Website for Sharing Knowledge

Design Document

Team 16
Client: Lotfi Ben Othmane
Advisor: Lotfi Ben Othmane
Team members: Andy Dugan, Jack Phillips, Jacob Abkes, Zhi Wang, Dylan Black
sddec21-16@iastate.edu
http://sddec21-16.ece.iastate.edu/

Executive Summary

Development Standards & Practices Used

Our engineering standards are as follows:

- Split production from development
  - Prevents developer from accidentally deleting production data
  - Better test environment
  - Good control of the software version
- Developing in the same environment
  - Helps mitigate cross environment confusion
  - Allows team members to collaborate more effectively
- Document before execution
  - Our project use cases will be identified and documented on a discovery basis
  - This prevents development from moving faster than user acceptance
- Iterative approach requires modifiability and modularity
  - Components should be easily modifiable and loosely coupled

Summary of Requirements

Functional requirements:

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

Environmental requirements:

- An authentication system which prevents unauthorized personnel from submitting blog posts, to protect the dataset from being 'poisoned' by bad data
- Design a browser-friendly interface for users that can be accessed anywhere

Economic requirements:

- Uses university resources to circumvent costs associated with hosting and processing

- Switch away from wordpress to avoid third party plugins that could invoke costs

Applicable Courses from Iowa State University Curriculum

- SE 329 - Software Project Management
- SE 339 - Software Architecture & Design
- COMS 352 - Intro to Operating Systems
- SE 319 - Constructing User Interfaces
- ENGL 314 - Technical Communication
- COMS 363 - Intro to Database Management Systems
- COMS 311 - Introduction to Algorithms

New Skills/Knowledge acquired that was not taught in courses

- All knowledge relating to using and implementing Wordpress
- Text mining algorithms
- Most cybersecurity and threat modeling knowledge
- Knowledge of the React framework
- Knowledge of the Node js framework
- Workflow of a React + node js application
- Use of the React-Bootstrap framework

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

1 Introduction

1.1 Acknowledgement

We would like to acknowledge Lotfi Ben Othmane for being not only the middle-man between our clients of cyber security experts, but also for being our adviser and giving us weekly advice and direction.

1.2 Problem and Project Statement

Problem

There is currently no centralized database that efficiently documents and organizes threat modelling patterns. A system such as this is necessary in the fight to ensure the safety and security of software systems around the world. The ability to easily gain the knowledge required to effectively mitigate current threats is something that many organizations and specialists wish they had, as such a database has the ability to adapt and change as new threats and vulnerabilities are discovered.

Solution

Our team has devised a web application that includes a database of documentation regarding various threat modelling patterns. This web application will allow users to submit knowledge regarding threat modelling patterns in the form of blogs, where these blogs will be data mined to be more efficiently organized so that a third party could more easily access the information they would need. Our application will also allow for modification of these blog posts, as it is understood that just as technologies are always evolving, so are the threats facing them.

1.3 Operational Environment

Due to our project being software-based, any environmental hazards are limited to the server. The server is hosted at Iowa State during development, and a copy of all of the files needed to run the website are saved on GitLab. The website itself from a user perspective will be able to run on any modern web browser.

1.4 Requirements

Functional requirements:

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

Environmental requirements:

- An authentication system which prevents unauthorized personnel from submitting blog posts, to protect the dataset from being 'poisoned' by bad data
- Design a browser-friendly interface for users that can be accessed anywhere

Economic requirements:

- Uses university resources to circumvent costs associated with hosting and processing

- Switch away from wordpress to avoid third party libraries which might invoke costs.

1.5 Intended Users and Uses

Our intended users are cybersecurity professionals. Only approved cybersecurity experts can make blog posts, while anyone will be able to view the posts and the information provided by the text-mining algorithm.

1.6 Assumptions and Limitations

Assumptions:

- All users with login credentials will be verified cyber security experts
- Concurrent user count doesn't exceed 10,000.
- Text Mining Algorithm will be provided

Limitations:

- Text mining is limited by the training data set
- Application cannot function without users inputting thread modelling data

1.7 Expected End Product and Deliverables

The end product that will be delivered to the client is all of the source code for the website and any related code and documentation needed to run the project.

Expected deliverables:

- Web application for submitting threat modeling patterns in the form of blogs
  - The web application will be the primary product. The system for submitting blogs is based on React.
- Use text-mining system to extract threat modeling information from each of the blogs
  - The text-mining algorithm will save information such as the user, context, problem, solution, and alternative solution derived from the blog posts to a database.
- Visualize the threat modeling knowledge to experts
  - Using the information saved to our database, we will then have a different page that allows the user to easily view statistics and graphics related to threat modeling patterns.

2 Project Plan

2.1 Task Decomposition

Deliverable 1: Web application for submitting threat modeling patterns in the form of blogs

- User interface created using React, where users can create accounts and submit blogs
- Backend request handling through Node.JS, where incoming posts are submitted to a MySQL database

Deliverable 2: Text-mining algorithm derives threat-modeling information from blog posts and saves it to a database

- Text-mining system will scan over all blog posts and retrieve the user, context, problem, solution, and alternative solution.
- It will then save these fields to a database

Deliverable 3: Polished web interface that allows for easy searching of blog posts and threat modeling patterns, in addition to visualizations such as graphs and charts generated from the database.

- Web interface will be polished for general release
- Website will include a search feature which can search for blog posts and/or threat modeling patterns
- A separate webpage will show statistics and visualizations such as graphs and charts to provide an overview of all of the threat patterns

2.2 Risks And Risk Management/Mitigation TABLE 1

Task	Risk Probability	Mitigation
• User creation and blog posting handled by React	.3	N/A
• Incoming blog posts are saved to local database via Node.JS	.1	N/A
• 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.
• Web interface will be polished for general release	.2	N/A
• 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.
• A separate webpage will show statistics and visualizations such as graphs and charts to provide an overview of all of the threat patterns	.3	N/A

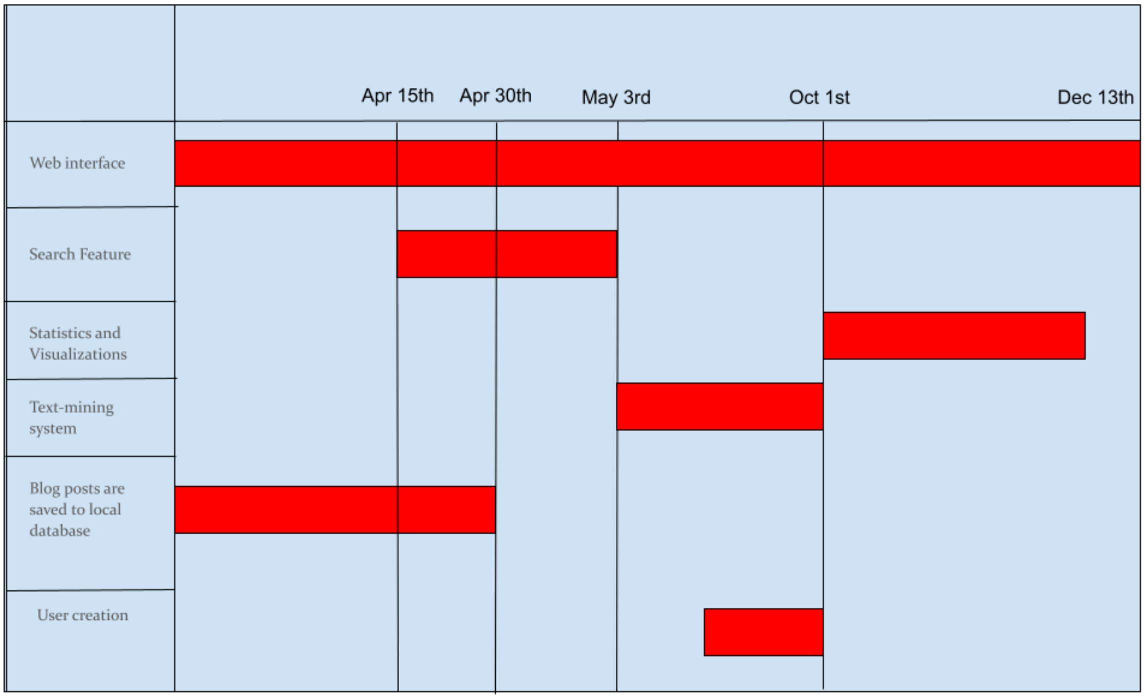
2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Our milestones will be tied with our deliverables and subtasks.

- Working with React for blog posting and user creation/permissions.
- Text-mining algorithm will extract all necessary fields from the blog posts with 80% accuracy
- Working search functionality that allows searching of all text blog posts
- Visualizations will be continuously updated with the addition of every new blog post

2.4 Project Timeline/Schedule

FIGURE 1



Our team schedule is based off of the expectations from deliverables 1-3. The goal by the end of the semester is to have finished deliverable 1 and started deliverable 2. The requirements are that we have deliverable 1 done with plenty of time to spare, as to give the client time to hand the project off to experts. This requirement is what will keep our group on schedule. While the web interface will always be changing and new features being added, this makes the task ongoing.

2.5 Project Tracking Procedures

- GitLab - Version control
- Trello - Issue tracking
- Discord - Team Instant Messaging
- WebEx - Team meetings
- Google Drive - Shared file storage
- Google Docs - Collaborative documentation

2.6 Personnel Effort Requirements TABLE 2

Task	Description	Projected hours required
Web interface	HTML/CSS design	200
Search feature	Searching of blog posts	35
Statistics and visualization	Uses information extracted from text-mining algorithm to	100
Text-mining system	Extracts information from blog posts to database	85
React blog posts	Blog posts are done through React	70
User creation	User creation is handled through React	40

2.7 Other Resource Requirements

- Server for web hosting

2.8 Financial Requirements

- The only financial constraint for this project is the requirement of server purchase. In the current phase, all servers are sponsored by the Department of Electrical and Computer Engineering of Iowa State University.
- The financial need for server deployment outside of the network of Iowa State University is expected to be provided by the team advisor.

3 Design

3.1 Previous Work And Literature

As Professor Lotfi Ben Othmane said himself, what we're doing in this project has no direct precedent. In regards to research, our group is not aware of any research that may or may not have led to the inception of this project.

3.2 Design Thinking

The definition process is focused on specifying a target to identify user needs and core issues, then reframing them to reflect new knowledge. Ideation is dedicated to diving into problems with teams to generate new ideas.

3.3 Proposed Design

As of 3/9/2021, we have an Ubuntu virtual machine hosted at Iowa State's Electrical and Computer Engineering department that is running a Node.js web server alongside an NGX load balancer combined with React frontend. We plan on creating a user-friendly interface that handles all frontend requirements by using the React framework. This will satisfy the functional requirements of having a blog-posting system, and also provides a basis for user creation and permissions.

Beyond the React web page, we will need to implement a text-mining feature that runs with the addition of every new blog post. In addition, we will need another library to visualize the data retrieved by the text-mining algorithm. All of these services will be hosted on the same server.

3.4 Technology Considerations

WordPress turned out to be too restrictive for our goals to tolerate, therefore, we decided to implement this web page using React for front end interfaces, and Node.JS on the backend for request handling.

React will require some time to learn to achieve an ideal understanding of web page design. However, React pairs very well with Node.js with regards to the functionality of this web application. Alongside this, React is a super lightweight javascript framework. This allows our team to only pull dependencies in when they are needed.

3.5 Design Analysis

As of 3/9/2021, it is too early to tell if our proposed design will be all that we need. It will be very difficult to change from WordPress, so we expect WordPress to stay with us the entire project. We will need to iterate the design with how we plan to implement the text-mining and visualization programs.

As of 4/1/2021, we've decided that WordPress does not implement as much functionality as we would like, therefore we are switching to the React framework for front end, and Node.JS as a back end framework. This will require a full restart of our project, but thankfully, React and Node.JS are very fitting for the requirements of this project and will require little work to get as far as we were with WordPress.

As of 4/22/2021 we have kept the decision to use React with a Node.js backend. In addition to this, we have added and implemented the React-Bootstrap framework in order to speed up the styling of the front-end.

3.6 Development Process

We will implement an iterative development process on a weekly basis. Every Tuesday we have a group meeting where we discuss what needs to be done for the week. Every Thursday we have a meeting with our client/adviser to discuss further requirements that are expected from our clients, as well as ask any questions related to the tasks for that week.

3.7 Design Plan

FIGURE 2

In the current phase of the project, which is to make a web application for submitting threat modeling patterns in the form of blogs. We have designed the architecture of our server.

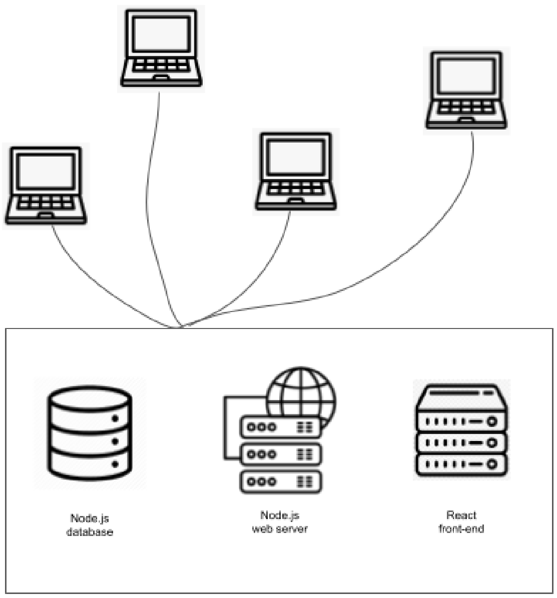


FIGURE 3

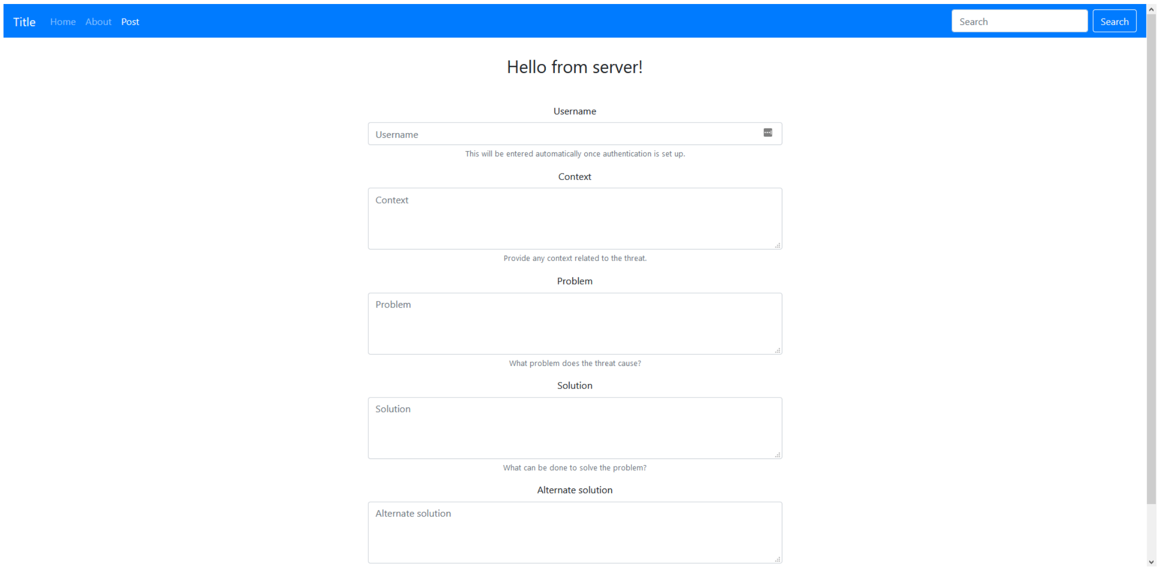
Below is our proposed database schema for a single threat model post.

```
CREATE TABLE ThreatModel (
                         ID int NOT NULL AUTO_INCREMENT,
                         Username TEXT NOT NULL,
                         Context TEXT NULL,
                         Problem TEXT NULL,
                         Solution TEXT NULL,
                         Alternative TEXT NULL,
                         PRIMARY KEY (ID) );
```

ID	Username	Context	Problem	Solution	Alternative

FIGURE 4

As of 4/22/2021 the blog post submission page looks like this. Subject to change.



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

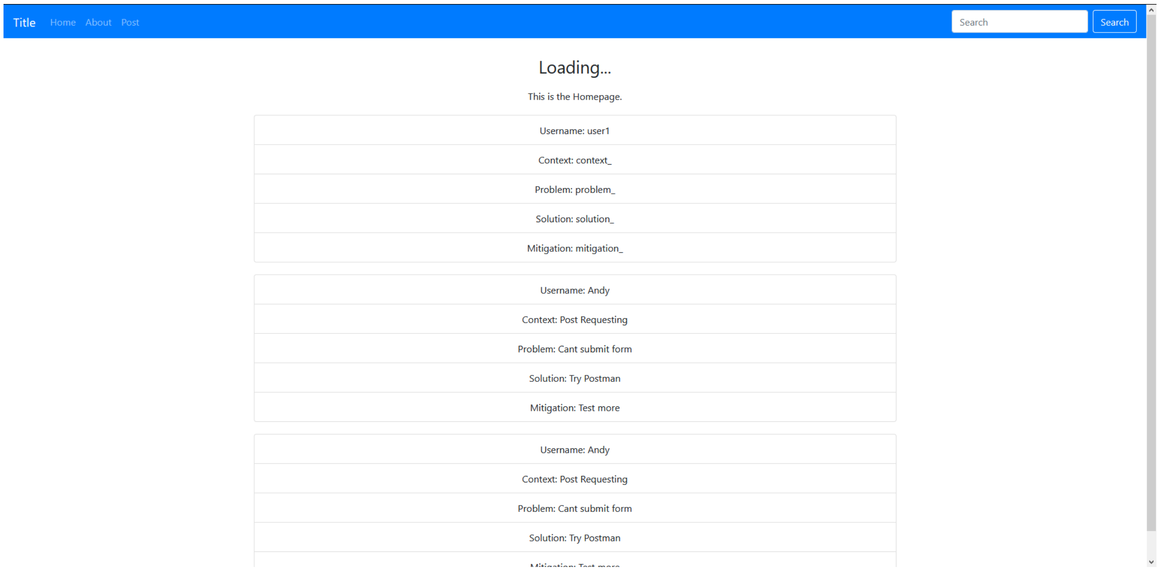
What can be done to solve the problem?

Alternate solution

Alternate solution

FIGURE 5

As of 4/22/2021 the home page looks like this. It is currently filled with example data. Subject to change.



Loading...

This is the Homepage.

Username: user1

Context: context_

Problem: problem_

Solution: solution_

Mitigation: mitigation_

Username: Andy

Context: Post Requesting

Problem: Cant submit form

Solution: Try Postman

Mitigation: Test more

Username: Andy

Context: Post Requesting

Problem: Cant submit form

Solution: Try Postman

Mitigation: Test more

4 Testing

4.1 Unit Testing

We plan to use the Jest JavaScript testing framework in order to run unit tests on our React code. Facebook, the creator of React, has stated they use Jest in order to test their React code, so it seems like the best option. As of 4/22/2021 we haven't implemented the Jest framework.

4.2 Interface Testing

We plan to use Selenium, a suite of tools for automating web browser testing, in order to test our interface across different browsers. As of 4/22/2021 we haven't implemented Selenium testing.

4.3 Acceptance Testing

We intend to communicate back-and-forth with the security experts collaborating with us on the project in order to achieve their desired functionality and look-and-feel. Our team will utilize the project advisor (Lotfi) as the middleman between us and the user acceptance testing.

4.4 Results

Each group member has installed a copy of the running environment to their local machines, so that most changes can be tested locally. For things that cannot be tested locally (such as database reads and writes), we coordinate running on our VM for brief periods of testing, ensuring that the stable branch goes live again once testing has concluded. In the future, we intend to use an additional server for beta testing, so that our live server can remain stable always.

The master branch of our application is the final 'production' copy of our software. This means any code that lives on this branch should have gone through thorough testing and review. This code could cause user issues, thus we must also have a dev branch for testing. This branch is less controlled and is used for testing your features on a production environment without actually polluting the production copy of our software.

5 Implementation

Due to the nature of our project and after talking to Lotfi, our project will not follow the typical senior design workflow of designing first and then creating the project. Instead, we will design and create parts in an iterative development style. Due to this, we will have some deliverables done before the start of the second semester. Anything that we don't get done during the first semester will be done in the second semester.

6 Closing Material

6.1 Conclusion

So far we have created the first iteration of our web application which includes very basic forms of functionality regarding the implementation of our web server and database, along with a first implementation of WordPress. However, we decided to forgo WordPress due to it not fitting with our requirements as initially thought. As of writing the second version of this design document, we are basically restarting our project in terms of code. We are switching to the React framework in order to make a responsive front-end, and using Node.js as our back-end. This will involve redoing the front-end and back-end.

6.2 References

“Support – Official WordPress.com Customer Support.” Support - WordPress.com. Accessed March 9, 2021. https://wordpress.com/support/.

“React – A JavaScript Library for Building User Interfaces.” – A JavaScript library for building user interfaces. Accessed April 4, 2021. https://reactjs.org/.

“Jest · Delightful Javascript Testing.” Accessed April 4, 2021. https://jestjs.io/.

“React-Bootstrap Documentation” Accessed April 22, 2021. https://react-bootstrap.github.io/.

6.3 Appendices