Cray EX40 Cluster Network Intrusion Detection System

External Network Traffic Analysis of Cluster Activity

LA-UR-23-28646

Daniel L. Wild

High Performance Computing Division, Los Alamos National Laboratory / University of New Mexico, dwild@lanl.gov

David F. McGee (Advisor)

High Performance Computing Division, Los Alamos National Laboratory, dmcgee@lanl.gov

Thomas A. Areba (Advisor)

High Performance Computing Division, Los Alamos National Laboratory, tareba@lanl.gov

Implementing security configurations or tooling on High-Performance Computing clusters often negatively impacts performance [6]. Capturing and analyzing network traffic is a potential low-cost opportunity to add security without impacting a cluster's performance. In this project, a mirror port was used to capture external network traffic to and from a High-Performance Computing cluster over a three-month period. Analysis of that traffic was performed using two network intrusion detection utilities, Suricata [9] and Zeek [10]. The server and tools were configured to accommodate high-performance computing requirements and tuned for performance [2, 8].

This project successfully identified security concerns related to excessive failed Secure Shell connection attempts and use of four invalid certificates. Zeek [10] and Suricata [9] identified configuration issues through anomalous Domain Name Service queries from switches and nodes along with incorrectly routed outbound Hypertext Transfer Protocol traffic to Automatic Private Internet Protocol Addresses. Network intrusion detection tools demonstrated effectiveness in monitoring external cluster network traffic, providing insight into security and configuration issues that could potentially improve cluster performance.

Additional Keywords and Phrases: Network Intrusion Detection System, HPC, Supercomputer, Network Security, Suricata, Zeek.

1 METHOD

A 10 Gigabit port mirrored front-end traffic from a High-Performance Computing Cluster to a Red Hat Enterprise Linux 8 server for analysis. A mirror port was used instead of a network tap to reduce cost. Suricata [9] and Zeek [10] were installed by compiling from source, with extended Berkeley Packet Filters [4] to manage elephant traffic flows. Ansible [1] roles were developed for installation and configuration management. Data analysis was performed using logs parsed through Splunk [7] dashboards.

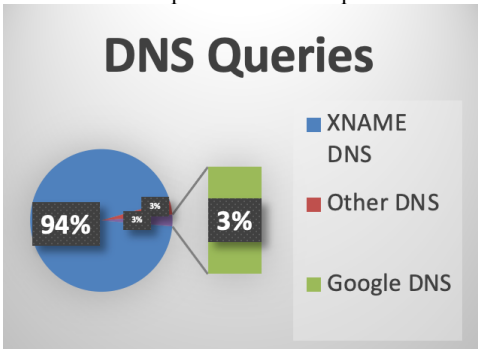
2 RESULTS

The network intrusion detection project successfully detected anomalies with Domain Name Service queries, excessive Secure Shell authentication failures, invalid Hypertext Transfer Protocol traffic, and invalid certificates.

2.1 Domain Name Service

Domain Name Service (DNS) queries represented approximately 35% of all traffic captured and 97% of those queries were anomalous. A single node within the cluster was responsible for 94% of the anomalous DNS traffic with approximately 40,000 per hour attempts to resolve non-existent component xnames (a Cray specific cluster component naming convention

[3]). These queries were not able to be resolved within the cluster, so they were forwarded to the institutional DNS server causing impact beyond the analyzed cluster. The composition of DNS queries is seen in Graph 1.

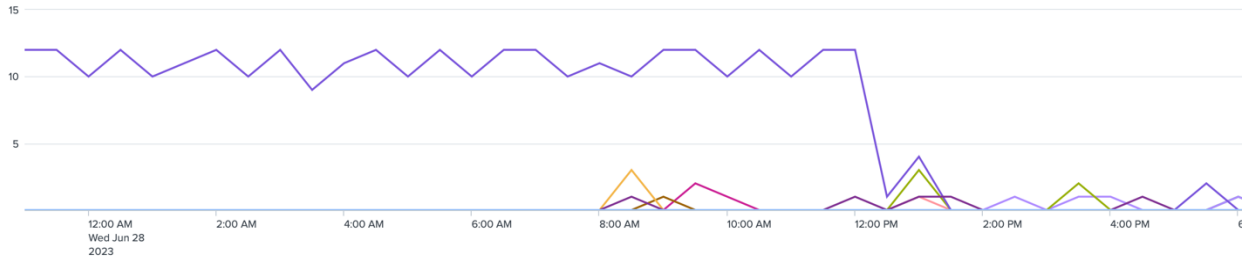


Graph 1: Domain Name Service xname anomalies and Google DNS anomalies.

The other 3% of the anomalous DNS traffic was isolated to misconfigurations on 9 switches fielded with Aruba Central Cloud enabled and Google's Domain Name Service server set as the default. Settings on those switches were subsequently adjusted and those anomalous queries are now resolved.

2.2 Secure Shell Authentication Failures

Frequent failed Secure Shell (SSH) connection attempts were detected and flagged as potential password guessing by both Suricata [9] and Zeek [10]. In one instance a single source had 1610 failed authentication attempts in a 14-hour period. These attempts were all determined to be the result of client misconfigurations by users. Authentication failures were caused by credential-caching file transfer clients with incompatible timeout intervals to current authentication configurations. In another case, a Git command was failing over to an askpass [5] client that was not configured correctly. These issues were resolved by working directly with the users and the impact can be seen in Graph 2.



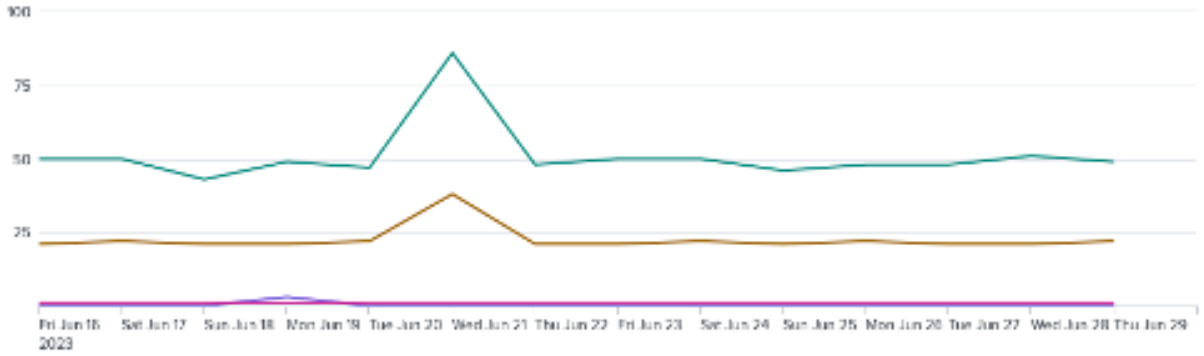
Graph 2: Secure Shell Authentication failures by IP - Resolved

2.3 Hypertext Transfer Protocol

Approximately 37% of all outbound Hypertext Transfer Protocol (HTTP) traffic was rejected by the institutional web proxy. Much of that traffic was invalid, to Automatic Private IP Addressing (APIPA) addresses. That traffic was likely intended for the localhost but instead forwarded to the web proxy due to component misconfiguration.

2.4 Invalid Certificates

Four invalid Secure Sockets Layer (SSL) certificates were identified in active use on the network. All four SSL certificates were confirmed to be intentionally self-signed. This practice may change in the future.



Graph 3: Invalid SSL Certificates

3 CONCLUSION

A network intrusion detection system provides security and insight into configuration issues which allows analysts to categorize network traffic. Categorizing network traffic to identify anomalies increases the visibility and identification of potential security events. The process of identifying anomalies as security or configuration concerns and tuning the network traffic baseline resulted in the identification of several configuration issues. Offloading and conducting network traffic analysis via an external host alleviated cluster performance impact concerns and provided relevant data on network security. The chosen low-cost mirror port architecture and open-source network intrusion detection tools (Suricata and Zeek) provided insight into the cluster's security while improving resource utilization through identifying and resolving configuration issues that were generating anomalous network activity.

ACKNOWLEDGMENTS

Los Alamos National Laboratory. LA-UR-23-28646.

REFERENCES

[1] Ansible. (2023). *Red Hat Ansible Automation Platform*. https://www.ansible.com

[2] Bainbridge, J and Maxwell, J. 2015, March 25. *Red Hat Enterprise Linux Network Performance Tuning Guide.* https://access.redhat.com/sites/default/files/attachments/20150325_network_performance_tuning.pdf

[3] Cray System Management. n.d. *Component Names (XNAMES).* https://cray-hpe.github.io/docs-csm/en-10/operations/component_names_xnames/

[4] eBPF Documentation. 2023. *What is eBPF?* https://ebpf.io/what-is-ebpf/

[5] Git-Credential-Manager-For-Windows. n.d. *Git Askpass for Windows.* https://microsoft.github.io/Git-Credential-Manager-For-Windows/Docs/Askpass.html

[6] Shah, A. 2023, January 20. *Top HPC Players Creating New Security Architecture Amid Neglect.* HPC wire: https://www.hpcwire.com/2023/01/20/top-hpc-players-creating-new-security-architecture-amid-neglect/

[7] Splunk. 2023. *Splunk.* https://www.splunk.com

[8] Suricata Docs. 2023. *High Performance Configuration.* https://docs.suricata.io/en/latest/performance/high-performance-config.html

[9] Suricata. n.d. Open Information Security Foundation (OISF): https://suricata.

[10] Zeek 2020. Retrieved from The Zeek Project: https://zeek.org