuration information, which is stored in an easy-to-understand .conf file. There are different configuration sections in the .conf file that correspond to the three different types of plug-in, including inputs, filters, and outputs. Each Logstash instance is customized based on its requirements in the overall architecture. The Logstash configuration is a key step in this process. We configure Logstash and customize the plug-ins to define the rules

of the data parsing process. One great thing about this process is that we can gather additional information using many helpful plug-ins, such as GeoIP and user agent info, which provides us more information to support our analytics and management. There are multiple plug-ins we have used to configure Logstash, which are shown as following Table 3.1.

Plug-in	Description
beats	receive events from the Elastic Beats framework
grok	parse unstructured log to be structured and queryable
mutate	perform general mutations on fields
date	parse dates from fields
geoip	add information about the geographical location of IPs
useragent	add information about user agent like operating system
stdout	a simple output which prints to the STDOUT
elasticsearch	output log data into Elasticsearch

Table 3.1: Logstash Configuration 1

3.2.3 Visualizations and analysis

Based on the configuration of Logstash and Filebeat, log files are parsed and analyzed and then indexed into Elasticseach for analytics and aggregation. We utilize the Kibana as the user interface to communicate with Elasticsearch. Kibana provides a development tool section for users to communicate with Elasticsearch, which basically are an IDE-like JSON commands interpreter. There are two ways to conduct data analytics: using the built-in aggregation APIs and using the Kinaba user interface. We have tried both methods to analyze the data.

Based on the enriched information of GeoIP plugin, We are able to analyze and figure out the countries that sent the most server requests and which cities

in each country sent the most requests. And the specific number of requests sent by these countries and cities. This information is presented directly to the user in the form of a histogram, allowing the user to clearly understand the useful information, as shown in Figure3.5. Also, we can customize a more clearly geo-coordinate graph indicating the countries and the city in the world map that sent the most requests, the most requests sent, the thicker mark on the map, as shown in Figure3.6

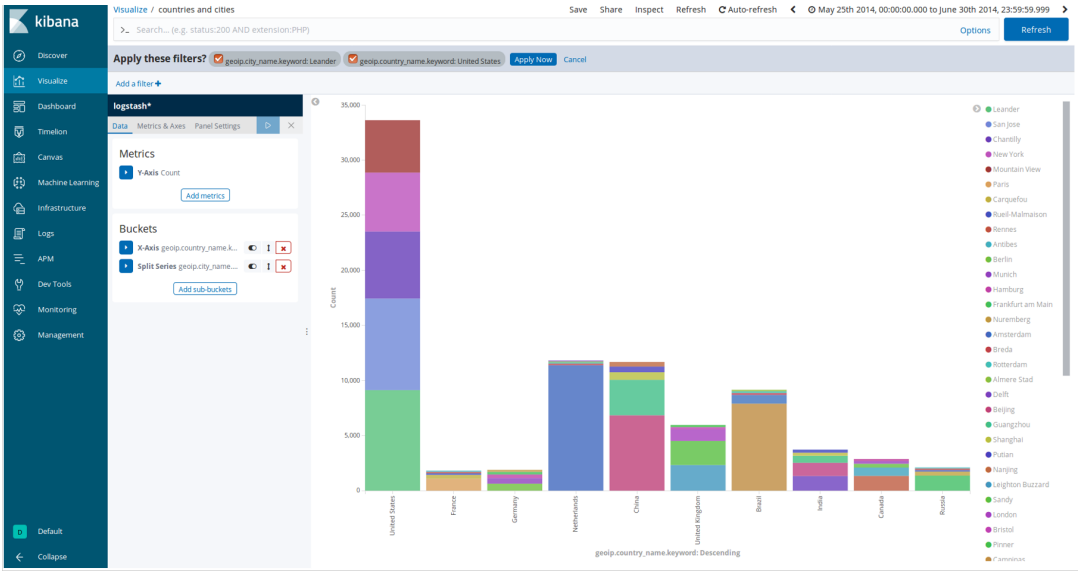

Figure 3.5: Dashboard 1

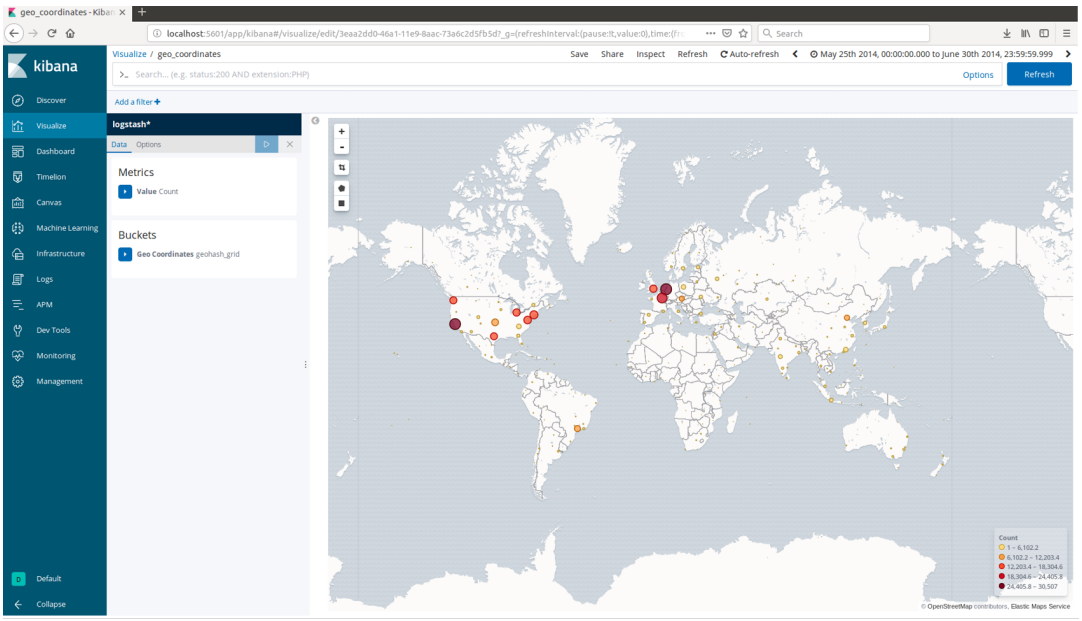

Figure 3.6: Dashboard 2

Furthermore, we are able to analyze the OSs that made the most requests, the browsers that made the most requests, along with with the historical graph indicating the day and the time that have the most clients visiting the web server. All the charts and graph could be arranged to be one dashboard and can be customized to fit the users use cases and requirement. As shown in Figure3.7

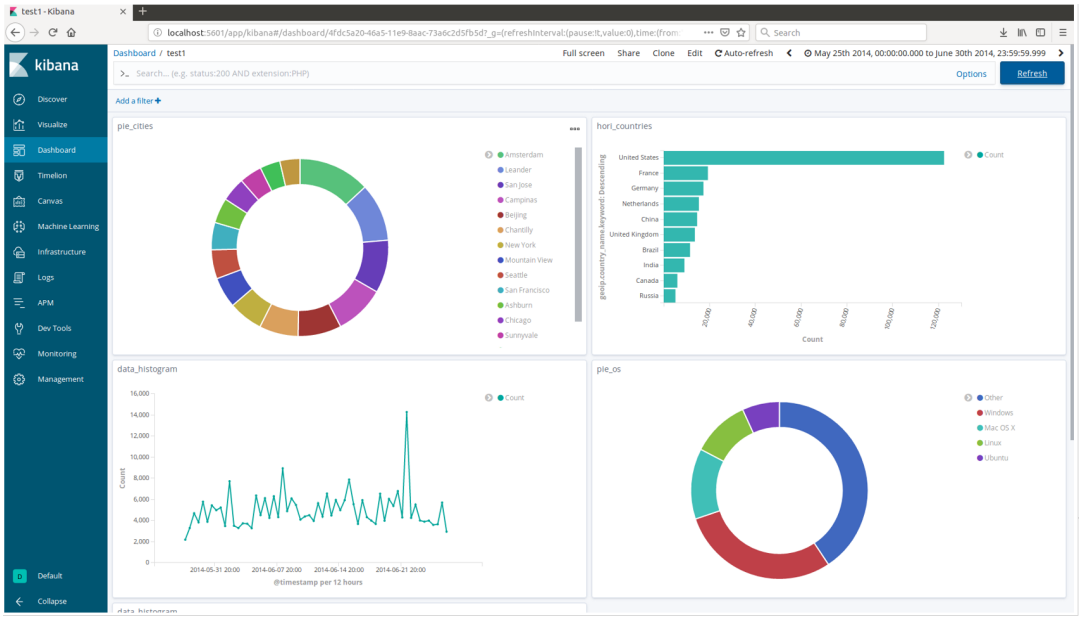

Figure 3.7: Dashboard 3

3.3 Security Analytics on Self-Deployed Web Server Log

3.3.1 Nginx Server and Node.js Web Application Server Log

In order to simulate the real-life production environment and to explore how ELK Stack can be utilized on the real-life production environment, we have built a real web application server and deployed it on Amazon AWS Cloud to let it run all the time to collect server log data. There data are streamed to a Elasticsearch endpoint to be indexed and eventually being visualized and analyzed.

Before anything could be done, we create an Amazone EC2 cloud server as a host server for our web application. Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the

cloud. It is designed to make web-scale cloud computing easier for developers. We create a Linux server with t2.micro instance usage based on our application requirement. And then, we build a light-weight node.js web application based on an open-source node.js project and we rewrite it to be an one-page web application and receiving static requests. We also utilize Nginx as a load balancer. There are some amazing features that attract us to choose Nginx:

1. Light-weight, taking up less memory and resources
2. High performance, processing requests asynchronously and non-blockingly
3. Simple configuration, easy startup, and good reliability
4. Log file formatting are clearly and easily understanding
5. More suitable for handling static requests

After we set up the nodejs application and successfully configure Nginx, we buy a domain from Goddady.com and bind it to the AWS EC2 IP address so that we can use our domain name to request the resource of the application, which is more similar to the real-life production environment. Once the web application has started up, access it by using the domain to send HTTP request the client would receive response as shown in following Figure3.8.

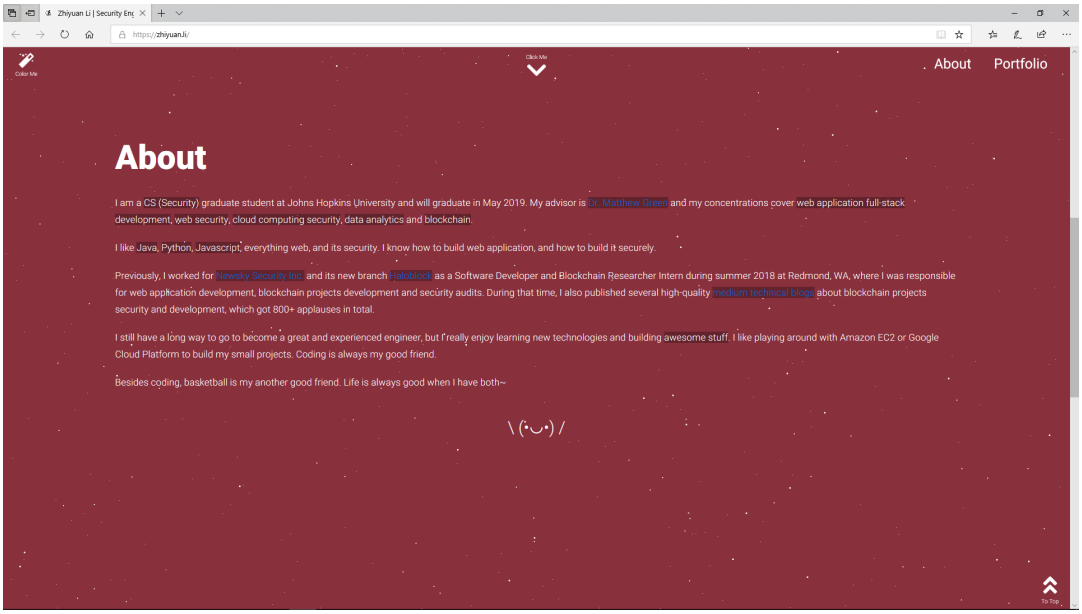

Figure 3.8: Server Screen-shot

After we set up our web application for a while, we are able to gather some log files for further analytics, including the server access log and error log. We gather all access logs and error logs during March and April 2019, and finally they are streamed to the Logstash endpoint and Elasticseach endpoint. A typical access log message and error message are shown in following Figure3.9 and Figure3.10.

```
52.53.201.78 - - [22/Apr/2019:13:00:43 +0000] "GET / HTTP/1.1" 200 870 "http://www.zhiyuan.li/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/61.0.3163.100 Safari/537.36"
180.76.15.34 - - [22/Apr/2019:13:22:21 +0000] "GET /static/js/bundle.js HTTP/1.1" 301 194 "-" "Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)"
180.76.15.155 - - [22/Apr/2019:13:22:23 +0000] "GET /static/js/bundle.js HTTP/1.1" 200 2432084 "-" "Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)"
```

Figure 3.9: Server Access Log

```
2019/04/11 11:22:05 [crit] 20062#20062: *29564 SSL_do_handshake() failed (SSL: error:1408A0A0:SSL routines:ssl3_get_client_hello:length too short) while SSL handshaking, client:
108.178.16.154, server: 0.0.0.0:443
```

Figure 3.10: Server Error Log

3.3.2 Nginx Module

Elk Stack provides a lot of helpful components for Nginx log analytics, one is Filebeat Nginx Module. The module is a feature provided by Filebeat to simplify the log management process. It provides a handy solution for the collection, parsing, and visualization of common log formats. Filebeat has built-in modules for most common log files, such as Apache, Nginx, Syslog, etc. In our project, we enable the Nginx module to simplify the process, as shown in Figure3.12.

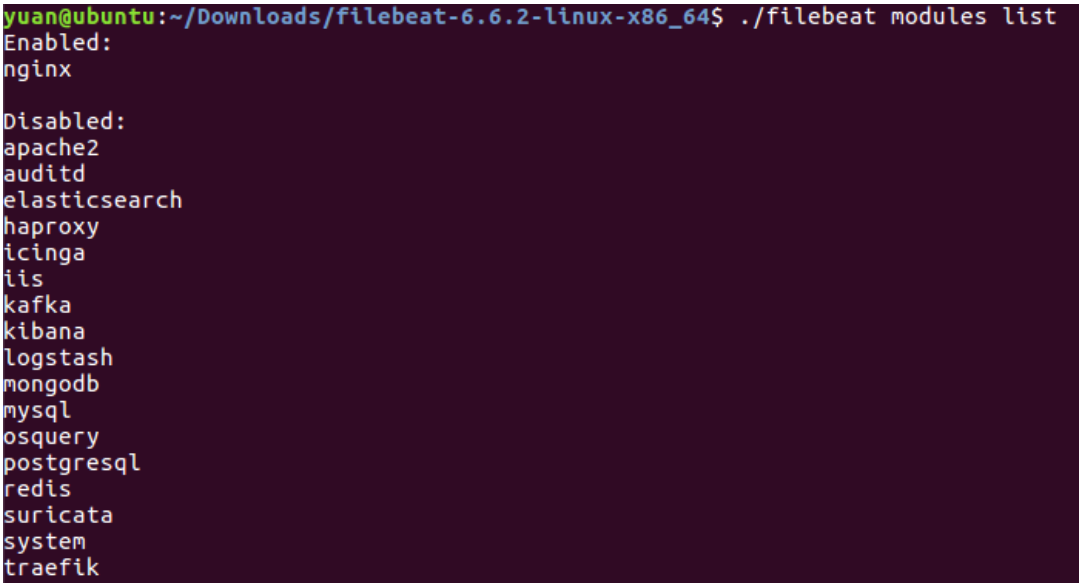

Figure 3.11: Filebeat Module

The convenience provided by Nginx module including loading the recommended index template for writing to Elasticsearch and deploying the sample dashboards for visualizing the data in Kibana. The sample dashboards could be very helpful for users, such as Overview of the Suricata Alerts dashboard, Dashboard for the Filebeat Nginx module, and Syslog dashboard from the

Filebeat System module, and etc. These pre-built templates not only can save users time of configuration of dashboards but also could be very effective in log management and security analytics.

3.3.3 Logstash Data Enrichment for Security Use Cases

For security use cases, since the log data files we used are mostly web application server access logs and error logs, even though these data are simple, they still can provide a ton of security-related insights of the web applications and servers. With Logstash data enrichment features, we can add additional security-related information into the data. By exploring the additional security-related insights provided by Logstash, we find that it is perfect for Logstash data enrichment applies for some situations shown as follows.

1. Detection of any Botnet IP or malicious IP hitting the server
2. Detection of any DNS Blocklist IP hitting the server
3. Detection of any Crimeserver IP hitting the server
4. Detection of any known malware URL or domain sending requests to the server
5. Alert of any DDOS attack against the server

Based on the features provided by Logstash data enrichment, our solution utilized some third parties data feeds, such as Malware Domain List and Blueliv threat data feeds. These data feeds are either some open-source free data feeds that available for public, or some commercial data feeds but providing free

data feeds service for research or studies purposes. They are all frequently updated to support users catch up with new threat data and information. By utilizing these free threat data feeds combined with ELK stack analytics functionality, we are able to gather security related insights from our web application server access logs and error logs.

3.3.4 Malicious and Botnet IPs Visiting

We have used multiple different threat data feeds as our input of Logstash. There are several major threat data feeds we are using in the project, including Malware Domain List and Blueliv threat data feeds. Malware Domain List is a non-commercial community project, providing a frequently updated threat data in a CSV format. It can be used to analyze malicious websites, domains, and URLs. Blueliv is a Cybersecurity company providing threat data feeds and can be accessed by using the API key provided by the company, some of their services are commercial and some of are free. They provide threat data including botnet IPs data feeds, crime servers data feeds, malware data feeds, and attacks data feeds. We mostly make use of their crime servers data feed in our project. The Elk Stack components, mostly Logstash, are needed to customize and configure correspondingly in order to be suitable for these different threat data feeds.

The first threat data feed we are using is the Malware Domain List, which provides threat intelligence data in a CSV format. Following Figure3.9 have shown the message format providing by MDL. Basically, it is a CSV file that includes thousands of line of threat messages, each message including

a malicious URL or malicious domain name, a corresponding malicious IP address, a category of malicious type, a region code, and some other related information. In this project, we mostly focus on the malicious IP addresses in this list, anytime a malicious IP hitting or visiting our server, we are able to tag this request as a malicious visiting and alert or record it in our Elasticsearch endpoint, visualizing it in Kibana so that corresponding responsive actions could be taken.

2216	2016/08/29_14:25	unlink.altitude.lv/vdgqb3.html	93.190.140.162	customer.worldstream.nl.	gateway to EK	-	49981	0 NL
2217	2016/08/29_15:40	cqji.artidentalkurs.com/vdgqb3.html	93.190.140.162	customer.worldstream.nl.	gateway to EK	-	49981	0 NL
2218	2016/08/30_12:20	rufex.aifingenieros.cl/dsgajo3.html	93.190.140.162	customer.worldstream.nl.	gateway to EK	-	49981	0 NL
2219	2016/08/30_12:25	wuvac.agwebdigital.com/dsgajo3.html	93.190.140.162	customer.worldstream.nl.	gateway to EK	-	49981	0 NL
2220	2016/09/01_11:55	tanner.alicerosenmanmemorial.com/hggfgl3.html	93.190.140.162	customer.worldstream.nl.	gateway to EK	-	49981	0 NL
2221	2016/09/01_14:55	sanya.vipc2f.com/ceqxwu3.html	93.190.140.162	customer.worldstream.nl.	gateway to EK	-	49981	0 NL
2222	2016/09/01_16:55	livre.wasastation.fi/ceqxwu3.html	93.190.140.162	customer.worldstream.nl.	gateway to EK	-	49981	0 NL
2223	2016/09/01_17:00	pogruz.wanyizhao.net/ceqxwu3.html	93.190.140.162	customer.worldstream.nl.	gateway to EK	-	49981	0 NL
2224	2016/09/01_17:35	tahit.wastech2016.in/xcqrsw3.html	93.190.140.162	customer.worldstream.nl.	gateway to EK	-	49981	0 NL
2225	2016/09/05_09:37	ad.9tv.co.il/serv4/www/delivery/ajs.php?zoneid=37&cb=54350405237&charset=utf-8	62.219.67.44	bzq-67-44.red.bezeqint.net.	iframe on compromised site leads to exploit	-	8551	0 IL
2226	2016/09/05_09:37	giants.yourzip.co/static/quotes.js?ver=d58072be2820e8682c0a27c0518e805e	5.200.55.58	-	leads to exploit kit	-	48096	0 RU
2227	2016/09/05_09:37	evans.babajilab.in/specimen/1479491/tire-something-detect-five-what-knot-unknown-entertain-stiff	85.143.219.181	60567.simplecloud.club.	exploit kit	-	201848	0 RU
2228	2016/09/05_10:07	ross.starvingmillionaire.org/unveiled/dropdown.js?ver=496e05e1aea0a9c4655800e8a7b9ea28	5.200.55.58	-	leads to exploit kit	-	48096	0 RU
2229	2016/09/06_11:49	structured.blackswanstore.com/plc/header.js	5.200.55.91	-	leads to exploit kit	-	48096	0 RU
2230	2016/09/06_12:42	jessisjewels.com/disk/update/postmaster/en/?ar=yourname@yourdomain.com	50.87.153.96	50-87-153-96.unifiedlayer.com.	phishing site	-	46606	0 US
2231	2016/09/15_08:48	tscl.com.bd/m/Rfi%20XINl%20QUOTATION%20LIST.zip	209.99.16.206	206.0/24.16.99.209.in-addr.arpa.	trojan inside zip file	-	394695	0 US
2232	2016/09/15_10:06	catjogger.win/ganel/gate.php	213.145.225.170	web02.chillydomains.com.	pony loader c&c	-	25575	0 AT
2233	2016/09/21_12:12	art-archiv.ru/images/animated-number/docum-arhiv.exe	81.177.139.111	-	trojan	-	8342	0 RU
2234	2016/10/13_14:03	elmissouri.fr/data.dpg	213.186.33.50	cluster017.ovh.net.	ransomware	-	16276	0 FR
2235	2016/10/30_01:52	kingskillz.ru/~kingskil/Prince/Man/lucy/mine/shit.exe	85.143.215.183	62695.simplecloud.club.	Trojan.Farelt	-	201848	0 RU
2236	2017/01/19_13:05	61kx.uk-insolvencydirect.com/sending_data/in_cgi/bbwp/cases/Inquiry.php	35.166.113.223	ec2-35-166-113-223.us-west-2.c	leads to ransomware	-	16509	0 US
2237	2017/01/19_13:05	daralasnan.com/wp-content/plugins/mkazaqbya/vmywyvz4.php	166.62.12.1	sg2nlhg800c1800.shr.prod.sin2.sec	leads to ransomware	-	26496	0 US
2238	2017/01/19_13:05	www.studiolegaleabbruzzese.com/wp-content/plugins/urxwhbnw3ez/flight_4832.pdf	62.149.142.206	webx440.aruba.it.	ransomware	-	31034	0 IT
2239	2017/01/19_13:05	raneevahijab.id/admin/box/workspace/	103.24.13.91	server3.e-cbncloud.co.id.	phishing site	-	132644	0 ID
2240	2017/01/25_20:15	www.lifelabs.vn/api/get.php?id=aW5mb0BzYXBjdXBncmFkZXMuY29t	118.69.196.199	-	Trojan.Backdoor, Office.Word.Downloader	-	18403	0 VN
2241	2017/01/25_20:16	falconsafe.com.sg/api/get.php?id=aW5mb0BzYXBjdXBncmFkZXMuY29t	43.229.84.107	-	Trojan.Backdoor, Office.Word.Downloader	-	38532	0 SG
2242	2017/02/09_14:04	fo5.a1-downloader.org/g2v9s1.php?id=yourname@yourdomain.com	188.225.32.177	vds-tibca.timeweb.ru.	trojan download	-	9123	0 RU
2243	2017/03/06_21:09	www.hjaoopoa.top/admin.php?f=1.gif	52.207.234.89	ec2-52-207-234-89.compute-1.a	Cerber ransomware	-	14618	0 US
2244	2017/03/06_21:09	up.mykings.pw:8888/update.txt	60.250.76.52	60-250-76-52.HINET-IP.hinet.net.	related to a Mirai windows spreader trojan	-	3462	0 TW
2245	2017/03/06_21:09	down.mykings.pw:8888/ver.txt	60.250.76.52	60-250-76-52.HINET-IP.hinet.net.	related to a Mirai windows spreader trojan	-	3462	0 TW
2246	2017/03/06_21:09	down.mykings.pw:8888/ups.rar	60.250.76.52	60-250-76-52.HINET-IP.hinet.net.	related to a Mirai windows spreader trojan	-	3462	0 TW
2247	2017/03/14_23:02	ssl-6582datamanager.de/	54.72.9.51	ec2-54-72-9-51.eu-west-1.comp	redirects to Paypal phishing	-	16509	0 US
2248	2017/03/14_23:02	privatkunden.datapipe9271.com/	104.31.75.147	-	Paypal phishing	-	13335	0 US
2249	2017/03/20_10:13	alegroup.info/ntnrrhbt	194.87.217.87	mccfortwayne.org.	Ransom, Fake.PCN, Malspam	-	197695	0 RU
2250	2017/03/20_10:13	fourthgate.org/Yryzvt	104.200.67.194	-	Ransom, Fake.PCN, Malspam	-	8100	0 US
2251	2017/03/20_10:13	dieutribenhkhop.com/parking/	84.200.4.125	125.0-255.4.200.84.in-addr.arpa.	Ransom, Fake.PCN, Malspam	-	31400	0 DE
2252	2017/03/20_10:13	dieutribenhkhop.com/parking/pay/rd.php?id=10	84.200.4.125	125.0-255.4.200.84.in-addr.arpa.	Ransom, Fake.PCN, Malspam	-	31400	0 DE
2253	2017/05/01_16:22	amazon-sicherheit.kunden-ueberpruefung.xyz	185.61.138.74	hosted-by.blazingfast.io.	phishing	-	49349	0 UA
2254	2017/06/02_08:38	sarahdaniella.com/swift/SWIFT%20$.pdf.ace	63.247.140.224	coriandertest.hmdnsgroup.com.	trojan	-	19271	0 US
2255	2017/12/04_18:50	textspeier.de	104.27.163.228	-	phishing/fraud	-	13335	0 US
2256	2017/10/26_13:48	photoscape.ch/Setup.exe	31.148.219.11	knigazdorovya.com.	trojan	-	14576	0 CZ

Figure 3.12: Malware Domain List Sample

In order to make use of the Malware Domain List as a filter in the Logstash, we configure the .conf file of Logstash to use the CSV file as a filter. The CSV file is converted into a YAML file, in order to be best suitable for Logstash configuration, the YAML file includes the same information as the CSV file, most importantly the information about malicious IPs and domains. The re-formatting process from the CSV file into the YAML file is done by a python script we wrote. It basically convert each CSV column into a key-value pair of a dictionary, and then output the dictionary in a YAML format. Also, it is

worthy to mention that we use the translate plugin of Logstash to read the information from the YANML file, and if there is a coming request sent by a malicious IP or Botnet IP, we would tag this request as malicious and stored it in Elasticsearch index, which could be visualized in Kibana to notify any user or administrator in front of Kibana endpoint. The YAML file of Malware Domain List, in this case, could be seen as a dictionary file, every request sent by clients would be checked based on this dictionary file. The plug-ins we have used to configured Logstash as shown in the following Table 3.2. A simple test could be made to check whether the threat data feed is read

Plug-in	Description
beats	receive events from the Elastic Beats framework
grok	parse unstructured log to be structured and queryable
mutate	perform general mutations on fields
date	parse dates from fields
geoip	add information about the geographical location of IPs
useragent	add information about user agent like operating system
stdout	a simple output which prints to the STDOUT
elasticsearch	output log data into Elasticsearch
translate	a general search and replace tool

Table 3.2: Logstash Configuration 2

successfully, we config the Logstash to receive std as input, and std to be output. And then we send a JSON message to Logstash, which including a known malicious URL from the list. The output from std would indicate this request would be treated as a malicious request and add a tag as malicious IP or URL, which can be visualized in Kibana. As shown in Figure3.13.

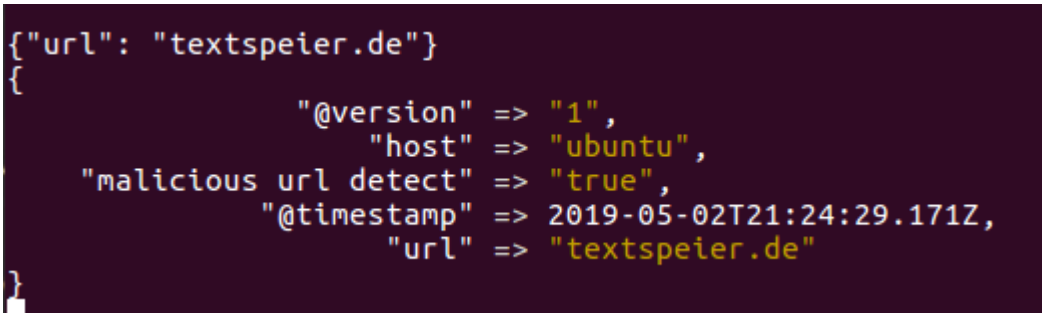

Figure 3.13: Testing on Malicous URL

After the log data is indexed into Elasticsearch, we create an index pattern for the data using Kibana. As shown in Figure3.14, the Kibana uses time-frame as a filter to visualize the log data. The number of requests of our server for each day can be seen clearly.

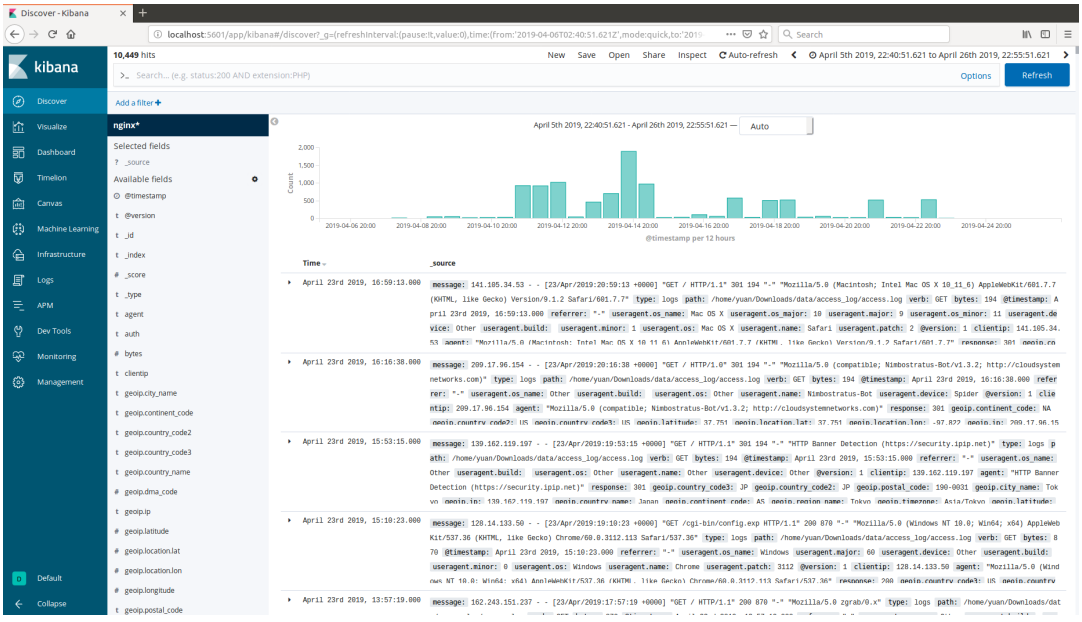

Figure 3.14: Data Pattern in Kibana

The data pattern indicates the information created and parsed by Logstash.

In each message, there are fields indicating the client and the request information. We focus on security-related information, as our expectation by using threat data feeds, every time a malicious IP or a botnet IP visiting our server it would be added a tag as a malicious URL, as shown in Figure3.15. This field could be used in visualizing other charts or graphs. On the other hand, the normal requests, or the requests that made by clients that not existed on the malicious and botnet IPs data feed, would not have such a malicious URL field.

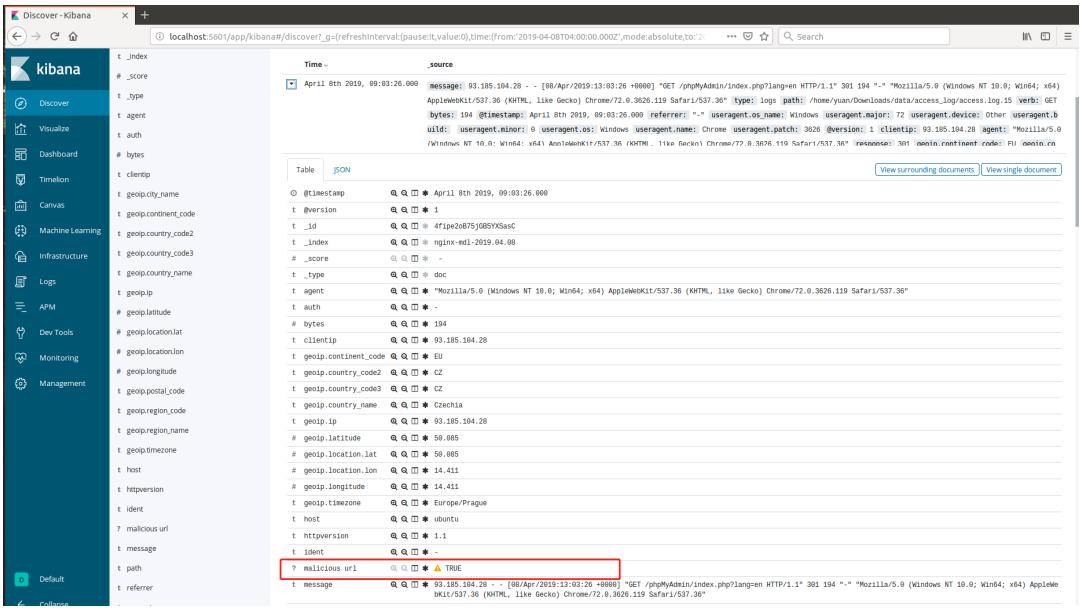

Figure 3.15: Data Pattern in Kibana

3.3.5 Crimeserver IPs Visiting

To achieve the goal that we can be able to detect any crimeserver hitting our web application server, we utilize another threat data feed: Blueliv threat data feeds. Blueliv is a cyberthreat intelligence provider, providing services cover from Botnet information, crimeservers information, to malwares and attack

patterns information. It also provides a Logstash plugin for Logstash users. Based on its official documentation, we are able to use the plugin and configure it for the altering purpose of our web application server. The Logstash plugin provides services and its description are as shown in following Table 3.3. The

Services	Type
Crimeservers	Partly Free
Botnet IPs	paid
Malwares	paid
Hacktivism	paid

Table 3.3: Plugin Services

detail of each service are:

1. The free feed only reports crimeservers from open source sites.
2. Crime servers: Malware distribution domains, C and Cs, phishing, exploit kits and backdoors, ID, type, country, domain, geolocation, ASN ID, status.
3. Bot IPs: Infected IPs, OS affected, user agent, IP address, geolocation, family type, version, status
4. Malwares: Malware hashes
5. Hacktivism: Social monitoring related to hacktivism operations. Ops/Hashtag, country, number of tweets per day, tweets.

After we configure the Logstash plugin accordingly, we get information from Logstash std output saying the threat data is read successfully, as shown in the following Figure3.16.

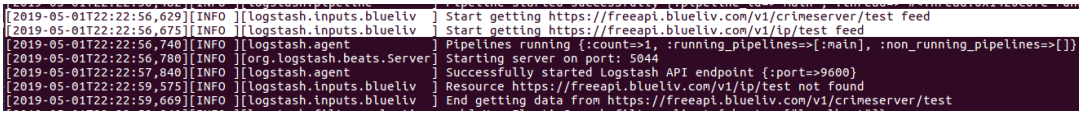

Figure 3.16: Read threat Data

3.3.6 DNS Blocklist IPs Visiting and DDOS Attack

IP DNS Blocklist checks the IP address of the sending mail server against a public list of mail servers known to send spam. The DNS blocklist is actually a list of IP addresses that can be queried. The DNS query method is used to find out whether an A record of an IP address exists to determine whether it is included in the DNS blocklist. The IP address in the list indicates that the spam has been posted externally. Therefore, it is used by the spam filter to filter spam sent by the list like a virus definition file.

Specifically, DNS Blacklist is a spam blocker that allows web administrators to block messages from specific systems that have a history of sending spams. As the name implies, these lists are Internet-based domain name systems that convert digital IP addresses (such as 66.171.248.182) into domain names like example.net, making the list easier to read, use, and search. If the maintainer of the DNS blacklist has received any type of spam from a particular domain in the past, the server will be blacklisted and all messages sent from it will be flagged or rejected. Therefore, DNSBL is a list of a large number of IP addresses that are considered to belong to a mail server that sends or forwards a large amount of spam.

In our project, we also make good use of this list, which provides us great insights about whether our web application server is under continuing visited

by any spam server. We utilize the data feed from zen.spamhaus.org, which is a database of IP addresses from which does not recommend the acceptance of electronic mail. We configure the Logstash to use the DNS filter to read data from zen.spamhaus.org, and if there is a match found, then we add an extra field to our indexed data indicating it is a spam address request. As shown in following Figure3.17.

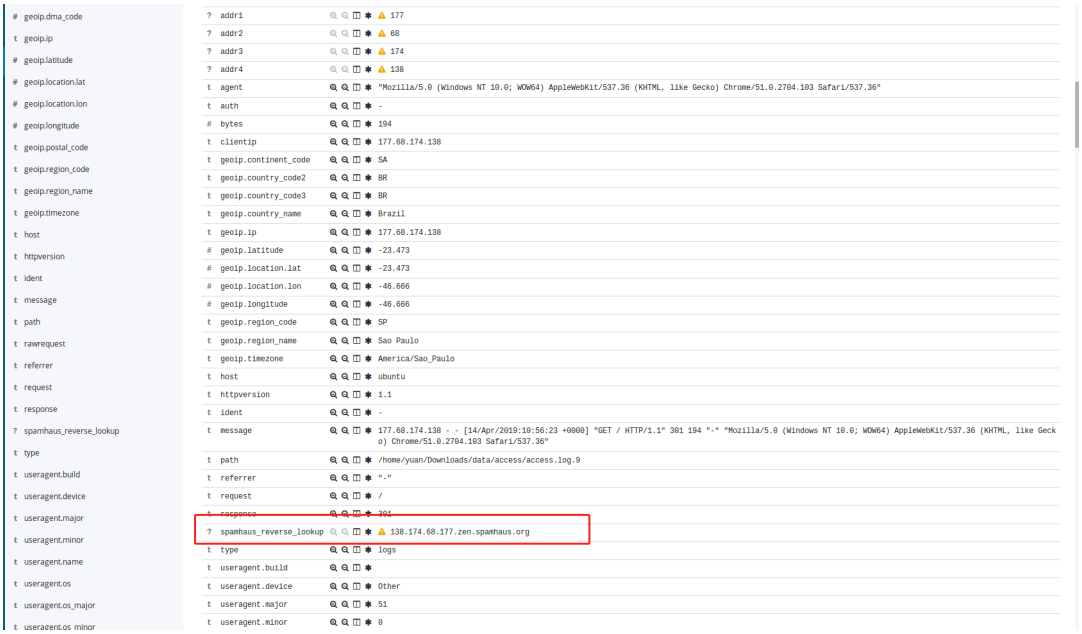

Figure 3.17: Spam Checking Field

For the configuration of the Logstash DNS filter, the details are shown as following Table 3.4. Specifically, Spamhaus defines a specific way of checking, which is the reversed IP checking. Therefore, we have to convert IP address in each message to the reversed format, and then send it to zen.spamhaus.org database to checking to see whether it is a spam or not. If the answer is yes, the zen.spamhaus.org would return a special return address, which is 127.0.0.2, to indicate there is a matching happen. And then, we add a field to our data

to be indexed.

DNS Filter Option	Value
resolve	spamhaus_reverse_lookup
nameserver	10.0.1.1
add_tag	dns_successful_lookup
action	replace

Table 3.4: Plugin Services

After we finish indexing data into Elasticsearch, we can set up some limits and alerts in Kibana to indicate some abnormal situations happening, such as servers under the DDOS attack. Kibana provides a lot of methods and options for users to create a threshold alert. We are able to utilize these functions to detect a potential DDOS attack happening, and the information would be sent to the email address we provided in Kibana.

3.4 Results and Analysis

By using Kibana, we can create and customize dashboards to visualize our data. As we gathering threat data such as crimeservers data from third parties information provider, we can also visualize that data using Kibana, as shown in the following Figure3.18. In the dashboard, we customize it to show the map of crime events, the status of crime events, and the types of crime servers, which are all can be seen clearly in our dashboard.

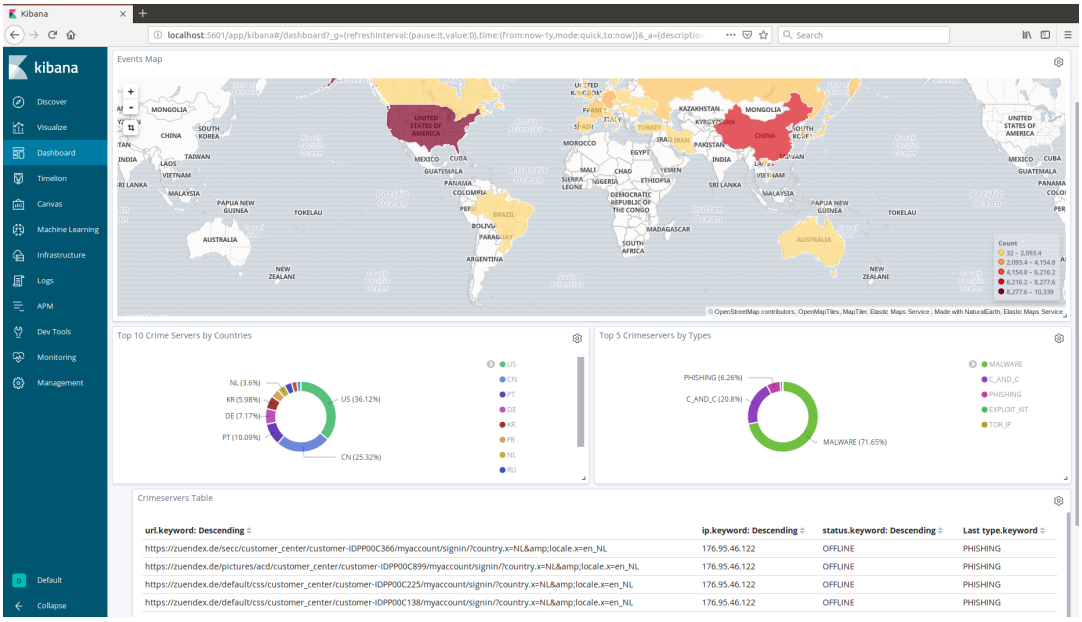

Figure 3.18: Crimeservers Visualization

As the configuration we set-up for ELK stack components, we are able to visualize our access and error log in Kibana. Based on the access logs and error logs we collected and indexed, we can visualization these data and conduct security analytics on it, as shown in the following Figure3.19. As it is shown in the figure, the crimeservers that hitting our web application server are all captured and visualize in our customized dashboard. Based on this information, web application server administrators can be aware of the status of the server they controlled and taking any necessary actions against potential attacks or intrusion.

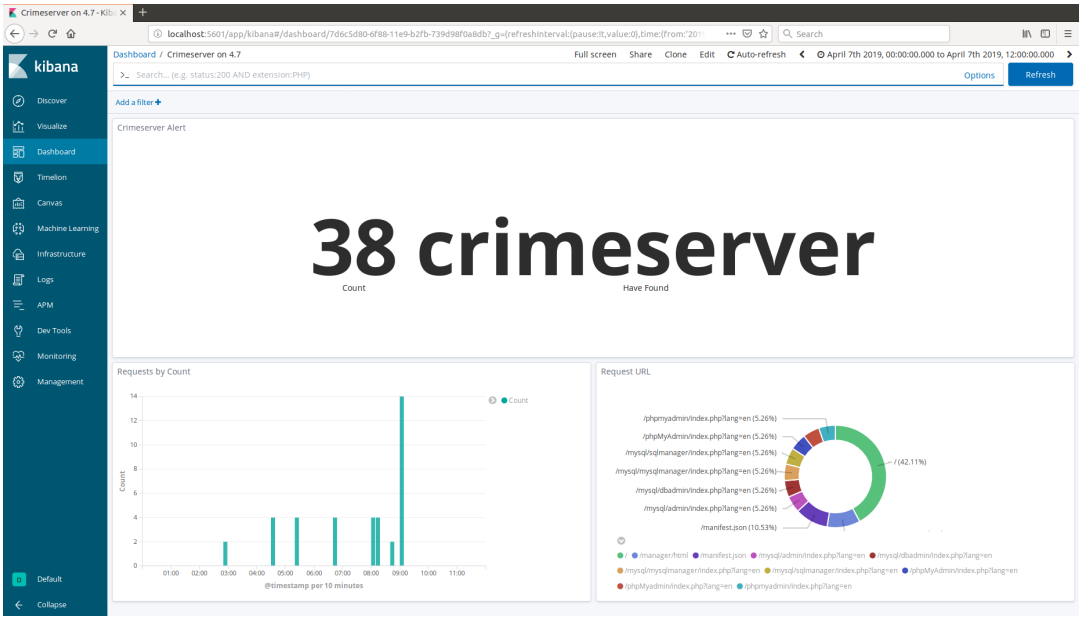

Figure 3.19: Crimeservers Visualization 2

We can also conduct searching in Kibana, as we configuring our Logstash, we are able to tag any malicious IP or Botnet IP that are visiting or have visited our server, these requests would be marked as malicious URL visitings. As shown in following Figure3.20, we can list all the matched results that have the tag, which provides administrators a lot of valuable information regarding these potential intrusions and allowing them to conduct further research on the information.

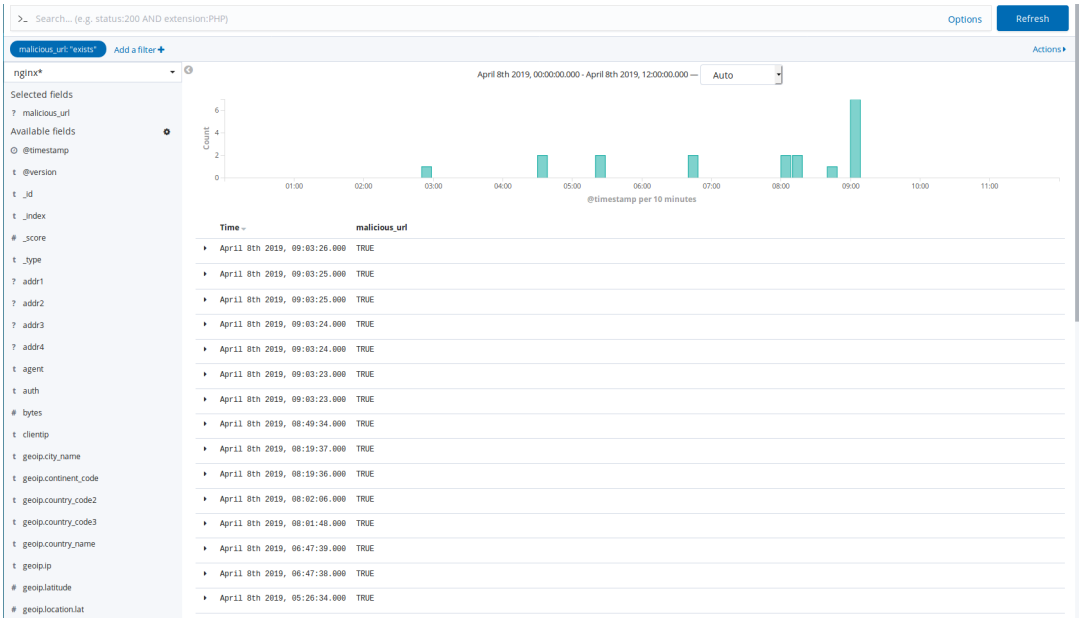

Figure 3.20: Malicious URLs Searching

Chapter 4

Summary

4.1 Contributions

Generally speaking, this project provides a perspective regarding utilizing ELK Stacks to conduct log analytics and security analytics. It is a relatively new solution but the applications of ELK stack becoming more and more populous. It is obvious to see that in the future Elk stack and its components will keep playing an important role in security analytics, especially when it comes to the big data environment. It is happy to see such a powerful tool could be made use of in many different use cases. In our project, specifically, we complete several tasks as shown in following.

1. Log management and Analytics

   The server Log files are analyzed and visualized in a clear format and we customize multiple dashboards to present as many as insights that the server log files can provide. This helps web application administrators, security engineers, or devOpp engineers gathering helpful information for troubleshooting, knowing the server better and conduct the best way

to manage and optimize their servers.

2. Security Analytics

   Thanks for the idea of sharing within cybersecurity community, we are able to make use of multiple free open source threat data sources that enables us to conduct different security analytics and gather different insights of our data. Based on this information, a responsive team can step in and conduct any necessary actions to make sure the server is under control.

3. Elk stack functionality

   We also explore multiple components and try different functions of Elk stack, especially Logstash and Elasticseach, we believe these powerful tools can be utilized in many more use cases in the future.

4.2 Future Work

Due to the limited of time and resources, there are still many limitations in our capstone projects, and we can have further research on the following directions:

1. More Types of Attacks

   DDOS is a widely used attack used by attackers, so we pay most of our attention to it. In fact, there are many other types of attacks such as Smurf or Trojans in the realistic scenario. In future work, we can try to make our system compatible with more types of attacks. In addition to the types of attacks, giving out more analysis reports and countermeasures

against different threats is meaningful.

2. Apply Machine Learning Models

   Currently, most log analysis and detection systems are based on misuse detection. The researcher has integrated the experience and knowledge of the safety experts and has carried out many types of research and enhancements.

   Traditional log analysis systems are detecting known or manually identifiable attacks. However, once an attacker slightly modifies some known malware, it poses a major challenge because the above detection methods will fail. If the attacker makes a large modification and changes the feature identifier of the malware, the method of detecting by using the identifier does not detect any abnormal content. At this point, machine learning can take advantage of it, by learning the existing data and building models to predict unknown data, so as to conduct security analysis and predict the newly generated logs.

3. Larger scale of data

   Compare with the scale of data used in industry, the data scale we used is really little. So Another direction is to make our project support much larger data scale. Perhaps it will be really difficult to process such a huge amount of data on off-line systems, we can turn to AWS or Azure for help. Taking advantage of the clouds, our project could handle more data and more practical scenarios.

References

Clinton Gormley, Zachary Tong (2014). "Elasticsearch:the definitive guide". In: *O'Reilly Media*.

https://www.elastic.co/cn/products/. "Elastic Website". In:

Dean J., Ghemawat S (2008). "Mapreduce: simplified data processing on large clusters". In: *Commun. ACM*.

Lu Siyang, Wei Xiang Rao Bingbing (2019). "LADRA: Log-based abnormal task detection and root-cause analysis in big data processing with Spark". In: *Future Generation Computer Systems*.

Jayathilaka H. Krintz C., Wolski R. (2017). "Performance monitoring and root cause analysis for cloud-hosted web applications". In: *Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee*.

Siregar Irdian; Niu, Yufu (2017). "Coal ash content estimation using fuzzy curves and ensemble neural networks for well log analysis". In: *International Journal of Coal Geology*.

A.Abdulraheem E.Sabakhy, M.Ahmed A.Vantala P.D.Raharja G.Korvin (2007). "Estimation of permeability from wireline logs in a middle eastern carbonate reservoir using fuzzy logic". In: *Society of Petroleum Engineers*.

Chen Fumei Han Dezhi, Bi Kun (2017). "Research on the Key technologies of distributed data stream processing system based on big data". In: *Journal of Computer Applications*.

Liao Xiangke Li Shanshan, Dong We (2016). "Survey on log research of large scale software s ystem". In: *Journal of Softwar*.

Zhao Yining, Xiao Hail (2017). "Optimization of the log pattern extraction algorithm for large scale syslog files". In: *Computer Engineering and Science*.

Bai Ju, Guo Hebin (2014). "The design of software integration for big log data real time search based on ElasticSearch". In: *Journal of Computer Applications*.

Merve Astekin Harun Zengin, Hasan SÃűzer (2018). “DILAF: A framework for distributed analysis of largeâĂŘscale system logs for anomaly detection”. In: *Software Engineering in Practice*.

Xu W. Huang L., Fox A. Patterson D. Jordan M.I. (2009). “Detecting large-scale system problems by mining console logs”. In: *SOSP, ACM*.

Gu X., Wang H. (2009). “Online anomaly prediction for robust cluster systems”. In: *Data Engineering, 2009 ICDEâĂŹ09 IEEE 25th International Conference on, IEEE*.

Zaharia M. Chowdhury M., Das T. Dave A. Ma J. McCauley M. Franklin M.J. Shenker S. Stoica I. (2012). “Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing”. In: *NSDI, USENIX Association*.