10-Week Internship at Lawrence Berkeley National Laboratory

During the summer of 2018 I took part in a 10-week undergraduate internship at Lawrence Berkeley National Laboratory (LBL). The department in which I was working under was the Data Science and Technology department, directed by Dr. Deborah Agarwal. During the course of the internship I was tasked with the development of user tools for a service by the name of Environmental Systems Science Data Infrastructure for a Virtual Ecosystem (ESS-DIVE). This included working on multiple different project during my 10 weeks at LBL.

ESS-DIVE is a data archiving service that is funded by the U.S. Department of Energy (DOE). The purpose of ESS-DIVE is to archive and publicly share data obtained from observational, experimental, and modeling research in Environmental Systems Science that is funded by the DOE's Office of Science.

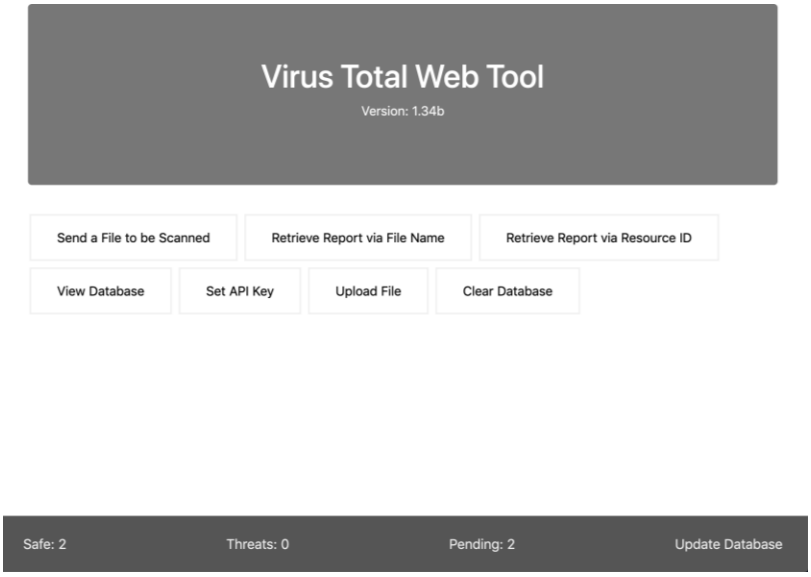
The internship began with the task of using a REST API from a virus scanning service called VirusTotal to create a command line tool. REST stands for *representational state transfer* and is a common pattern for creating an application programming interface or API as a web service. The purpose of this project was to have a working prototype that could later be implemented into a larger project. The command line tool was written in Python and made use of the "requests" library for making REST API calls. It also used a library called argparser which allowed the use of several different"flags" within a terminal, all of which enabled easy access to the different services and functions within the VirusTotal API. Along with learning REST API, this project introduced me to SQLite3 database management and command line argument parsing. The end-goal is to use the VirusTotal API to automatically scan files that are uploaded to ESS-Dive. This is important because the ESS-Dive archive is hosted on LBL's National Energy Research Scientific

```
usage: vt.py [-h] [-db] [-clean] [-version] [-post POST] [-update] [-get GET]
             [-item ITEM]

optional arguments:
  -h, --help  show this help message and exit
  -db         Will print the current state of the database that houses
              informationon each file passed through the api.
  -clean      Will delete all records in the current database.
  -version    Program version
  -post POST  Used to send file to virustotal api to be scanned. | python
              vt.py -post <filepath> |
  -update     Will update current file status or create new record in existing
              database
  -get GET    Value should contain either the keyword <sha> when sending a
              file's full path or the keyword <resource> when sending a file's
              resource id. | python vt.py -get sha -item <filepath> | python
              vt.py -get resource -item <file resource id> |
  -item ITEM  Value should contain either resource id for the desired file or
              the file's full path.
```

Computing Center (NERSC) and allowing users to upload harmful files is a potential security risk.

Following the first project, I was tasked to create a web application that implemented the core functionality of the previous project. The purpose of this web app was to create a more user friendly GUI (graphical user interface) using the VirusTotal API. This was achieved using Flask, which is a microframework for Python, along with other web development tools such as AJAX and JavaScript. This web app used a lot of the existing functionality from the command line tool while also adding extended functionally. Each file that is sent to the VirusTotal API has four possible statuses that it can be in: *"file sent"*, *"safe"*, *"threat"*, and "*queued*". The status (along with additional information) of every file sent through this web app is stored in a SQLite3 database. *"Safe"*, *"threat"*, and *"queued"* are fairly self explanatory but *"file sent"* is slightly different. For the status "*file sent"* the user has successfully sent a file to the API but a status has not yet been retrieved from the API. The user can also choose to update all files that hold a status of *"file sent"* or *"queued"*. As an alternative the user can also update each file individually on the database viewing page.



The final project that I worked on for LBL involved using the Globus Transfer API to create a second web application that allowed a user to create a shared folder on the ESS-Dive shared endpoint. The issue that this web application intends to solve is that the ESS-DIVE web portal has a file upload size limit. With this functionality a user will be able to fill out a form with their Globus information and create a shared folder on the ESS-Dive shared endpoint, to which they can upload large datasets to the endpoint and notify the ESS-Dive team once they're

done. Once the ESS-Dive team has been been notified, these datasets can be scanned for viruses, and then uploaded to the ESS-DIVE archive using internal tools that don't have the same upload restrictions. This enables a much higher performance and high volume data upload interface for ESS-DIVE users.

In conclusion, the experience gained from my internship at LBL will come to benefit me in the upcoming years of not only my college career, but also my professional development. I am extremely grateful for the opportunity to work with such an amazing team on a project that has such a significant impact.