2 Project Plan

2.1 Project Management/Tracking Procedures

Our team has chosen agile as a project management style because our project goal of developing a knowledge graph will require many iterations. New knowledge and metrics that we acquire during sprints will help us better understand and plan for the next set of tasks we need to perform in the next sprint.

Our team is going to use Git as a VCS and GitHub to host our repository. We will also be using JIRA as a ticketing system to better track tasks, subtasks, sprints, backlog items, and responsibility for each task.

2.2 Task Decomposition

1. Determine list of sources to obtain news articles and blog posts.
    a. Inspect sources for legitimacy and reputation
    b. Record URLs of main page and/or specific pages of interest (e.g., Malware blog)
2. Develop scraper to obtain news articles, blog posts, etc., using Scrapy
    a. Select a programming language
    b. Select libraries to perform downloads and parsing of HTML
    c. Create file specification for storing list of sources
    d. Write code for scraper
    e. Write testing code for scraper
    f. Output artifacts for later use by NER model
    g. Containerize with Docker
3. Develop one or more methods to clean up articles (may vary depending on type of article)
    a. Research existing methods for cleaning up irrelevant information
    b. (Potentially) Manually annotate relevant vs. irrelevant data in documents
    c. Verify on test cases that relevant information isn't being destroyed
4. Extract relevant entities (vulns, companies, software, exploits, etc.) and the relationships between them
    a. Research existing annotation techniques and cybersecurity-specific NER models
    b. Determine if we need to train custom NER model
    c. (Potentially) Train NER model:
        i. Manually annotate set of documents from selected sources
        ii. Perform supervised machine learning to train NER model
    d. Generate entities and relationships from cleaned-up source information
5. Use extracted entities and relationships to generate knowledge graph
    a. Collect output from Information Extraction
    b. Insert into graph database
6. Run pipeline created in steps 2-5 periodically and continuously on new articles
    a. Create job to run at interval
    b. Determine if new articles of interest have been posted
    c. Run new articles through pipeline

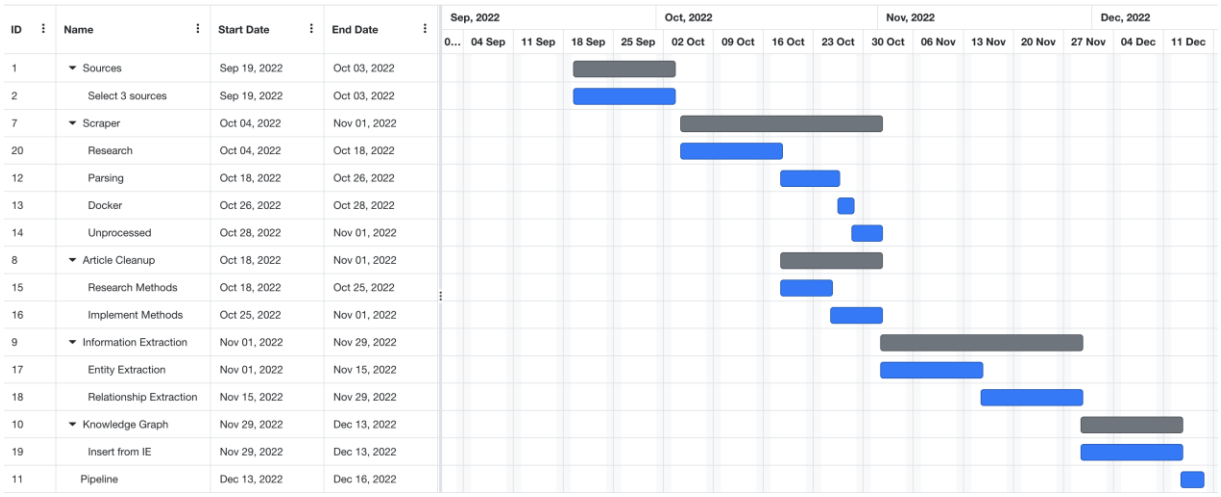
7. Develop a web interface to run queries on the graph
    a. Design
        i. Develop prototype
        ii. Receive feedback from client
    b. Develop
        i. Select framework to make website
        ii. Create API to query knowledge graph
        iii. Implement designs from prototype
        iv. Implement filters to query the graph
        v. (Stretch goal) Use natural language to query the graph

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

1. Sources
    1.1. 3 or more sources have been selected to scrape for information
2. Scraper
    2.1. Scraper is able to download and parse input sources.
    2.2. Runs inside Docker container.
    2.3. Identifies more recent unprocessed articles with 100% accuracy.
3. Article cleanup
    3.1. Articles achieve removing unnecessary information with 25% accuracy.
    3.2. Articles achieve removing unnecessary information with 50% accuracy.
4. Extract entities and relationships
    4.1. Software can identify subjects (companies, operating systems, vulnerability, etc.) with 75% accuracy.
    4.2. Software can identify relationships (vulnerability works on this OS and application run by this company) with 50% accuracy.
5. Generate knowledge graph
    5.1. Contains more than 15 entities including companies, operating systems, applications, malware, etc.).
    5.2. Nodes have properly labeled edges with 75% accuracy.
6. Pipeline periodic and continuous running
    6.1. The job runs on the specific interval 100% of the time.
7. Web Interface
    7.1. Prototype delivered to client with 80% satisfaction (satisfaction to be quantified with rating survey).
    7.2. API created to query 100% of knowledge graph entities and relationships.
    7.3. Filtering by Company, Application, OS, Vulnerability, or Malware can be performed.
    7.4. (Stretch) Natural Language query is able to serve intended results 50% of the time.

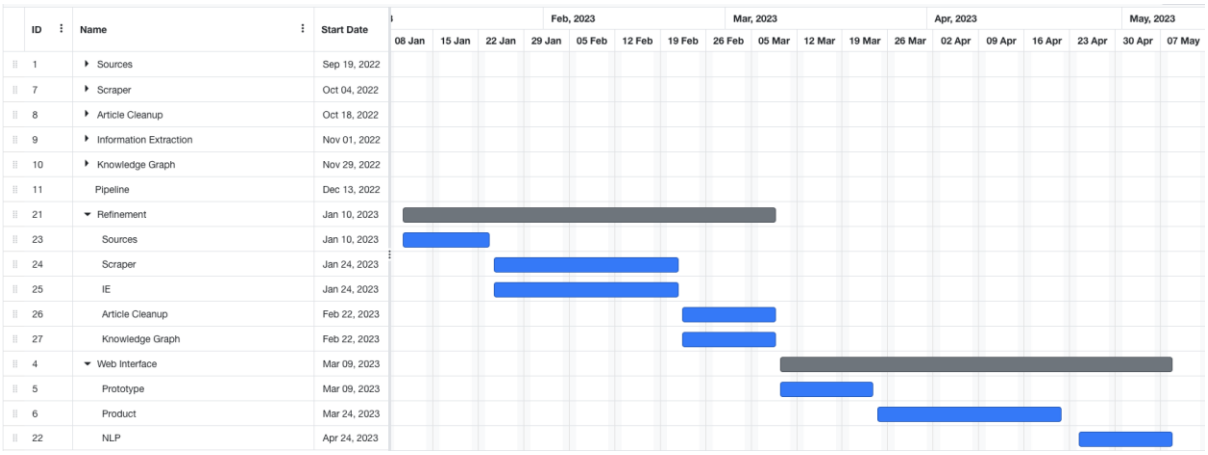
2.4 Project Timeline/Schedule

Semester 1 Schedule



ID	Name	Start Date	End Date
1	Sources	Sep 19, 2022	Oct 03, 2022
2	Select 3 sources	Sep 19, 2022	Oct 03, 2022
7	Scraper	Oct 04, 2022	Nov 01, 2022
20	Research	Oct 04, 2022	Oct 18, 2022
12	Parsing	Oct 18, 2022	Oct 26, 2022
13	Docker	Oct 26, 2022	Oct 28, 2022
14	Unprocessed	Oct 28, 2022	Nov 01, 2022
8	Article Cleanup	Oct 18, 2022	Nov 01, 2022
15	Research Methods	Oct 18, 2022	Oct 25, 2022
16	Implement Methods	Oct 25, 2022	Nov 01, 2022
9	Information Extraction	Nov 01, 2022	Nov 29, 2022
17	Entity Extraction	Nov 01, 2022	Nov 15, 2022
18	Relationship Extraction	Nov 15, 2022	Nov 29, 2022
10	Knowledge Graph	Nov 29, 2022	Dec 13, 2022
19	Insert from IE	Nov 29, 2022	Dec 13, 2022
11	Pipeline	Dec 13, 2022	Dec 16, 2022

Semester 2 Schedule



ID	Name	Start Date
1	Sources	Sep 19, 2022
7	Scraper	Oct 04, 2022
8	Article Cleanup	Oct 18, 2022
9	Information Extraction	Nov 01, 2022
10	Knowledge Graph	Nov 29, 2022
11	Pipeline	Dec 13, 2022
21	Refinement	Jan 10, 2023
23	Sources	Jan 10, 2023
24	Scraper	Jan 24, 2023
25	IE	Jan 24, 2023
26	Article Cleanup	Feb 22, 2023
27	Knowledge Graph	Feb 22, 2023
4	Web Interface	Mar 09, 2023
5	Prototype	Mar 09, 2023
6	Product	Mar 24, 2023
22	NLP	Apr 24, 2023

2.5 Risks And Risk Management/Mitigation

Description	Likelihood	Consequences	Risk	Mitigation
Training uses too many resources	Unlikely	Major	High	Optimize code, set resource usage limits, allocate more GPUs for cluster

Model is trained incorrectly	Moderate	Moderate	High	Start early, involve Michael in most decisions due to experience, Research
Source is too difficult to perform extraction	Moderate	Moderate	High	Pre-scout source's format, plan extraction logic ahead
Resources too high to display query	Moderate	Moderate	High	Limit number of displayed responses
Sources go down or block our traffic because of too much scraping too fast	Likely	Major	Extreme	Strict rate limits

2.6 Personnel Effort Requirements

Tasks This Semester	Sources (1)	Scraper Design (2)	Clean Up Articles (3)	Extract Entities and Relations (4)	Generate Knowledge Graph (5)	Total for Semester 1
Estimated Hours Per Person	2 hours	15 hours – Program Team	10 Hours – CybSec Team	20 - 30 hours – Both Teams (Varies Greatly)	8 hours – Both Teams	68-78 Hours

- Program Team Consisting of: Brandon Richards, Michael Watkins, and Micah Gwin.
- CybSec Team Consisting of: Alice Cheatum, Nicklas Cahill, and Carter Kitelinger.

Based on the table above, we broke the hours down into what we thought we could finish in the first semester. Using the tasks from the Task Decomposition section, we decided it would be best if we split into two teams: the Program Team, handling most of the Python programming, and the CybSec Team, handling most of the non-programming work that involves Cyber Security knowledge. As provided in the table above, each team was given an estimated hours per person. This also gives each team a different part to work on during sprints, since things can be worked on in parallel.