CprE 492 Bi-Weekly Report 02

2/7/2021 - 2/21/2021

Group #: 50

Project Title: Cy-Sec Game

Advisor: Manimaran Govindarasu

Front End Members: Jon Greazel, Hayden Sellars, Joseph Strobel

Back End Members: Harrison Majerus, Nicholas Battani, Stefan Peng

---

Bi-Weekly Summary: During the last two weeks we have made steady progress toward a completed project. We are on schedule and very close to completing the attack tree analysis engine. The backend API connection is nearly complete, our analysis engine is done and needs testing, and our front end is taking shape and able to send pertinent information to the backend.

Past 2 Weeks Accomplishments:

- Jon:
    - Node palette & retaining shapes: This involved gaining a deeper understanding of go.js and how it declares its palette constants. Eventually I was able to create the custom shapes we needed and assign them to the correct nodes
    - API handler: Created a post method to give the b/e our node data and edge data. They will run this through their algorithm and return the correct risk probabilities
    - Styling: this was mostly CSS to give our app a more consistent formatting and color scheme
- Hayden:
    - Fixed the diagram to be in Graph Link Format
    - Restructured inspector component
    - Basic HTML and CSS formatting to make frontend more interesting and attractive
    - Importing Angular Material components into front-end
    - Added Node attributes for Probability and Text
    - Removed Root Node

- Joe:
    - Recreate project for tree format: Initially we decided to change layouts in the GoJS library. After some research and testing this change, the frontend team decided to revert back to the graph layout, so I made the change back and reorganized the project from experiments to our finalized file structure under the frontend folder.
    - Saving graph data: The graph can now be saved to a .json file for the user to upload later. This json object contains a list of the nodes and a list of the links in their graph.
    - Loading graph data: The user can upload their saved graph to the application to be used again and reanalyzed.
- Harry:
    - Iterated on attack tree algorithm pseudocode.
    - Created working prototype algorithm to run hard-coded example JSON with processing helper functions and scenario class
- Nick:
    - Set up the backend API and database.
    - Connected backend with frontend using RESTful API.
- Stefan:
    - Attack tree algorithm: Created initial pseudocode attack tree algorithm. Worked on a prototype algorithm and started to integrate it with the API to allow processing data from the frontend.

Pending Issues:

Individual Contributions:

Team Member	Contribution	Weekly Hours	Total Hours
Jon Greazel	Created node palette, fixed issue where shapes changed on graph entry, API handler for f/e, f/e styling	6 / 7	13
Hayden Sellars		6/6	12
Joe Strobel	Experimented with tree layout and finalized layout decision; saving graph data; loading graph data	6 / 7	13
Harry Majerus	Attack tree algorithm and data processing functions	6/7	13

Nicholas Battani	Set up backend API/Database, connected frontend and backend using RESTful API.	6/6	12
Stefan Peng	Attack tree algorithm: research, design, implementation	6/6	12

Plans for the Upcoming 2 Weeks:

- Jon:
  - Create a new node for the safe path and update our inspector. The new inspector will assign an additional value, impact, and these values will need to be removed from and/or nodes since they shouldn't have any value.
- Hayden:
  - Create new Inspector components for the different nodes because they require different values.
  - Use Materials.io to implement some visualization on front end for analyzing
- Joe:
  - Begin data validation ticket - this entails checking that the sent to the backend includes proper values for probability, impact and that there is only one safe path. I'll also continue research into updating the graph diagram.
- Harry:
  - Additions to attack tree algorithm including factoring in impact of attacks into algorithm output, a function for normalization of probabilities assigned to attacks, and modifications for new consideration of a 'safe path' in user tree (probability of no attack). Understand attack-defense tree processing from an algorithm perspective.
- Nick:
  - Hook up API to algorithm libraries, assist in research of new algorithms.
- Stefan:
  - Start research and prototyping on the attack-defense tree algorithm. This will require adding the ability to process defense nodes as well as computing the Nash equilibria for the possible payoff functions.

Summary of Weekly Advisor Meetings: During our meetings with Manimaran we spent time demonstrating the work we'd done the previous week. He is pleased with our progress so far and is excited to see the back end and front end connected during our next meeting. We’ve also come to the conclusion that our backend won’t be powered by 3 engines, but just 2 now. We plan to combine attack-defense trees with game theory. Burhan also gave a presentation on potential implementations of this during our last meeting.