EE / CprE / SE 492 Bi-Weekly Report

Project title: Knowledge Graphs for Cybersecurity Reasoning

01/24/23 – 02/18/23

Group number: sdmay23-01

Client & Advisor: Benjamin Blakely

Team Members:

Brandon Richards - *Frontend Development Lead*

Micah Gwin - *Python/ML Development*

Alice Cheatum - *Programmer*

Nicklas Cahill - *Tester/Programmer*

Michael Watkins - *Python/ML Development*

Carter Kitelinger - *Client Interaction*

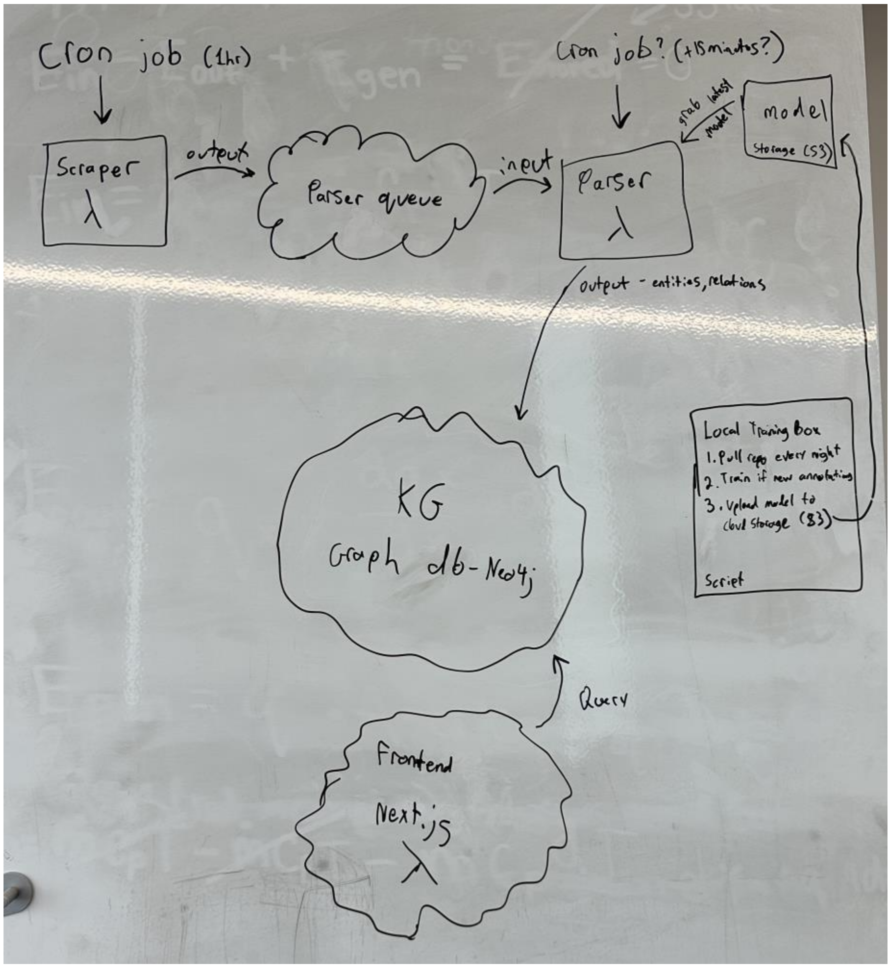
Summary

There were numerous objectives for these past two weeks, most of which were achieved. The annotated documents are now in the correct format for BRAT, BRAT can run on x86 or ARM architecture, and the entities and relationships set out in the design document have been configured in BRAT. An initial implementation of the parser was completed with basic NER capability and strong interfaces for future implementations. This parser uses articles in MongoDB as an input and the graph database Neo4j as an output for entities and relationships. Infrastructure was planned and AWS was selected as the cloud service provider for this project.

Past Two Week Accomplishments

- Convert annotations to BRAT – Brandon
  - Annotations performed in previous software saved in a format that is incompatible with current annotation tool BRAT. Wrote Python script to input annotations done in previous format and write new annotation files in new format. Annotations can now be opened and performed in BRAT.
- Finish BRAT container – Brandon
  - Added previously completed annotations into shared data folder.
  - Added some entities and relationships defined in design document into BRAT configuration file.

    - Created a README for running the Docker container.
    - Added support for ARM architecture to Dockerfile.
    - Fixed bug where running BRAT with Cygwin wouldn't find data directory.
- Create initial NER Parser implementation – Brandon
    - Added MongoDB Docker container to same network as parser. Parser now uses input implementation that gets articles from MongoDB.
    - Added Neo4j Docker container that serves as the persistent knowledge graph.
    - Added NER component using spaCy and a generic English model. Currently captures information such as organizations, locations, and numbers.
- Update annotations for BRAT annotations - Nicklas
    - Update BRAT configuration file with all entities and relationships supported in our design document.
- Validate Scraper and MongoDB – Micah
    - Updated and troubleshooted Dockerfile to ensure scraper module is running as expected and is storing output in Mongo as intended.
    - Given certain sources ensure the scraper returns with the appropriate articles and they are parsed correctly
- Infrastructure Testing and Experimentation – Micah
    - Experimented with CI/CD workflow that monitors GitHub repo and deploys needed infrastructure using GitHub actions and terraform
    - Began testing CI/CD integration with modules, deployed some example infrastructure to ensure the workflow will work on more complicated modules
- Get lists for parser matching - Alice
    - Found lists of companies and attack groups that can be matched in text without NER
- Research RE training techniques – Alice
    - Found multiple tutorials for how to do relationship extraction using spaCy
- Find and annotate new articles for training data – Alice
    - Collected several articles from the sources we previously decided on
    - Added article texts to brat data in a new branch on GitHub
    - Annotated new articles
- Research RE – Carter
    - Found a way to train both NER and RE at the same time, potentially reducing training time
    - Found a way to use spaCy to train RE, as well as instructions on how to create a custom RE trainer
- Research testing metrics/testing strategies – Carter
    - Found other ML creators used data synthesis libraries such as Synthetic Data Vault (SDV) to create synthetic data sets
    - Found some companies have ML model testing platforms for automation (may or may not apply to us, not quite sure)
    - Code Quality tests (linting)
    - Stress Testing ML model with weird input
    - Manual test review for output of annotated articles and knowledge graph creation (when we get there)

- Plan infrastructure – Everyone
    - Planned system infrastructure, including all components. Planned the platforms the components will run on (e.g., Lambda Function, Database, Local Linux OS, etc.).
    - Decided to use cron job to run scraper and parser components.
    - Decided to train locally using Python scripts and annotated data, then upload trained model to S3 for use by parser.
    - 

Pending Issues

- Current complications are getting the spaCy pipeline set up for NER and RE training. Steady progress is being made but there is a lot of work to be done with it in a short time span.

Individual Contributions

Name	Hours past two weeks	Hours cumulative
Brandon Richards	18	52
Micah Gwin	13	31

Alice Cheatum	9	34.5
Nicklas Cahill	8	25
Michael Watkins	5	25
Carter Kitelinger	8	30.5

Summary of weekly advisor meeting

The meeting started off with items completed in the previous sprint. This included the parser's support for MongoDB input and Neo4j output. We also discussed potential solutions for orchestrating the different components/containers. Our best idea by the end of the meeting was Amazon SQS (although this later got changed to a cron job). Lastly, we discussed a fallback if we're unable to perform relationship extraction in time. The fallback is recording how strongly different entities are correlated by measuring their frequency of mentions and distance to other entities.

Plans for Coming Two Weeks

- NER + RE Model – Brandon
  - Create pipeline for training NER and RE models simultaneously.
  - Create Python script or Shell script to run training easily.
  - Create task to check if new training data exists every night. If so, train and upload to S3.
- Infrastructure Deployment – Micah
  - Continue improving GitHub actions pipeline and activate it in the real repo
  - Write terraform code for all the different infrastructure and test if it deploys using the pipeline
- Further Testing Research/Preparation and Annotate Articles – Carter/Nicklas
  - Continue to find information about testing strategies and metrics
  - Talk to others about creating tests
  - Annotate Articles for training ML model
- Preparation for ML Training – Alice
  - Continue to annotate more articles for training data
  - Other tasks as necessary