

Website for Sharing Information

sddec-21-16

Advisor: Lotfi Ben Othmane

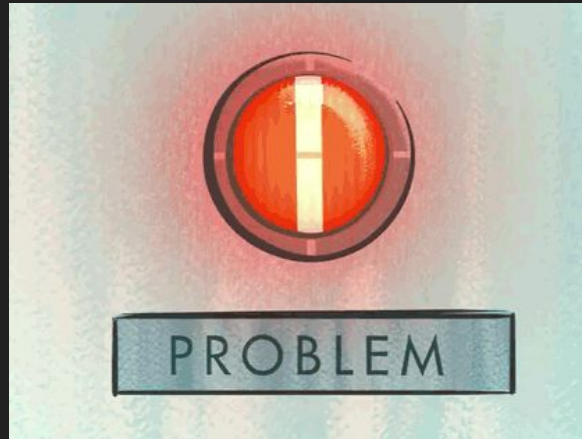
Client: Lotfi Ben Othmane

sddec21-16

Website for Sharing Information

The Problem

- Currently no centralized database of Threat Modeling Patterns
 - Different from CVE
- Threat Modeling Patterns
 - Allows designers to be aware of possible areas of high risk



Concept

- A website which allows cybersecurity experts to post various threat modelling patterns in the form of blog posts
- A text mining algorithm will mine the blog posts for relevant information, allowing easy search and viewing of data
- Visualization in the form of graphs/charts

Functional Requirements

- Ability for security experts to post blog posts
- Ability to search posts
- Ability to see a breakdown of different threat models based on blog posts

Non-Functional Requirements

- Up to 10,000 concurrent active users

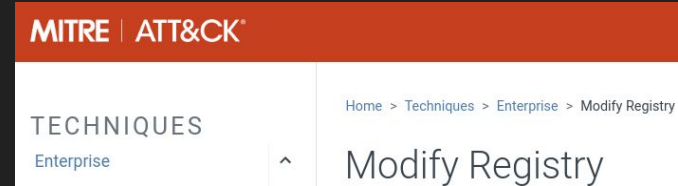
- 1s or less response time

Technical Constraints/Considerations

- Text mining algorithm is provided
 - Spike will occur by client/advisor to find and implement algorithm
- Maintain project sustainability
 - Project will be handed off to future senior design groups

Market Considerations

- Mitre & Nist
 - CVE vs Threat Model



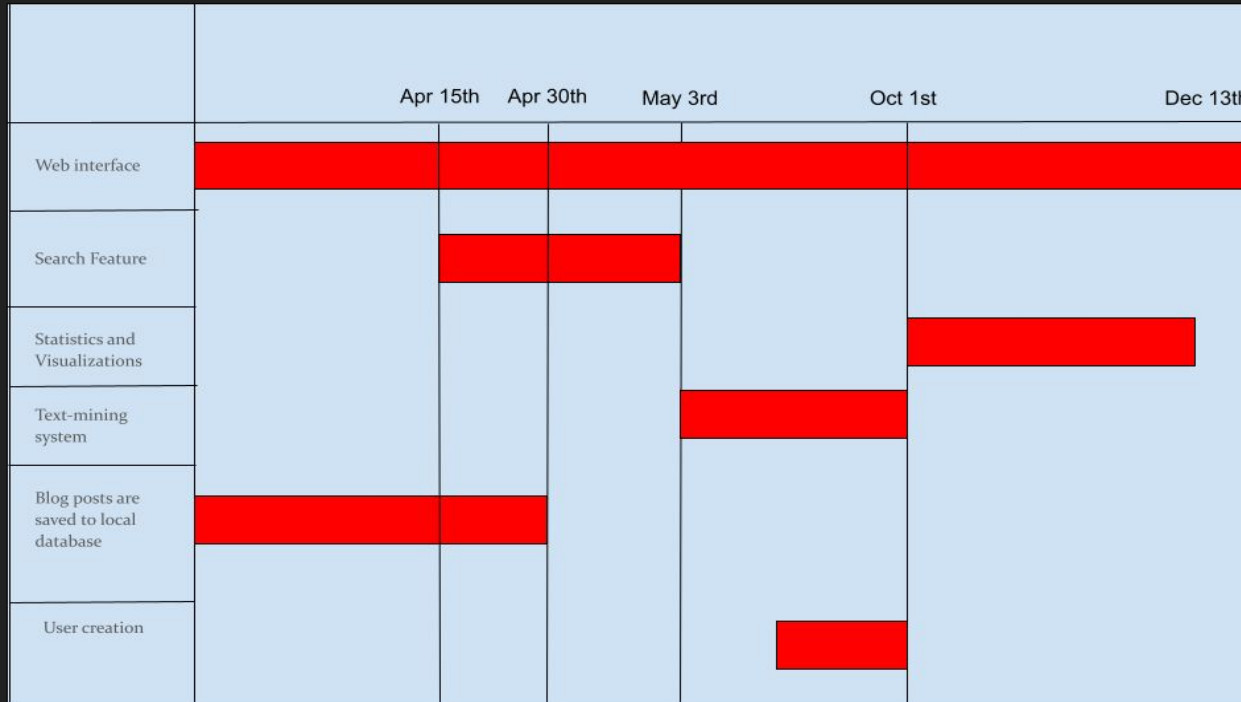
Potential Risks and Mitigation

Task	Risk	Mitigation
Text-mining system will scan over all blog posts and retrieve the user, context, problem, solution, and alternative solution.	.6	Develop a system to regularly sanitize inputs/flag potential data poisoning attempts. Not possible to eliminate due to the uniqueness of the project.
Website will include a search feature which can search for blog posts and/or threat modeling patterns	.5	Developing this feature ourselves would be very costly and potentially dangerous for SQL injections and other forms of mishandled edge cases. This feature should not be developed by the team, we will use a predeveloped library or built in functionality of React.

Resources

- Website hosting server
- Amazon AWS or Google GCP
- Text-mining algorithm

Project Schedule



Functional Decomposition

- Threat model handling
 - Threat model submission
 - Submission form
 - Threat model parsing & cleansing
 - Threat model storage
 - Threat model viewing
 - Threat model search
 - Threat model storage
 - Threat model multi-display
 - Threat model single-display
 - Threat model analysis
 - Text mining

Design - Interface Definition

- Project will utilize a RESTful API
- Example;

GET /posts/list

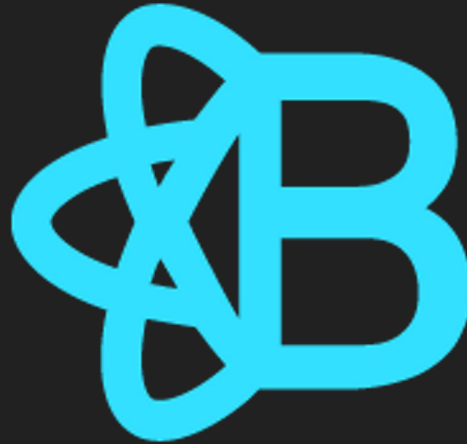
Description: Get all posts submitted

Request Body: Null

Response Body: {
 "username" : string,
 "context" : string,
 "problem" : string,
 "solution" : string,
 "mitigation" : string
}

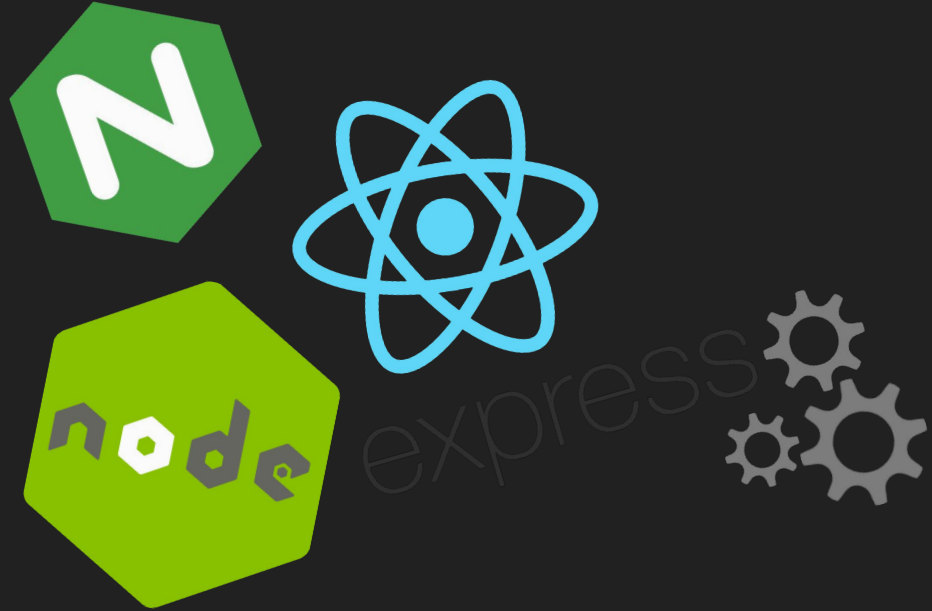
Design - User Interface

- UI design is in the early stages
- Will use React-Bootstrap for styling and formatting



Technology Utilized

- Frontend
 - React
 - React-Bootstrap
- Backend
 - Node.js
 - Express.js
 - Nginx
 - SQL
- Other Hardware/Software
 - University provided virtual machine
 - Linux command line server
 - SSH



Test Plan

- Jest framework for JavaScript unit testing
- Selenium for browser testing
- Postman for basic API testing



Current Project Status

- Working blog post submission to the database
- Displays blog posts from the database
- Backend completed search blog post search functionality
- Automated deployment
- Currently working on searching for the frontend
- Currently researching user management

Prototype - Threat Model Submission

Title [Home](#) [About](#) [Post](#)

Hello from server!

Username

Username

This will be entered automatically once authentication is set up.

Context

Context

Provide any context related to the threat.

Problem

Problem

What problem does the threat cause?

Solution

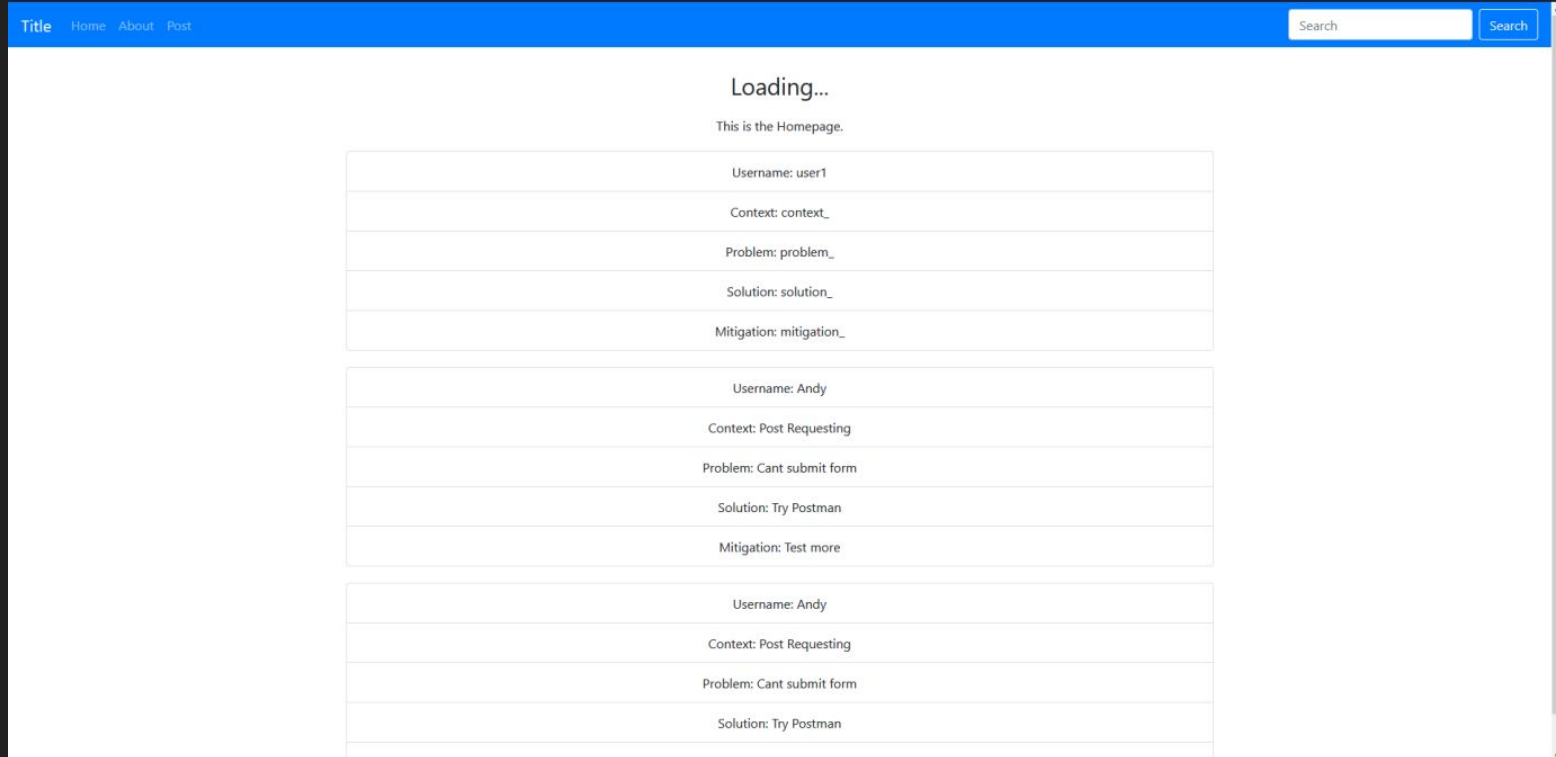
Solution

What can be done to solve the problem?

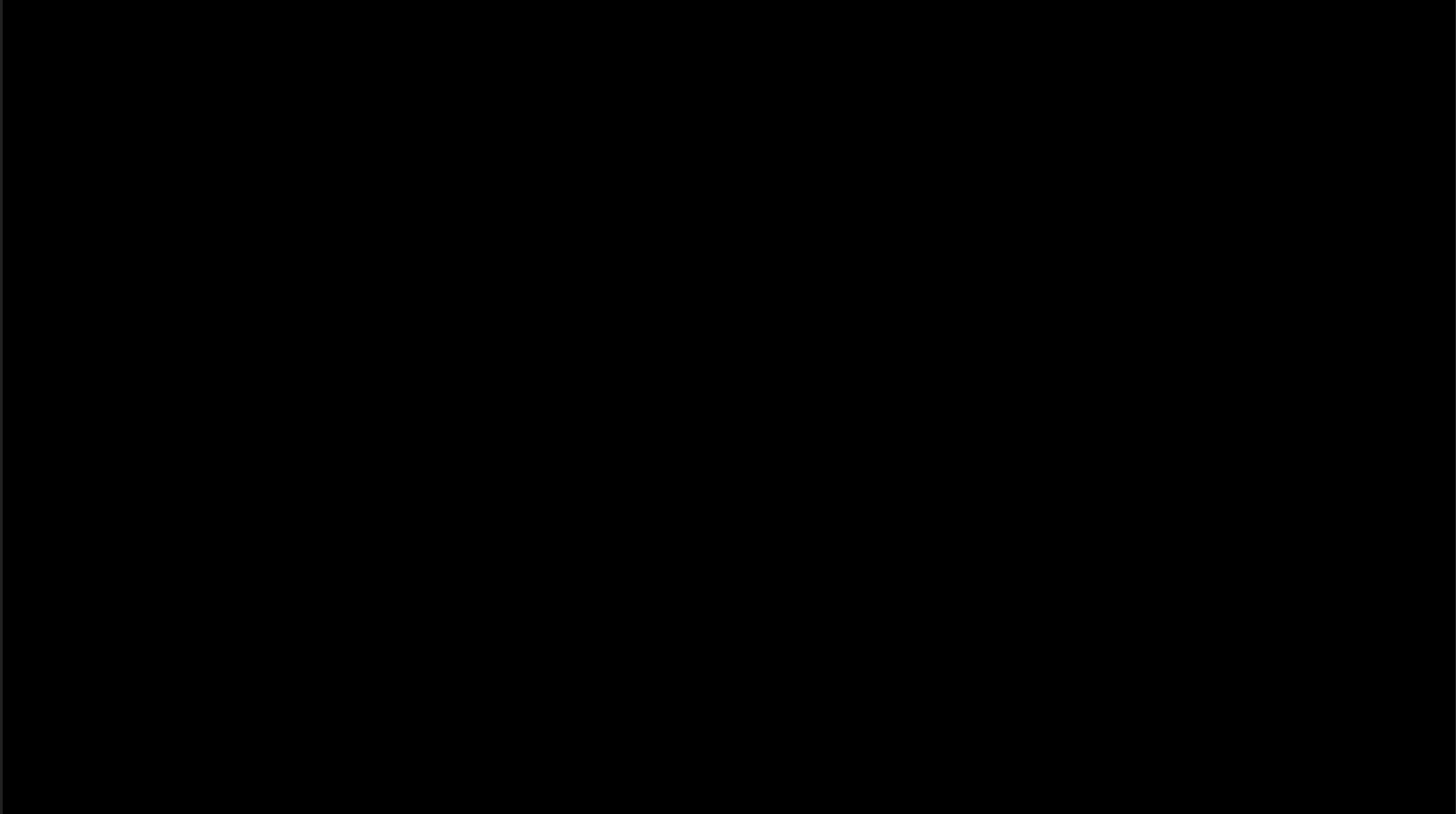
Alternate solution

Alternate solution

Prototype - Threat Model Post Viewing



Prototype - Video



sddec21-16

Website for Sharing Information

Member Responsibility/Contributions

- Jacob Abkes
 - Project planning, organization, documentation, initializing development environments and projects, client engagement
- Dylan Black
 - Front-end development, integration testing, Unix knowledgebase
- Andy Dugan
 - Back-end API development such as the threat modeling submissions and search functionality
- Jack Phillips
 - Front-end UI development
- Zhi Wang
 - Server management, web hosting service, project runtime environment and database setup

Plan for next semester

- Complete implementation of search feature.
- User authentication and management.
- Integrate text-mining algorithm provided by advisor.
- Improvement of user interface and general website design.