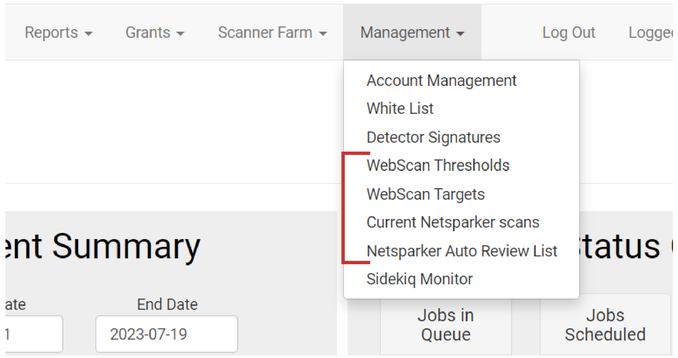
Refactoring the Beholder Dashboard

FERMILAB-POSTER-23-243-STUDENT-V

Simon Greenblatt, *Brown University,* Under the mentorship of Jason Ormes and Jeny Teheran
Security and Emergency Management Division, Cybersecurity Team, Fermi National Accelerator Laboratory, Batavia, Illinois 60510

The Beholder Dashboard

Beholder is Fermilab's main engine for cybersecurity defenses integrating detectors, vulnerability scanners, and blocking mechanisms. Written using the **Ruby on Rails** framework, the Beholder dashboard displays a summary of scanner farm events and scanning tasks. This dashboard allows the Cybersecurity Operations Team to track telemetry data while providing quick access to all major security tools. My project involved redesigning and implementing changes to parts of the dashboard by identifying new metrics to be listed.



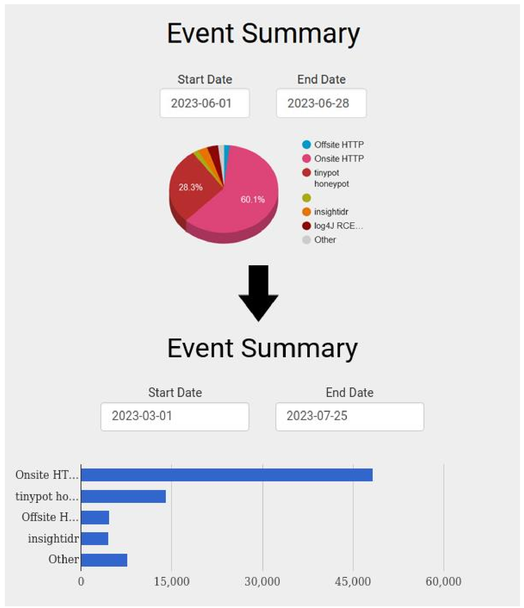
Unused menu items in the dashboard

Since several parts of the navigation bar were either outdated or deprecated, I began by removing these elements. While making these changes, I documented my work and included instructions for creating new dashboard sections. These instructions specify which parts of the **HTML, JavaScript, and Ruby** code to modify along with explanations of how the Beholder view is constructed.

Redesigning and Developing Elements

The old Event Summary pie graph, which displayed cybersecurity events such as honeypots and unauthorized services, wasn't very legible given the large number of infrequent events during scans. Using **Google Charts**, I used a bar graph to more clearly represent the most common events while still preserving a sense of proportion. Now, only the four most common events are displayed while all other events are grouped together as 'Other'.

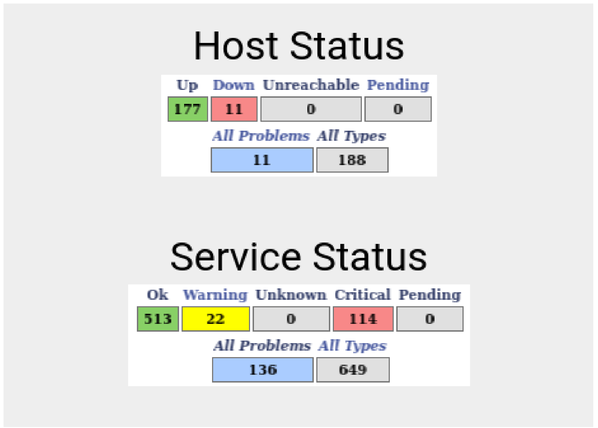
Based on the needs of the cybersecurity operations team, I added a new section containing information about hosts and services on the network that are down. I used **iframes** to import this information from the Nagios monitoring dashboard. Given that this dashboard is hosted on both a development and production server, I needed to use **YAML** configuration files with URL variables to correctly test and deploy this feature.



The old and new Event Summary chart

Future Improvements

Revisions to the Beholder dashboard could include real-time pop-up notifications for emerging cybersecurity events. Beholder also needs to be integrated with Burpsuite in order to verify web services from both onsite and offsite perspectives. This change will require a new section and interface for the dashboard to display metrics from Burpsuite.



The new Host and Status section.

Conclusion

The Beholder dashboard is a constantly evolving center for aggregating security information and tools. As old features become outdated and new systems are incorporated into everyday operations, the dashboard needs to be updated to reflect these changes. Additionally, the various and changing needs of operations team members makes the Beholder dashboard a continuous project.

This research was supported in part by the U.S. Department of Energy (DOE), Omni Technology Alliance Internship Program. The program is championed by the DOE's Office of Chief Information Officer (OCIO) and represents a partnership with the leadership of the Office of Economic Impact and Diversity, the Office of Science, the Office of Nuclear Energy, and the National Nuclear Security Agency. The program is administered by the Oak Ridge Institute for Science and Education.

This work was produced by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy. Publisher acknowledges the U.S. Government license to provide public access under the DOE Public Access Plan DOE Public Access Plan