**A Multi-Pronged Approach to Log Anonymization in GlideinMonitor**

**and**

**Computing support of
Fermilab's Summer Interns**

**Mirica Yancey**

SIST 2020 Program

Valparaiso University

Supervisor: Marco Mambelli

August 7th, 2020

**Introduction**

My SIST 2020 internship at Fermilab comprised of two distinct projects: a scientific project, “A Multi-Pronged Approach to Log Anonymization in GlideinMonitor”, that will be described in most of this final report, and an education project in support of the computing needs of other interns at Fermilab, that will be summarized in the last section of this report.

**Abstract**

Within the complex system of distributed computing environments offered by Fermilab, it is crucial for the developers of these systems, and specifically the systems responsible for job provisioning, to be able to gather statistics from log data and failures within the program. One of these systems is GlideinWMS which is a provisioning tool for distributing High Throughput Computing (1). The log files generated from running jobs in GlideinWMS are displayed in GlideinMonitor, a web application developed to view these logs. I looked into applying log anonymization techniques to GlideinMonitor to obscure the personally protected data of the job submitter in order to allow these logs the be better distributed to developers. The result was a python script capable of locating user data such as IP addresses, full names, usernames etc. and suppressing that data and displaying the filtered logs in GlideinMonitor. I then, integrated the filter into GlideinMonitor to display the filtered logs.

**Background**

GlideinWMS is a pilot-based resource provisioning tool for distributing High Throughput Computing (1). In simpler terms, it’s a job distribution tool. The purpose of GlideinWMS is to provide reliable and uniform virtual clusters to run scientific computations such as analysis, simulations, and reconstructions. It is configured to provision computers from many sources such as local clusters, the Grid, the Cloud and Supercomputers.

Once these jobs are run, information about them is saved into a log file that is then displayable on GlideinMonitor. GlideinMonitor is a web application used to view logs from GlideinWMS. It is a user interface tool used to archive, query, and decode log file content. These log files are useful to developers to compile and understand statistics and understand and correct bugs in GlideinWMS and errors in the user jobs or the resources. However, GlideinMonitor has no way to censor these logs to protect the identities of the job runners. Therefore, information like full name, employee id, email and the IP Address of the job submitter remain in the logs.

The goal is to create an automatic filter, a python script, to anonymize these personally identifying fields to protect the identities of the job runners and allow for easier distribution to developers and troubleshooters.

**Methodology**

To begin, I examined the different anonymization techniques already available and the different layers of anonymization. The first factor examined for my solution was the idea of reversible versus non-reversible anonymization. Reversible Anonymization allows for the recovery of data, often through a hashed look up table. It protects data while trying to uphold the most data use. Non-Reversible Anonymization permanently changes the data so that it is unavailable for recovery – even by the original administrator. It ensures the permanent loss of data. This method returns less information overall but still returns usable data. Ultimately, I decided to use non-reversible anonymization because the data anonymized was not data I believed would need to be recovered.

Then, I researched several different privacy models to continue to plan the implementation. I had several to choose from such as K-Anonymity, I-Diversity and T-Closeness. K-Anonymity, in its base form, is the suppression or generalization of data. This is done either by omitting data, such as, by omitting the last digit in a zip code, or by making the data a wider range. For example, instead of Patient 0 who is twenty-six years old, it is now Patient 0 who is between the ages of twenty and thirty. I-Diversity reduces the specificness of the data to a greater extend than K-Anonymity does by eliminating data, grouping similar categories, and further generalization. This may turn from Alice who had pneumonia, to a frequency of 1 in group 1 with pneumonia. T-Closeness is a more refined version of I-Diversity. T-Closeness focuses on the distance between two attributes and the “distribution of sensitive attributes within each quasi-identifier group” (2).

Ultimately, K-Anonymity was chosen for a few reasons. One, its design was excellent in its simplicity and allowed me to simply suppress the data without additional considerations. Secondly, I believed I could cover for its weakness of background knowledge (where it is easier to crack given you already have a few key characteristics) with a robust design. This makes K-Anonymity, simple, quick, and difficult to break without needing much heavy lifting.

Next, I focused on designing the script. The goal was something that would run automatically and accurately identify each necessary piece of information from a variety of file types, anonymize that information, and save it to an output directory to be displayed in GlideinMonitor. I individually annotated through a group of log files belonging to the same job to find the common denominators for references to user data. Then, I began by developing a regex script for the IP address that found the pattern for IV4 and IV6 IP Addresses and did an in-line replacement of each instance. However, further testing found this clunky and time-consuming. I then switched to a faster regex-identifying method that used more memory but didn’t read line by line, taking less time.

Afterwards, I focused on a general user information identifier script that was later split into multiple parts to accommodate for the different log file types. These methods searched for a specific identifier, saved the following information on the line, and then stripped it to what was actually the user information that was needed.

Finally, I developed helper scripts to overwrite files, call methods, write the filtered log files from the input directory to the output director, and test the program. This included some unit tests that tested each file type against a criteria for a pass or fail value.

**Conclusion**

At completion, the script is capable of several additional things. The script is able to filter logs so that they contain no personal identifiers that the unit tests can determine. Each method was tested individually to ensure function and speed without sacrificing too much to memory. Finally, the script was integrated into GlideinMonitor and included in the RPM release. Figures 4 shows an example of an anonymized log file.

**Future Work**

With the completion of this project, there are many avenues open for the future interactions of both GlideinMonitor and the log anonymization script. To begin with, I’d like to test the script against some common de-anonymization tools to see how robust the program is against attacks of different kinds. Next, I’d like to enable the anonymization of partial logs and log predictions which will predict if the job will be runnable or not by the initial log and make a decision. Finally, I’d like to create a set of aggregated data and statistics from this data to more easily identity problems in the system.

**Computing Support**

During the SIST Internship I also completed additional work to support future and current Fermilab interns. My first task was the redesign and reformatting of the TARGET Python website. The TARGET Python website was designed for Fermilab high school interns as a resource for computer science concepts and python assistance. It includes education presentations and self-guided lessons with corresponding challenges to learn the basics of Python using Jupyter Notebooks. I reformatted both the website and the presentations to create a more focused curriculum for future TARGET interns and redesigned the website to provide more helpful links with guided whitespaces to lead to each necessary section. This simplified the user’s experience and optimized their search speed. I also edited the presentations to provide clarity for some of the ideas using examples, images, and additional idea breakdowns.

My other task for assisting in the computing support of Fermilab Interns was to monitor the Fermi Coding Club #summer-interns slack channel and provide advice and guidance for any interns needing assistance in Python. I checked in daily to aid the interns and either answered questions in the slack channel itself or did one-on-one sessions through zoom.

**Acknowledgements**

I would like to thank my supervisor Marco Mambelli for his endless patience, assistance, and wisdom during this project. I would also like to thank my GlideinWMS Intern teammates LeRayah Neely-Brown and Namratha Urs for their encouragement and comradery. I’d also like to thank the GlideinWMS team and my SIST Mentoring group for their consistent welcoming attitude and belief in growing my skillset. This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

**References**

Mambelli, M., & Hein, T. *GlideinMonitor*. United States. doi:10.2172/1605567.

Shmatikov, V. (2018). *Anonymization and Re-Identification* [PowerPoint slides].

<http://www.cs.cornell.edu/~shmat/courses/cs5436/anonymization.pdf>