sFlow Monitoring for Security and Reliability

Xava Grooms
University of Kentucky
xavagrooms@uky.edu

Robert Rollins
Michigan Technological University
rvrollin@mtu.edu

Collin Rumpca
Dakota State University
collin.rumpca@trojans.dsu.edu

ABSTRACT

In the past ten years, High Performance Computing (HPC) has moved far beyond the terascale performance, making petascale systems the new standard. The drastic improvement in performance has been largely unmatched with insignificant improvements in system monitoring. Thus, there is an immediate need for practical and scalable monitoring solutions to ensure the effectiveness of costly compute clusters. This project aims to explore the viability and impact of sFlow enabled switches in cluster network monitoring for security and reliability. A series of tests and exploits were performed to target specific network abnormalities on a nine-node HPC cluster. The results present web-based dashboards that can aid network administrators in improving a cluster's security and reliability.

KEYWORDS

HPC Monitoring, Security, sFlow, Splunk, Data Visualization, System Administration

1 INTRODUCTION

Typically HPC monitoring has tended to be of low priority during the design and planning stage of a cluster [1, 2]. After the system architecture has been solidified and a vendor has been chosen, system monitoring design can be decided. However, monitoring systems require ongoing flexibility and scalability throughout the lifecycle of a cluster [8]. sFlow is a network protocol that collects network traffic information using a packet sampling method [9]. Los Alamos National Laboratory (LANL) possesses switches capable of utilizing sFlow which are currently under utilized in its HPC production system. This project aims to explore the viability and impact of sFlow enabled switches on a cluster's network.

To explore sFlow for system monitoring we conducted testing on a custom network consisting of a nine-node stateless compute cluster connected via an sFlow enabled switch. The series of tests performed targeted specific network abnormalities including but not limited to, cluster boot issues and services that communicate beyond secure network boundaries. Monitoring data from testing was collected through sFlow and was analyzed through a software called Splunk. Splunk visualizes high-volume machine data within a network in both real time and historical perspectives [5]. Using Splunk, the data collected via sFlow was visualized to display network abnormalities during cluster boot and other normal network activity. By identifying abnormalities, cluster security and reliability can be improved through network monitoring with sFlow.

2 METHODS

The workflow of this research includes several steps such as cluster network setup, network anomaly testing, sFlow data collection, and data visualization.

2.1 System Setup

The nine-node cluster was accessed and statelessly booted by a single master node. The network used for testing included the aforementioned cluster on its own private subnet and an external pc with an IP address belonging to a separate subnet. All nodes consisted of Dell PowerEdge servers running the CentOS 7 operating system (OS). Cluster provisioning and configuration management were administered by the Warewulf and Ansible softwares respectively. Network interconnections included Ethernet cables connected to two Ethernet switches (Dell & Arista) and Infiniband (IB) cables connected to a Mellanox IB switch. Lastly, sFlow was enabled on the Arista switch for traffic monitoring and the data was forwarded to Splunk using Splunk Stream [6].

2.2 sFlow

sFlow captures network traffic along with its metadata such as the frame size of a packet, destination/source mac and IP addresses. The sampling method that sFlow uses has trade-offs that depend on the traffic being monitored. With monitoring services that create large amounts of network traffic, a moderately large sampling size (1:250 packets) does not diminish the ability to effectively monitor. However, when monitoring services that create small amounts of traffic such as a cluster boot, it is necessary to set the sampling rate to 1:1 in order to effectively troubleshoot bottlenecks during boot.

2.3 Abnormality Testing

Various network services were mimicked during the abnormality testing stage. In order to verify anomaly detection, first a baseline of known good traffic was created. Then, Secure Shell (SSH) and Domain Name Service (DNS) were exploited to show real world exploits against the cluster. Another, anomaly tested was Hypertext Transfer Protocol (HTTP) network traffic to show abnormal network traffic.

2.3.1 Standard Network Traffic. To simulate standard network traffic a script was created to generate network traffic at a steady rate. There is a randomness component in the script on how many requests are sent out, along with how much data is sent so that it is realistic. The types of protocols that were simulated are HTTP, SSH, and File Transfer Protocol (FTP) data. This allows for a variety of requests to come into the cluster in addition to having data leave the cluster.

2.3.2 SSH Enumeration. SSH is a protocol that allows for secure remote connection to a server. Public key authentication is a way to connect via SSH that is widely used for automation.This process works by having a cryptographic pair of keys, the private key and the public key. The public key is then configured to grant access to anyone that has the key, also known as the SSH key [10]. SSH Enumeration is an attack on a server that attempts to find valid

usernames with a brute force method of testing a list of usernames. The exploit works by sending an ssh connection request to the server with a random username and malformed public key authentication request to the SSH service [7]. This exploit is verified to work on any version of OpenSSH 7.7 and lower [4].

2.3.3 DNS Tunneling. DNS is a protocol that allows human readable URLs to be translated to an IP address. DNS tunneling is an exploit that circumvents system firewalls. DNS tunneling is not easily detectable when a trusted HTTP domain server is used. An open-source software called dnscat2 was used during the testing stage to simulate DNS tunneling traffic [3].

2.3.4 Mass HTTP Data. HTTP is a protocol for sending data with a specific format typically through port 80 or 8080. Mass HTTP data was used to create an asymmetric data flow from the cluster. An Apache HTTP Web Server was configured on the cluster with several Hypertext Markup Language (HTML) files of various sizes. Next, a script was ran on the external server that sent requests for the HTML files. This simulated mass data exiting through port 80.

3 DATA ANALYSIS WITH SPLUNK

Splunk is a data analysis software that allows for straightforward visualization of network traffic data. Once normal and abnormal network traffic data was collected, several web-based dashboards were created in Splunk with simple alerts of potential network abnormalities. The resulting dashboards present system administrators with an automated data visualization tool to easily detect anomalies.

3.1 Port Traffic Dashboard

The ***port traffic dashboard*** shows internal and external network traffic with the ability to refine port specific data. This dashboard consists of several bar graphs, pie charts, and statistics in which metadata for traffic entering and exiting the network are displayed. These values include average, minimum, maximum, standard deviations, and the number of standard deviations from the average data. A positive value on the ***network traffic difference graph*** correlates to more traffic entering the cluster network and vice versa. Anomaly detection was defined as any data value greater or less than two standard deviations from the average. Port specific information can be found in two pie charts that depict the ports used for data entering and exiting the cluster. Similarly to the ***network traffic difference graph***, the ***requests sent to cluster graph*** displays only the requests sent to the cluster in five minute intervals. Port specific request data can be found in the corresponding pie charts.

3.2 Cluster Boot Dashboard

The ***cluster boot dashboard*** displays the boot process of a cluster and tracks the various stages for each node. The phases of boot that can be determined through network traffic are Dynamic Host Control Protocol (DHCP), FTP, Virtual Network Functions (VNFs), and Network Time Protocol (NTP). By selecting a node, the status of boot can be determined through the dashboard. As mentioned in Section 2.2, the sampling rate in sFlow was set to 1:1 to effectively troubleshoot issues during a cluster boot. In addition to boot phase status, this dashboard depicts the overall status of boot for each node. The ***cluster boot dashboard*** can significantly decrease the time and effort of a system administrator when troubleshooting bottlenecks and failures in a cluster boot.

4 RESULTS

In order to properly detect an anomaly there must be a baseline of known good network traffic for comparison as mentioned in Section 2.3.1. Once a baseline was formed, each anomaly was simulated and created unique results. Every anomaly that was tested was detectable by our ***Splunk dashboards*** as discussed in Section 3. Although various anomalies were identified in Splunk, the most significant results were detected by the port traffic and cluster boot dashboards as described in Sections 3.1 and 3.2, repsectively. Both the ***port traffic dashboard*** and ***cluster boot dashboard*** showed that the security and reliability of a cluster can be increased through abnormality detection via sFlow.

4.1 Security

Since most of the anomalies tested in this research came from security exploits, the ***port traffic dashboard*** aimed to aid in monitoring security on the cluster. With respect to SSH Enumeration, the dashboard was able to detect the anomaly using port specific network data. Although the attack does not generate a significant increase in network traffic data difference, it does produce a large amount of the network traffic through a specific port, 22. By refining the dashboard to display port 22 data, the anomaly can be detected.

4.2 Reliability

The ***cluster boot dashboard*** aimed to improve the reliability of a cluster by simplifying the troubleshooting process of a cluster boot for system administrators. To simulate a failed boot, the DHCP configuration file was edited with a false IP address. This false IP address caused the DHCP handshake to be incomplete and made the DHCP service status to be ended. From this dashboard a system administrator would have better insight into service and node failures root causes. Therefore, allowing for quicker troubleshooting during boot issues and thus, helping with the overall reliability of a cluster.

5 CONCLUSION

As HPC clusters scale up, effective network monitoring is both more difficult due to high traffic volume, and more critical, due to the inherent complexity of large scale networks. sFlow packet sampling offers a solution to the mass amount of data generated. Web-based dashboards created using Splunk allow for real-time and historical anomaly detection for a cluster network. The ***port traffic dashboard*** and ***cluster boot dashboard*** created in this research show that both the security and reliability of a cluster can be increased through anomaly detection via sFlow.

ACKNOWLEDGMENTS

This research was undertaken as part of the Computer System, Cluster and Networking Summer Institute (CSCNSI) within LANL's HPC division. We would like to extend a special thank you to the CSCNSI program leads

Catherine Hinton and Reid Priedhorsky, as well as our project mentors Michael Mason, Marc Santoro, and Nicholas Jones. We would also like to thank the instructor of the CSCNSI bootcamp Lowell Wofford as well as the entire HPC division at LANL. This document was approved for release LA-UR-19-27103.

REFERENCES

[1] V. Ahlgren, S. Andersson, J. Brandt, N. Cardo, S. Chunduri, J. Enos, P. Fields, A. Gentile, R. Gerber, M. Gienger, J. Greenseid, A. Greiner, B. Hadri, Y. He, D. Hoppe, U. Kaila, K. Kelly, M. Klein, A. Kristiansen, S. Leak, M. Mason, K. Pedretti, J. Piccinali, J. Repik, J. Rogers, S. Salminen, M. Showerman, C. Whitney, and J. Williams. 2018. Large-Scale System Monitoring Experiences and Recommendations. In *2018 IEEE International Conference on Cluster Computing (CLUSTER)*. 532–542. https://doi.org/10.1109/CLUSTER.2018.00069

[2] Amanda Bonnie, Mike Mason, and Daniel Illescas. 2017. Monitoring Infrastructure: The Challenges of Moving Beyond Petascale. In *2017 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 785–788.

[3] Ron Bowes. 2013. dnscat2. https://github.com/iagox86/dnscat2.

[4] Justin Gardner. 2018. Exploit: OpenSSH 7.7 - Username Enumeration. (Aug 2018). https://www.exploit-db.com/exploits/45233

[5] Splunk Inc. 20019. Splunk. https://www.splunk.com/

[6] Splunk Inc. 20019. Splunk Stream. https://docs.splunk.com/Documentation/StreamApp/7.1.3/DeployStreamApp/AboutSplunkAppforStream

[7] Brian Judd, Justin Gardner, Kyle LeDuc, Ryan Zagrodnik, and Dylan Webb. 2018. OpenSSH < 7.7 - Username Enumeration Exploit. https://www.shellntel.com/blog/openssh-user-enum

[8] Sam Sanchez, Amanda Bonnie, Graham Van Heule, Conor Robinson, Adam DeConinck, Kathleen Kelly, Quellyn Snead, and J Brandt. 2016. Design and Implementation of a Scalable HPC Monitoring System. In *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 1721–1725.

[9] sFlow.org. 2003. Traffic Monitoring using sFlow. https://sflow.org/sFlowOverview.pdf

[10] SSH.com. 2018. https://www.ssh.com/ssh/protocol/#sec-Strong-authentication-with-SSH-keys