

# Server Log Analysis Using ELK Stack

Ms. Sanobar Sultana Shaikh  
Department of IT  
Thadomal Shahani Engineering College  
Mumbai, India  
sanobar.shaikh@thadomal.org

Sushant Sawant  
Department of IT  
Thadomal Shahani Engineering College  
Mumbai, India  
sushantsawant865@gmail.com

Pranav Tella  
Department of IT  
Thadomal Shahani Engineering College  
Mumbai, India  
tellapranav@gmail.com

Anuj Vyas  
Department of IT  
Thadomal Shahani Engineering College  
Mumbai, India  
anujvyas.work@gmail.com

**Abstract** — *In real life there are lots of network operations, security management issues, and lots of users. That's why we have a lot of log files like Netflow, Syslog, server access log, service logs, and audit logs. 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. With the help of the ELK stack, we can aggregate logs from all our systems, analyze these logs, and create visualizations for application and infrastructure monitoring, faster troubleshooting, security analytics, and more. We are aiming to analyze intrusion from the logs, and our system can find out the attacks in real-time or from the logs stored offline.*

**Keywords** — Log Analysis, ELK Stack, Beats

## I. Introduction

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 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.

The goal of this paper is to give an idea of how we can conduct security-related log analytics on web server access logs, windows event logs, network log files, and how we can detect any threat attacking our web application with the help of ELK stack components. Also how we can design the whole system and configure ELK stack components based on our use-case and requirements.

## II. ELK Stack and its Security Aspects

### A. ELK Stack Overview

The most urgent task is to use a centralized log management platform to collect and collect logs from all servers for monitoring and analysis. ELK Stack is a real-time data analysis framework that has certain advantages for log stream processing, and other auxiliary analysis programs are still needed for the analysis requirements of specific systems or environments.

ELK Stack consists of the following 4 main components:

#### 1. Elasticsearch:

Elasticsearch is a No-SQL database that is based on the Lucene search engine. It is not similar to any relational database. It has its way to store data, indexing data for further queries, and the most important aggregation and analytics on data [1].

#### 2. Logstash:

Logstash is a pipeline tool that accepts inputs from various sources, executes different transformations, and exports the data to various targets. The main functionality of Logstash is to collect and parse log files.

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 [1].

#### 3. Kibana:

Kibana is used as the user interface of Elasticsearch and the visualization tool for our input data. It provides a configurable user-friendly interface for visualization data that is stored and indexed in Elasticsearch [1].

Kibana is popularly used with Elasticsearch because it is a browser-based user interface along with its capabilities to perform much more things apart from visualization like alerting, monitoring, anomaly detection, etc.

#### 4. Beats:

The shippers/beats are also Elk Stack components. Elasticsearch provides a lot of different kinds of shippers called Beats for different use cases, for example, Filebeat, Winlogbeat, Packetbeat, Metricbeat, Auditbeat, Heartbeat.

Beats are easy to use and computationally cheaper as compared to Logstash and beats can directly send data to Logstash or Elasticsearch [1].

In this paper along with the ELK stack we have used three different beats which are:

Filebeat: To read the open-source apache log files and send them to the Elasticsearch

Packetbeat: To gain insights into network-related data and to monitor the network protocols and other security-related aspects.

Winlogbeat: It is a beat specifically built for windows and it is capable of sending all the different events related to the system, security, and windows event logs.

### B. Security Aspects of ELK Stack

For security use cases, since the log data files we used are mostly web application server access logs, even though these

data are simple, they still can provide a ton of security-related insights of the web applications and servers. 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.

By utilizing these free threat data feeds combined with ELK stack analytics functionality, we can gather security-related insights from our web application server access logs.

Winlogbeat and Packetbeat also provide a lot of security-related aspects like with the help of Winlogbeat we can see if any critical error events are generated or not also we can see the top events of our system. Packetbeat can help us to gain insights into server locations that we are visiting and also information about HTTP, TLS, and DNS protocols.

### C. Dataset Description

In this paper, to detect the DDOS attacks, SPAM requests, and other intrusions, we use the following datasets to conduct our experiments:

#### 1. Blueliv Dataset:

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 from enterprises. Depending on this dataset, we can 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 that 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]

#### 2. Malware 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 if anyone in our organization tries 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 (Spamhaus Block List), XBL (Exploits Block List), and PBL (Policy

Block List). In our paper, we use ZEN, which is the collection of the above three, that is, the data including XBL, SBL, and PBL. [3]

## V. Design and Implementation

We have stored the Apache log files on our system. To process these log files we have used Filebeat; Filebeat will gather the logs stored in the directory and the path for that directory will be given in the "filebeat.conf" file and Filebeat will give output to the Logstash and then Logstash will perform the data enrichment using Logstash plugins and filters and store the log files into the Elasticsearch so that log data can be visualized in the Kibana.

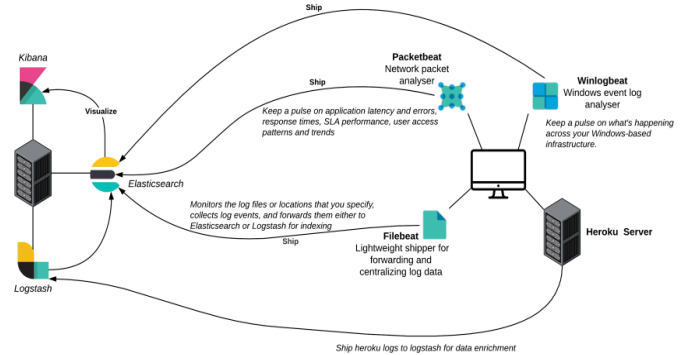


Figure 1: Design details

Based on the above design, we use the Heroku server as a host to deploy our web application server. We can access the logs of our Node.js server through the Heroku dashboard and we can directly download the logs to our system and then as soon as logs files get downloaded; Logstash will continuously monitor the downloads directory of our system as soon as the file gets downloaded it will apply all the specified filters on the data to get insights of the data and after performing the data enrichment it passes the output to the Elasticsearch.

To check any crime servers hitting our web server we have used Blueliv's API to gather this crime server's data. We have written a .conf file which we will run periodically to get the updated data and all the crime server data is stored in the different database named 'Blueliv'. If any match happens between IP Addresses hitting our website and the IP address of the crime server we will add the tag of "crimeserver" in that record. [2]

To check any malicious IP/URL visiting our web servers we have taken a CSV file and for compatibility, with Logstash we have converted the CSV file to a YAML file. ThisYAML file is stored in the same directory in which our conf file is stored. If the file is stored somewhere in a different location we have to specify the complete path in the translate plugin of the Logstash which is written in the conf file. If any match happens between IP Addresses hitting our website and the IP Address / URL present in the Malware Domain List we will add the tag of "malicious url" in that record.

For real-time monitoring of our system(windows system) and network data, we have set up Winlogbeat and Packetbeat. These beats will directly send the data to Elasticsearch. We have to do a small enrichment for the data in the Packetbeat for that we have used a GEO-IP processor of Logstash.

## A. Log Analytics on open-source server log files

Since the goal of this project is to mainly focus on utilizing the ELK Stack framework for log management and security analytics, the best way to get started is to make 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, 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.

For our project, we have used the open-source apache log files which have the following log format and consist of the following information.

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 an identifier such as a computer's hostname or browser's version

## ELK Stack Configuration:

We use Filebeat as a log files collector and a lightweight 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 Filebeat 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 analysis. We have used the following plugins for our requirements.

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

Figure 2: Logstash Plugins

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.

## B. Security Analytics on Heroku Web Server Log

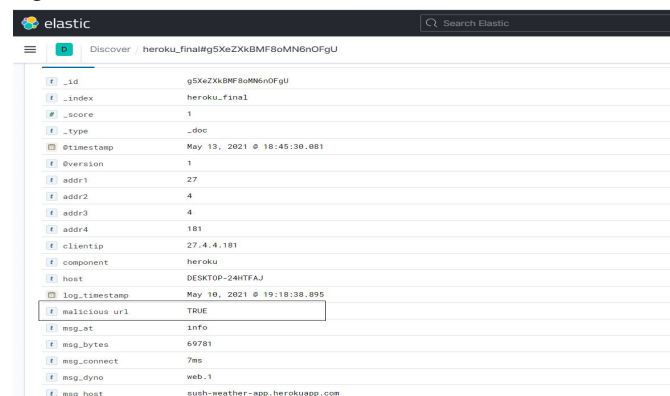
To simulate the real-life production environment and to explore how ELK Stack can be utilized in the real-life production environment, we have built a real web application server and deployed it on Heroku server to let it run all the time to collect server log data. These data are streamed to an Elasticsearch endpoint to be indexed and eventually being visualized and analyzed. We built a lightweight node.js web application based on a Node.js project and to receive static requests. After we had set up the node.js application and successfully deployed it on the Heroku server; We could

access it by using the domain to send the HTTPS requests the client would receive a response.

## 1. Malicious And Botnets IP's Visiting

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.

If there is a request sent by a malicious IP or Botnet IP, we would tag this request as malicious and store it in the Elasticsearch index, which could be visualized in Kibana to notify any user or administrator in front of the 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. every time a malicious IP or a botnet IP visits our server it would be added a tag as a malicious URL, as shown in the following Figure



The screenshot shows the Elastic Kibana interface with a search bar at the top. Below the search bar, a table of log entries is displayed. The entry shown has the following fields: \_id (g5XzXbWf8oMn6nOfgu), \_index (heroku\_final), \_score (1), \_type (\_doc), @timestamp (May 13, 2021 @ 18:45:38.881), @version (1), addr1 (27), addr2 (4), addr3 (4), addr4 (181), clientip (27.4.4.181), component (heroku), host (DESKTOP-24HTFAJ), log\_timestamp (May 10, 2021 @ 19:18:38.895), malicious\_url (TRUE), msg\_et (info), msg\_bytes (69781), msg\_connect (7ms), msg\_dyno (web.1), and msg\_host (sush-weather-app.herokuapp.com).

Figure 3: Malicious URL field in data

This field could be used in visualizing other charts or graphs. On the other hand, the normal requests, or the requests made by clients that did not exist on the malicious and botnet IPs data feed, would not have such a malicious URL field.

## 2. Crime Server IPs Visiting

To achieve the goal that we will be able to detect any crime server hitting our web application server, we utilize another threat data feed: Blueliv threat data feeds. Blueliv is a cyber threat intelligence provider, providing services from Botnet information, crime servers information, to malware and attack patterns information. It also provides a Logstash plugin for Logstash users. Based on its official documentation, we can use the plugin and configure it for the altering purpose of our web application server. The Logstash plugin provides many services. For our paper we have used crime servers that were free to use and provide the following services:

- a. The free feed only reports crime servers from open source sites.
- b. Crime servers: Malware distribution domains, C and Cs, phishing, exploit kits and backdoors, ID, type, country, domain, geolocation, ASN ID, status. [2]

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 Figure

Figure 4: Reading crime server threat data from Blueliv

Figure 8: Packetbeat output to ElasticSearch



## IV. Result and Observation

### 1. Apache open-source server log files

Based on the configuration of Logstash and Filebeat, log files are parsed and analyzed and then indexed into Elasticsearch for analytics and aggregation. We utilize the Kibana as

Based on the enriched information of the GEO-IP 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. The user interface to communicate with Elasticsearch. is information is presented directly to the user in the form of a histogram, allowing the user to clearly understand the useful information, as shown

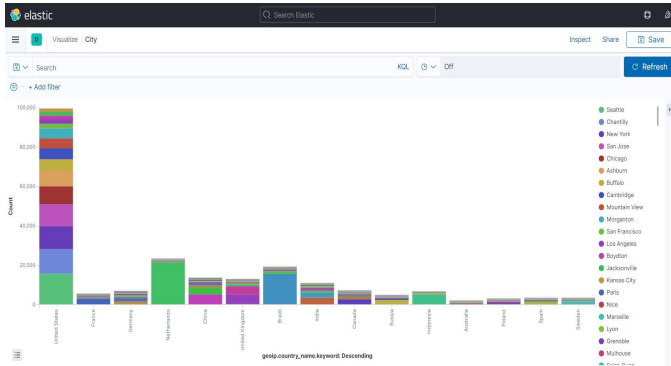


Figure 9: Visualization of cities on Apache logs

Furthermore, we are able to analyze the historical graph indicating the day and the time that has the most clients visiting the web server. All the charts and graphs could be arranged to be one dashboard and can be customized to fit the use cases and requirements.

### 2. Security related aspects of Node.js web server

We have performed the analysis on the access log files of our web server, The results of Node.js including data enrichment from the datasets that we have used we got different additional fields in our index like malicious URL, spam address, and crime server. Instead of checking index records to get all the info at one glance, we have created a single dashboard that contains all information. The dashboard is as shown below

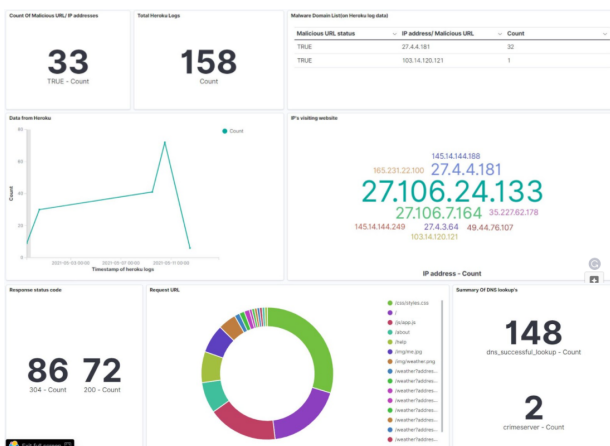


Figure 10: Security analysis on Heroku data

### 3. Dashboard created from Blueliv API

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

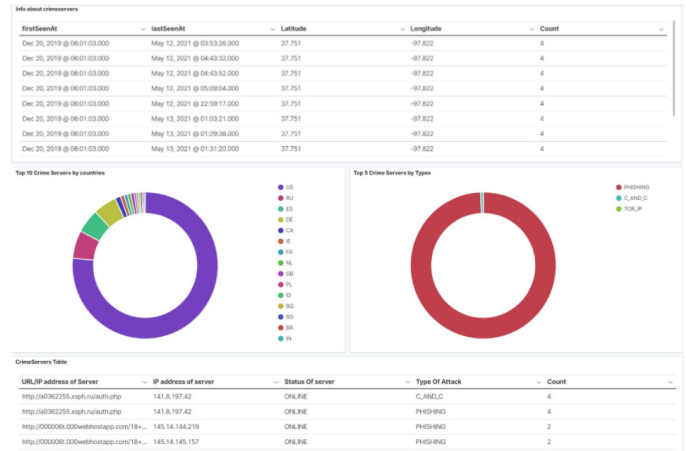


Figure 11: Crimeserver dashboard from Blueliv API

### 4. Insights from Packetbeat and Winlogbeat

After successfully setting up the Packetbeat with ElasticSearch and Kibana, we were able to gather different 1058 fields in the ElasticSearch and we built visualization using these fields. At this time we have around 75903+ unique documents which contain the data about the network flows and different events collected from the Packetbeat and stored in the ElasticSearch database. Similarly from Winlogbeat, we were able to gather 1066 fields and 161488+ documents. Some of the visualizations that we have created using data from beats are shown below,

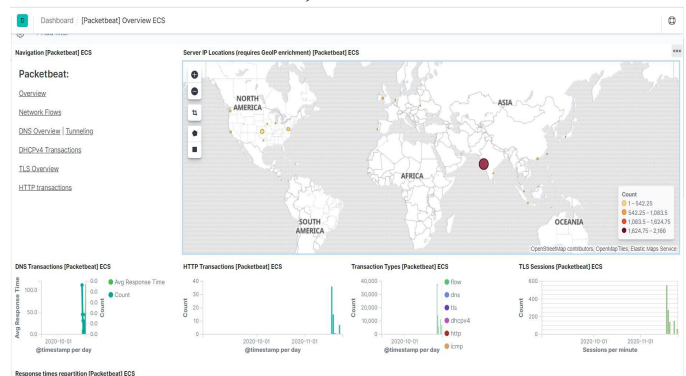


Figure 12: Packetbeat dashboard

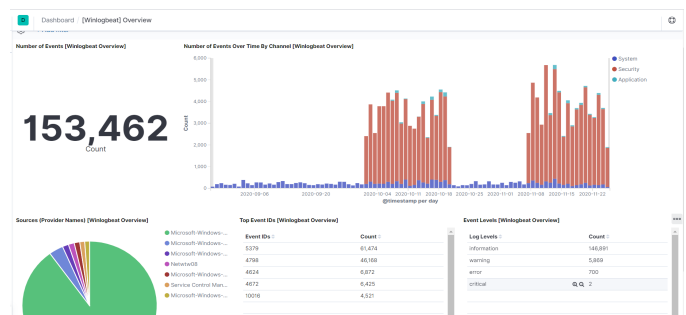


Figure 13: Winlogbeat dashboard

## **V. Conclusion**

In this paper, specifically, we complete several tasks as follows:

### **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 DevOps engineers to gather helpful information for troubleshooting, knowing the server better, and conducting the best way to manage and optimize their servers.

### **2. Security Analytics:**

We are able to make use of multiple free open source threat data sources that enable us to conduct different security analytics and gather different insights into 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 the ELK stack, especially Logstash and Elasticsearch, we believe these powerful tools can be utilized in many more use cases in the future.

## **VI. References**

[1] Elastic Stack Documentation

(<https://www.elastic.co/guide/index.html>)

[2] ELK Config Examples

(<https://github.com/Blueliv/elk-config-examples/blob/master/documentation.pdf>)

[3] DNS Blocklist Basics - Spamhaus Technology

(<https://www.spamhaus.com/resource-center/dns-blocklist-basics/>)

[4] Configure Winlogbeat - Elastic

(<https://www.elastic.co/guide/en/beats/winlogbeat/current/configuring-howto-winlogbeat.html>)

[5] Configure Packetbeat - Elastic

(<https://www.elastic.co/guide/en/beats/packetbeat/current/configuring-howto-packetbeat.html>)