Use of the ELK stack in Conjunction with GROK filtering as a Cyber Security Application on Network Analytics and Monitoring

Cayler Miley
Department of Computer Science and Engineering, University of Nevada, Reno

Under Supervision of: Jeff Springer
January 18, 2016

Science Technology Engineering & Mathematics Bilingual Online Peer Sessions (STEM BOPS)
Supported by NSF Grant #IIA-1301726

Introduction and Summary

Elastic™ offers a cluster of applications that coordinate in order to analyze the contents of logs, index logs, and provide a graphical user interface (GUI) for viewing data. These applications are collectively called the ELK stack (Elasticsearch, Logstash, Kibana) for the names of the programs involved in the cluster. Contrary to the acronym ELK, the cluster receives data first through Logstash, then sends it to Elasticsearch, then to Kibana rather than the implied order: Elasticsearch, Logstash, and Kibana (see figure 1). Logstash is used as a parsing tool to generate readable output from device and software logs. Elasticsearch receives the output information of Logstash and creates an index that can be queried and searched. Kibana creates a GUI for analyzing the information provided by Elasticsearch. Kibana can also be customized to create graphs and charts for viewing large scale data.

In conjunction with the ELK stack, a filtering tool can increase the usability of the ELK cluster by allowing the user to parse logs in a more organized fashion. Logstash’s parsing mechanisms are insufficient for organizing the volume and breadth of data needed for analyzing a network because it cannot reliably parse every source automatically without extensive manual input. To alleviate this problem, GROK is used as a supplement to the ELK stack for organization and guaranteeing correct output for log parsing.

The motivation for using these programs was to form a monitoring tool that functioned as close to real-time as possible for the network at the University of Nevada, Reno (UNR). The ELK stack organizes metadata, produces an indexed and searchable database of metadata, and creates visualizations to find trends in network traffic. The data visualization and ability to query make it ideal for cyber security analytics as analysts can search for potential threats across multiple log sources and find intervals where attacks are occurring more or less often on the network.

Background Information

Logstash’s output and the input data for Elasticsearch are determined by extracting information with the use of a template that standardize the fields contained in logs. Logstash uses a template for each log source and parses logs according to the individualized template. The purpose of Logstash is to make logs that are difficult to read appear in a format that allows viewing of log information in a more reasonable fashion. GROK filtering simplifies the code for Logstash parsing by using regular expressions to abstractly define fields included in logs such as timestamp, IP address, DNS server, etc. The output of Logstash is stored as a JSON object (a
JavaScript data structure for organizing connected fields) with field names and field values from the log sources. Elasticsearch receives the Logstash output and indexes it by source. Lastly Kibana queries Elasticsearch to create visualizations and graphs for viewing trends in log data, allows a user to search for data either on a single source or across multiple sources, and presents each individual log in table format.

The ELK stack documentation portrays ELK as a business metrics system that allows IT departments to collect data and give data metrics to marketing or sales departments in determining statistics such as website traffic and volume, or the number of users active on the local network during office hours [Agitare Technologies Inc., 2014]. In this report however, cyber security and network monitoring are the primary goals of use of the ELK stack. The ELK stack was chosen for its horizontal scalability, use of JSON structures, and ability to import logs as soon as they are created.

Problem Analysis

The problems in monitoring a large network become more pronounced as the complexity of the network and the number of users increase. In an environment with many different computers (windows machines, Apple products, Linux machines, and legacy machines), compilation of data into a steady and reliable stream of information is extremely important. It is necessary to compile relevant information from various sources into a readable and useful format for statistical or graphical analysis. In this sense, Logstash allows for compilation of all logs (from each source) into a common format that is useful for analysis.

Once the log information is compiled into a database, making the data readily accessible and organized decreases the response time for investigations. As the data cluster increases in size, exponentially based on the number of log sources, a database that is indexed and easily searchable is required for investigations and maintaining multiple investigations simultaneously.

Lastly, making use of the data in a meaningful way can be difficult to implement. Sorting through data from multiple sources to link events such as location, or to create a timeline of a network attack is cumbersome based on simply reading raw logs. To follow up on an event, all of the logs pertaining to the instance (including those from different sources) need to be condensed into a form that is readable and can be used as a timeline as the investigation is ongoing.

The ELK stack in conjunction with GROK and JSON offer a solution to each of these difficulties in monitoring a network. At UNR, the ELK stack utilized with GROK and JSON was used as a satisfactory and temporary solution to fix many of the smaller problems presented in implementing a real-time monitoring solution.

Solution

First, the ELK stack can solve all of the problems iterated in the Problem Analysis section in a relatively simple fashion. Logs from all sources on the network can provide metadata for passage to the ELK stack. Each source is filtered through a discrete GROK filter (created by someone familiar with the log source) which then passes the information to Logstash where it is organized into a JSON structure. Elasticsearch then indexes the data based on the JSON structures automatically for the user based on the GROK filter. Kibana queries Elasticsearch for data necessary for visualizations.

Second, Kibana’s interface allows for filtering of data based on a search query. Kibana reads the JSON structures to present data after querying Elasticsearch. By indexing multiple logs from many different sources, one can search for useful information such as an IP address across
multiple log sources and create a custom visualization to display the necessary information. The visualizations can be as simple as a graph showing website hits each day or a global map showing ping requests based on GeoPoint software included with the ELK stack. This allows for readability and structure for metadata that without the ELK stack is often long strings of seemingly arbitrary characters. Kibana’s interface improves the analytics of monitoring by allowing graphs and visualizations to be created based on data instead of simply reading a table of values.

Third, Kibana offers a solution that is close to real time as the program can be set to refresh its data at specific time intervals. Using the lowest time interval (5 seconds at the time of this writing) creates a steady stream of information that can be monitored as the network manager sees fit.

Lastly, the ELK stack is horizontally scalable (The Elastic Stack) meaning that all of the programs can increase efficiency on a theoretically infinite scale by increasing the number of servers running each ELK program instance. The key benefit of scalability mean that ELK is ideal for long term network monitoring and should not be held back by hardware constraints.

Figure 1.

![Diagram of ELK Stack](Diagram.png)
Conclusion

At the University of Nevada, Reno, the ELK stack is used as a cyber security application to monitor the network and process events that occur. Log information from servers, firewalls, and other devices on the network are sent to Logstash to be processed with GROK filters designed by the researchers working on developing efficient ways to process security events. Elasticsearch is handled as an indexing source for all of the log sources and the output for Logstash is stored in Elasticsearch as JSON objects for ease of manipulation in Kibana and other visualization applications. With Kibana, UNR monitors the locations of any source IP address that passes through the network, checks email logs for spam and phishing attacks, and other basic attacks on the network. The ELK stack offers many solutions to monitoring various types of attacks through logs, but the visualizations of Kibana are limited. More complex visualizations need to be implemented from the ground up. In the case of UNR, it was necessary for us to build our own visualization applications in addition to Kibana to better understand the volume of data on the network.

References