Improving the High-Performance Computing System Monitoring Infrastructure to Reliably Detect Events of Interest

Daniel Moon,[1] Jordan Webb[2]

[1]University of Tennessee Knoxville, Knoxville, TN 37996

[2]Oak Ridge National Laboratory, Oak Ridge, TN 37830

Abstract: Simple Event Correlator (SEC) is a tool used for custom monitoring of HPC systems for specific events of interest. It also allows for automated hardware triage ticket creation which minimizes system downtime.[1] To relieve stress on admin nodes, scalable leader units are used to balance the load of monitoring all the system's compute node event logs. One issue is that there is no monitoring tool to ensure that SEC is running on the leader nodes, so if SEC goes down on one of them the monitoring of its respective compute nodes is dropped. Therefore, the focus of this project is to create a monitoring tool for the SEC processes that run on the HPC systems. To start the research, I need to figure out a way to establish a connection between the leader and admin nodes to ensure that a "heartbeat" is being recorded. Therefore, a SEC calendar rule is created so that every hour each leader node will check in with the admin node to verify that SEC is operational. Once the admin node receives a "heartbeat" it calls a script that opens a Python dictionary to store a key/value pair of the leader node and the epoch time it last checked in. The script also contains a threshold check function that will determine if any leader node has not checked in with the admin node in over an hour by comparing epoch values for each node contained in the dictionary. Nagios, an event monitoring and alerting service, sends automated requests to the admin node that runs the threshold check function and returns the output to the site Nagios

dashboard that staff monitor. A Warning message is displayed on Nagios indicating which nodes are not running SEC, otherwise nodes report an "OK" status.[4]

I. INTRODUCTION

Supercomputers are capable of computing and processing data at speeds measured in floating point operations per second (FLOPS). They run jobs that perform complex calculations and simulations, typically in the field of research, artificial intelligence (AI), and big data computing. Therefore, the software and hardware management of systems is immensely important. However, when a piece of hardware such as a Graphics Processing Unit (GPU) or Central Processing Unit (CPU) malfunctions or even breaks, there needs to be some automated process that notices this and allows for some triage process that brings the issue to timely resolution. Simple Event Correlator (SEC) is one of the many tools that we use to monitor our systems. SEC is an event correlation tool that can perform highly customized event processing, which is utilized to monitor our high-performance computing (HPC) systems for event log monitoring, network and security management, and any other task that is associated with event correlation.

Having SEC running in the HPC systems enables monitoring for any hardware issues that may arise. [1] Scalable unit leader nodes are used to relieve stress and balance the load of monitoring on the admin nodes of large systems such as Frontier that have several thousand compute nodes. Each compute node produces its own individual event logs, so this can become too much volume for one server to handle. The standing issue that we had was that there was no monitoring of whether SEC was operating on our leader nodes or not. So, if the SEC daemon were to go down on one of the leader nodes, then the monitoring of its respective compute nodes would be dropped. The main idea and concentration of this project is to create a program to monitor SEC and its status throughout all the HPC systems maintained by the HPC Scalable Systems group.

II. Software Tools

A. **SEC** The event monitoring tool that we are trying to monitor on all the scalable leader unit nodes. Event correlation is a course of events that are ran to define and determine certain event groups that occurs within a specific window of time. SEC is written in Perl and does not have any platform-dependent code.[1]

B. **RegEx** Regular Expressions are useful for information extraction from items such as text, code log files, documents, etc. To use regular expressions, a sequence of characters is written to match a specific pattern in text.

C. **Nagios** An open-source monitoring solution used to run periodic checks on critical parameters of application, network, and server resources. In this project, Nagios provides a visual display of SEC's status on the scalable leader nodes. [4]

D. **Python Dictionary** A mutable data structure that allows the storing of key-value pairs.

E. **Red Hat Package Manager (RPM)** A package management system that is used for installing, updating, and managing software. It simplifies the process of software deployment by bundling dependencies, applications, and metadata into a single, standardized package format. This ensures easy installation and maintenance of software. [2,3]

F. **Gitlab** A web-based DevOps lifecycle tool that provides a Git repository manager with features such as CI/CD, issue tracking, and project management. It permits teams, such as HPC, to collaborate on code, automate the software development process, and deploy applications efficiently.

G. **Puppet** An automation solution for configuration management, compliance, Continuous Integration/Continuous Delivery (CI/CD), patch management, and more. Puppet integrates with GitLab to manage its code repository, enabling version control and team collaboration. [3]

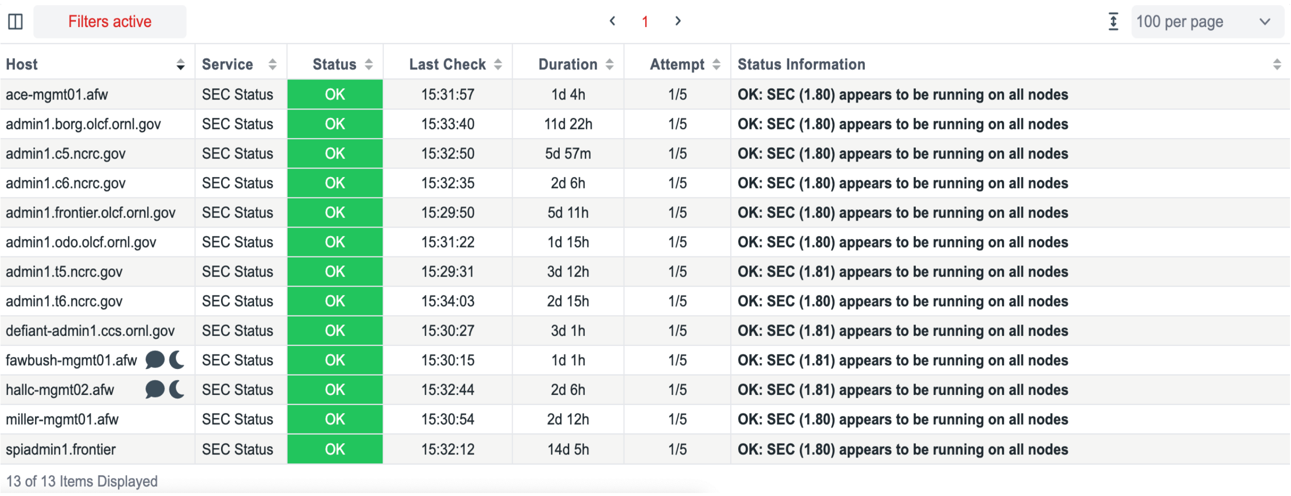
III. The Process

A. **Heartbeat Rule** The first step in the process is to establish a connection between each of the leader nodes and the cluster admin node to communicate that SEC is running. To do this, a SEC calendar rule was created on the leader nodes where they would send a message to the admin node every hour, thus creating a "heartbeat" message. When the admin node successfully receives a heartbeat, RegEx is used to make a group that filters out any excess characters and only intakes the node's name. The admin node then calls the Python script, listed as sec_heartbeat_check.py.

B. **Health Check Script** The Python script utilizes epoch value, which is a time conversion that can be used to calculate a threshold later in the process. When the admin node calls the Python script, a JavaScript Object Notation

(JSON) file is immediately created. If the file already exists, it will utilize that pre-existing JSON file and not make a new one. Once this JSON file is created, a Python dictionary is also created. The Python dictionary then takes in the leader nodes and pairs its value with the current time. That current time is converted into an epoch time to assist with calculations. The epoch time is used for a function called "threshold check". This function takes the current epoch time once again and then subtracts it with the epoch time in the dictionary. This equation is created because if a leader node does not respond, then the dictionary epoch time will remain as the previous. If the difference is greater than 60 minutes, then a "Warning" message will display stating which specific leader node's SEC process is currently down. If the difference is less than 60 minutes, then an "OK" message will display stating that the SEC process is running. If all nodes are fully operational with SEC, then there is an "OK" message stating that SEC is fully operational on all nodes. If there are any nodes that are not running SEC, then a message sends the number of nodes down as well as a warning sign. Another implementation is to obtain the version number of the SEC process that is running on all leader nodes. This is to ensure that all nodes are running the same version of SEC. This requires the Python subprocess library which is used to query the system for the installed SEC rules RPM and extracts the version number.[1, 2] To obtain the version, a RegEx library in Python was used to extract the specific version number. Then, with the overall output, the version number is printed alongside the information containing the SEC status on the nodes. Now it needs to be applied to Nagios.

C. **Visual Display** Nagios sends a Simple Network Management Protocol (SNMP) request, which is an Internet standard protocol to collect and organize information about managed devices on IP networks. From the Nagios server, a dashboard displays the output of the Python script. On the dashboard there is a list of hostnames, which are the names of the HPC systems such as Summit, Frontier, Defiant, etc. Next to these hostnames is the status of SEC on the leader nodes. It will display yellow if there is a warning and green if all nodes are running SEC properly.[4]



Host	Service	Status	Last Check	Duration	Attempt	Status Information
ace-mgmt01.afw	SEC Status	OK	15:31:57	1d 4h	1/5	**OK: SEC (1.80) appears to be running on all nodes**
admin1.borg.olcf.ornl.gov	SEC Status	OK	15:33:40	11d 22h	1/5	**OK: SEC (1.80) appears to be running on all nodes**
admin1.c5.ncrc.gov	SEC Status	OK	15:32:50	5d 57m	1/5	**OK: SEC (1.80) appears to be running on all nodes**
admin1.c6.ncrc.gov	SEC Status	OK	15:32:35	2d 6h	1/5	**OK: SEC (1.80) appears to be running on all nodes**
admin1.frontier.olcf.ornl.gov	SEC Status	OK	15:29:50	5d 11h	1/5	**OK: SEC (1.80) appears to be running on all nodes**
admin1.odo.olcf.ornl.gov	SEC Status	OK	15:31:22	1d 15h	1/5	**OK: SEC (1.80) appears to be running on all nodes**
admin1.t5.ncrc.gov	SEC Status	OK	15:29:31	3d 12h	1/5	**OK: SEC (1.81) appears to be running on all nodes**
admin1.t6.ncrc.gov	SEC Status	OK	15:34:03	2d 15h	1/5	**OK: SEC (1.80) appears to be running on all nodes**
defiant-admin1.ccs.ornl.gov	SEC Status	OK	15:30:27	3d 1h	1/5	**OK: SEC (1.81) appears to be running on all nodes**
fawbush-mgmt01.afw	SEC Status	OK	15:30:15	1d 1h	1/5	**OK: SEC (1.81) appears to be running on all nodes**
hallc-mgmt02.afw	SEC Status	OK	15:32:44	2d 6h	1/5	**OK: SEC (1.81) appears to be running on all nodes**
miller-mgmt01.afw	SEC Status	OK	15:30:54	2d 12h	1/5	**OK: SEC (1.80) appears to be running on all nodes**
spiadmin1.frontier	SEC Status	OK	15:32:12	14d 5h	1/5	**OK: SEC (1.80) appears to be running on all nodes**

13 of 13 Items Displayed

Figure 1. Nagios display of the SEC monitoring project. The "Host" lists all the HPC systems that SEC is currently running on, "Status" shows the message display, and "Status Information" will reveal if there are any nodes down or if all the nodes are operational. [4]

D. Applying into Production and Automation The Python script and the Nagios configurations are managed within Puppet. This allows management operations to be performed from one central location for multiple systems so that separate configurations do not have to be maintained on all systems running SEC. Updates that are pushed to puppet will automatically update the systems the next time they run their puppet agent on the hour. The SEC rules are managed through their own separate repository. The rules contained within this repository is packaged up and built into an installable RPM so that systems that do not run Puppet can have the custom SEC rules installed on them. [2]

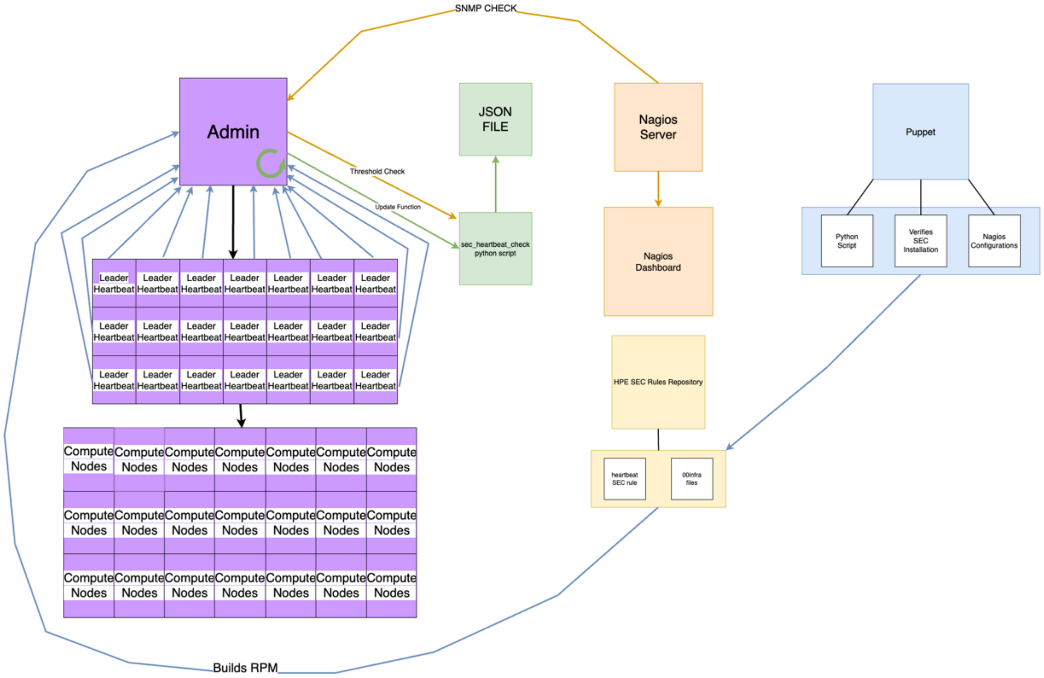


Figure 2. The purple figure displays the script working through an HPC system. The "Leader Heartbeat" sends the message to the admin node which then performs an update function using the "sec_heartbeat_check.py" script. The Nagios server performs an SNMP check, which is sent to the admin node. [4] The admin node then runs the threshold check using the script. The output is then displayed onto the dashboard. The Puppet diagram illustrates the contents of the puppet

repository used for this automation, and the “HPE SEC Rules Repository” displays the components needed to build the RPMs that are installed on the admin node.

IV. Conclusion and Final Thoughts

In this study, I have developed a sustainable tool to monitor the health status of SEC on our HPC systems. This is extremely crucial and without the triage process to ensure that the hardware is operating properly, processes running on these machines could be killed or halted due to faulty hardware or software. Having a health and version monitor is effective and suitable for a systems administrator. Overall, the benefit of having this program running is that system availability improves. This automated monitoring system for SEC helps critical HPC resources ensure their integrity and continued operation.

V. Acknowledgements This research was sponsored by the HPC Scalable Systems team in the Systems section of the National Center for Computational Sciences (NCCS) division.

VI. References

1. R. Vaarandi, June 03, 2023, “Man page of sec,” July 8, 2024
2. V. Bajrami, Nov. 27, 2020, “How to create a Linux RPM package,” July 10, 2024
3. Red Hat Documentation, "Introducing Configuration Management Using Puppet," July 10, 2024
4. A. S. Gillis, July 2023, “Nagios,”, July 11, 2024