Kaleigh Glatfelter

Domain: Logging and Monitoring

Question 2: Challenges of Collecting Large Amounts of Log Data

What are the challenges of collecting huge amounts of log data? How do security analysts deal with them?

Collecting large amounts of log data can pose many different challenges, challenges in which security analysts have to deal with on a daily basis. One of the biggest challenges when facing a large amount of data is finding something worth noting in a sea of information. When faced with the daunting task of looking over the log files of a few, or many, users and systems, one security event could easily be overlooked by weary eyes.

After creating a virtual network, I wanted to track the daily activity that was occurring on my server to monitor any spikes in byte usage or signs of malicious activity. But when pulling log files that span several hours or days, you can get thousands of responses, which can be overwhelming.

I needed a way to see the byte usage per unique user in a visual representation, one where I could easily see a spike above the normal usage patterns. A spike in byte usage could show me that a hacker is attempting to utilize hacker tools, or it could simply show that a user is attempting to download a large packet. A way to pull the logs into a user-friendly GUI, in which I could analyze graphs and use filters to search for specific information would allow me to sort relevant data to the task at hand. To analyze large amounts of log data, a tool, such as Elasticsearch, can be utilized. Elasticsearch contains three open-sourced platforms; Elasticsearch, Logstash, Kibana, and is generally referred to as the ELK stack. Elasticsearch pulls the log files from the system, and sends it to Logstash which then translates and stores the data into like-formatted files. Kibana is the GUI for the system, pulling the files from Elasticsearch after Logstash has translated them, and giving the user a friendly platform to perform searches, sorting, and analytics.

To find the spikes in byte usage, I simply located the Unique User vs. Byte Usage table in Kibana to see a visualized version of the data in bar graph form. From there I was able to select the largest byte usage easily, and click on the bar to see even more information related to that specific log event. When clicking on that specific spike, I was able to then pull up more information related to the event, including what activity the user was performing, which in this specific case was the download of a compressed file. To see if that file was malicious, I was able to switch to a Discover dashboard in Kibana and see specific information related to the user utilizing the mass amount of byte data. I was able to determine what country the user was based in, the IP address of his machine, what URL he was accessing and the file he was trying to download. In this case, the user was attempting to download a new version of the Kibana interface to his Linux-based machine.

Kibana provided much more insight into the log files, much of which was not needed to identify the reasoning behind the spike in byte usage in this case. Because I was able to locate the URL being accessed, it was not necessary to know what country the user was based in. Knowing this information did not change my conclusion of the event.