EE / CprE / SE 492 Bi-Weekly Report

Project title: Knowledge Graphs for Cybersecurity Reasoning

02/18/23 – 03/04/23

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

Finished the main implementation and configuration of NER and RE training separately, with combining them into one model happening within the next week. Annotations now being converted into spaCy format, then trained on using transformers and a GPU, with useful machine learning metrics now being printed after training. Wrote Python script to fetch CVE metadata of interest for inserting into the knowledge graph. Annotated more articles, created example Cypher queries, and lastly successfully deployed code onto AWS infrastructure.

Past Two Week Accomplishments

- Convert annotation format for training – Brandon
    - Fork open-source annotation format converter to convert from BRAT annotation format to .jsonl format for spaCy model training.
    - Add functionality to software to output relations (previously only output entities).
    - Output entities and relations from articles in BRAT annotations folder.
- Train NER using transformers and GPU – Brandon
    - Create spaCy configuration for NER model training that utilizes GPU.
    - Setup Windows machine with NVIDIA RTX 3060 TI graphics card for training.
    - Randomly select 20% of articles for testing the trained model against.
- Output useful model evaluation metrics – Brandon

    - Write “evaluate” script that runs after model training.
    - Output common machine learning metrics including accuracy, precision, recall, and F1 score.
- Write CVE fetch script – Alice
    - Research NVD CVE web API
    - Implement python script to retrieve information about a given CVE
    - Output relevant information fields
- Annotate articles for training data – Alice
    - Find new articles from chosen sources
    - Annotate entities and relationships using BRAT
- Determine which fields of the CVE JSON object we are interested in – Everyone
- Built and deployed pipeline - Micah
    - Get green pipeline build that is able to deploy zipped code with dependencies and the terraform without issue
    - Deployed the remaining AWS resources (DynamoDB and S3 bucket) and modified the scraper lambda output pipeline to send to DynamoDB
- Utilized NVD API to get CVE information including CVE description and CVSS metadata using Python script– Nick
- Annotated articles – Nick
- Annotated Articles - Carter
- Created sample Neo4j Cypher queries - Carter
    - Researched Cypher queries and created small examples to query information contained in the knowledge graph

Pending Issues

- How do we improve metrics of accuracy, recall, precision, and f1 score in our NER and RE model?

Individual Contributions

Name	Hours past two weeks	Hours cumulative
Brandon Richards	16	68
Micah Gwin	14	45
Alice Cheatum	8.33	42.83
Nicklas Cahill	7.33	32.33
Michael Watkins	0	25
Carter Kitelinger	8.2	38.7

Summary of weekly advisor meeting

Our meeting started by sharing out infrastructure plan with our advisor to get feedback. One recommendation he served was putting AWS CloudFormation in front of our S3 bucket that serves the frontend of the project. He also recommended saving the version of the scraper used when scraping articles, that way if we upgrade our scraper we can re-scrape articles that may not be as accurate. We discussed switching from Neo4j to AWS Neptune, but ultimately decided not to due to cost concerns.

Plans for Coming Two Weeks

- Improve scores – Brandon
    - Improve accuracy, precision, recall, and F1 score of NER and RE models.
- Fetch CVE metadata in background – Nicklas, Alice
- Continue annotating articles – Alice
- Finish CVE info Fetcher - Nicklas
- Create more Cypher queries and annotate articles - Carter
- Parser Lambda Updates - Micah
    - Modify parser lambda to run green without any runtime or dependency issues
    - Have the parser lambda get the model from s3 before a run instead of it being uploaded manually.
    - Modify the input to the parser lambda to be from DynamoDB.