Attack Atlas

A Website for Sharing Threat Information

By group sddec21-16: Jacob Abkes, Andy Dugan, Jack Phillips, Dylan Black, and Zhi Wang
With help from faculty adviser and client Lotfi Othmane

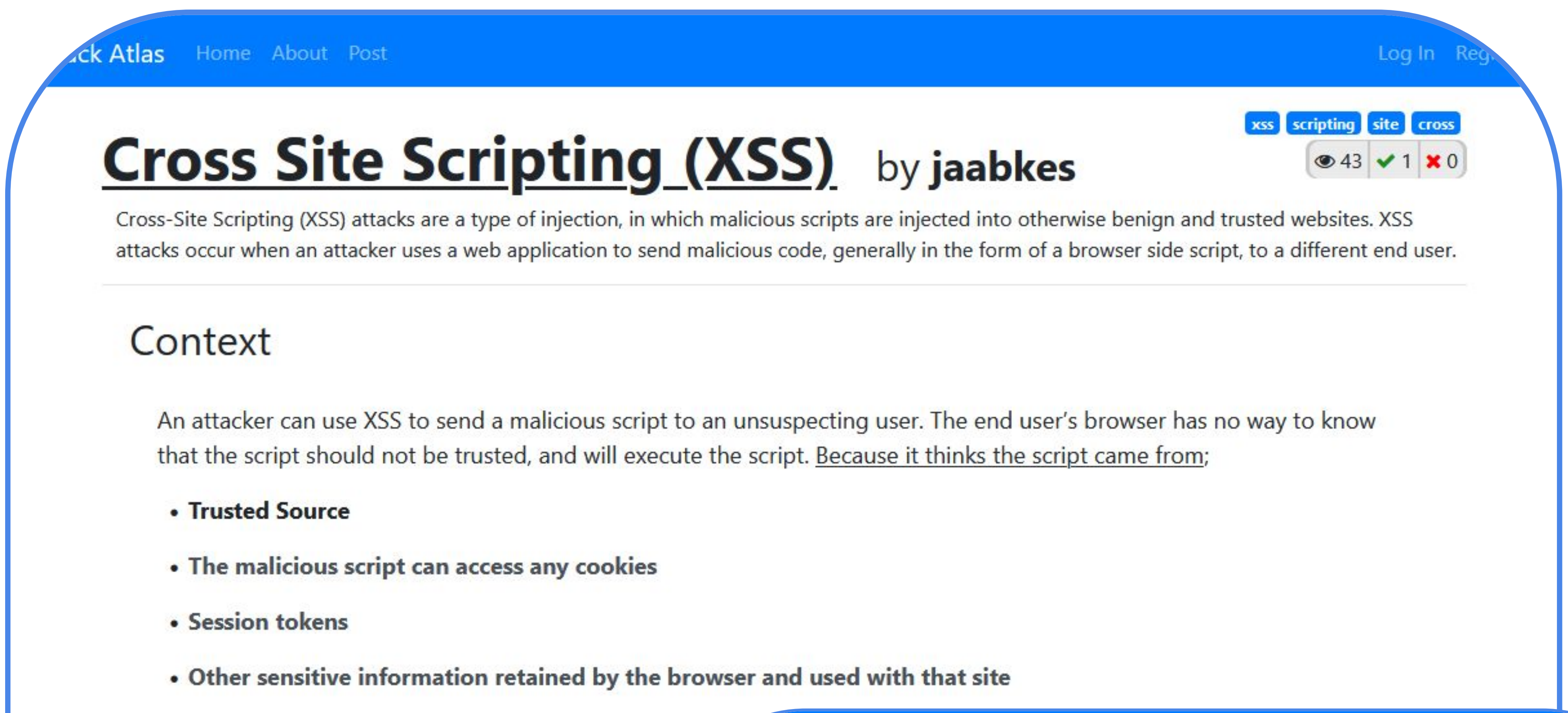
Introduction

Our goal was to develop a website that can be used as a centralized database for threat modelling patterns. It can be used to in the risk management process to identify, mitigate or resolve potential security concerns for a software product.

Users and Uses

Intended for those looking to seek information on cybersecurity in the form of threat models provided by professional cyber security consultants.

Cybersecurity experts are able to submit threat posts, while standard users can comment and view these submissions in the form of blog posts.



Cross Site Scripting (XSS) by **jaabkes**

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.

Context

An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from;

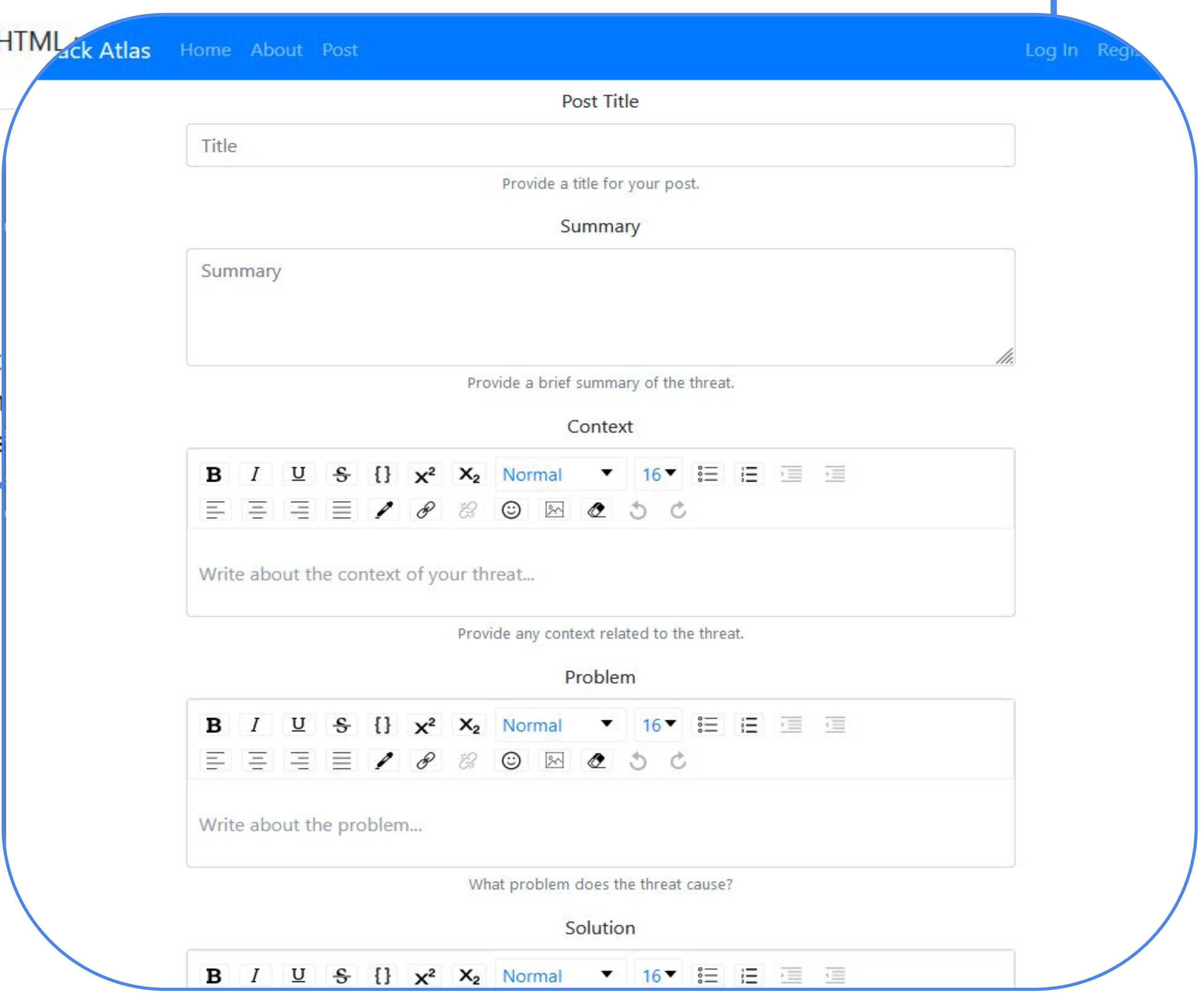
- **Trusted Source**
- **The malicious script can access any cookies**
- **Session tokens**
- **Other sensitive information retained by the browser and used with that site**

These scripts can even rewrite the content of the HTML

Problem

Stored XSS Attacks

Stored attacks are those where the injected script
message forum, visitor log, comment field, etc. Th
requests the stored information. Stored XSS is als



Post Title

Title

Provide a title for your post.

Summary

Summary

Provide a brief summary of the threat.

Context

Write about the context of your threat...

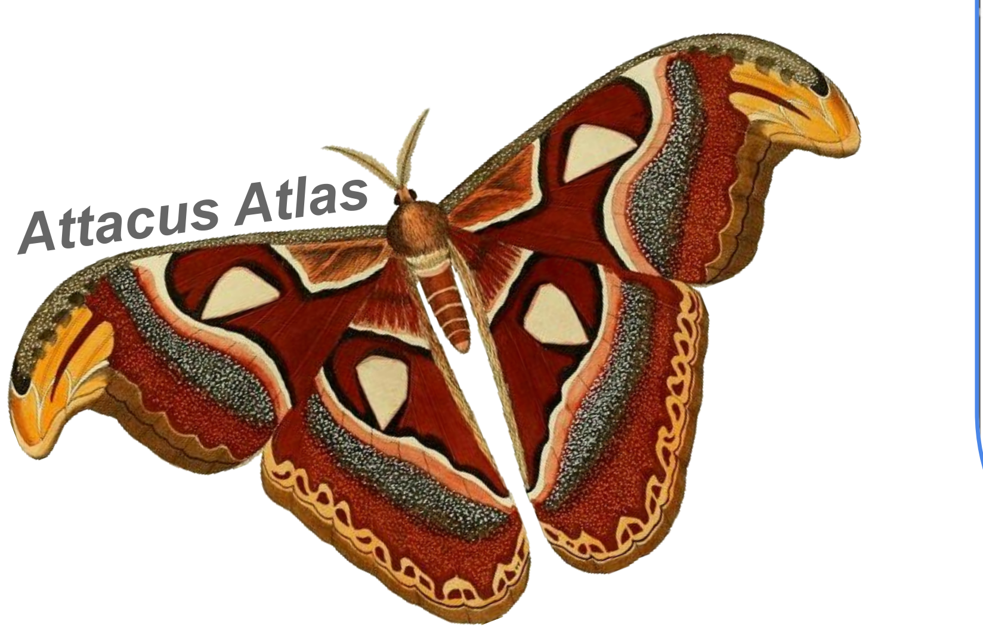
Provide any context related to the threat.

Problem

Write about the problem...

What problem does the threat cause?

Solution



Design Requirements

Functional:

- Security experts can create blog posts
- Anyone can search and view blog posts
- Ability to see a breakdown of different threat models

Non-functional:

- Simple to use
- Blog-like system

Engineering constraints:

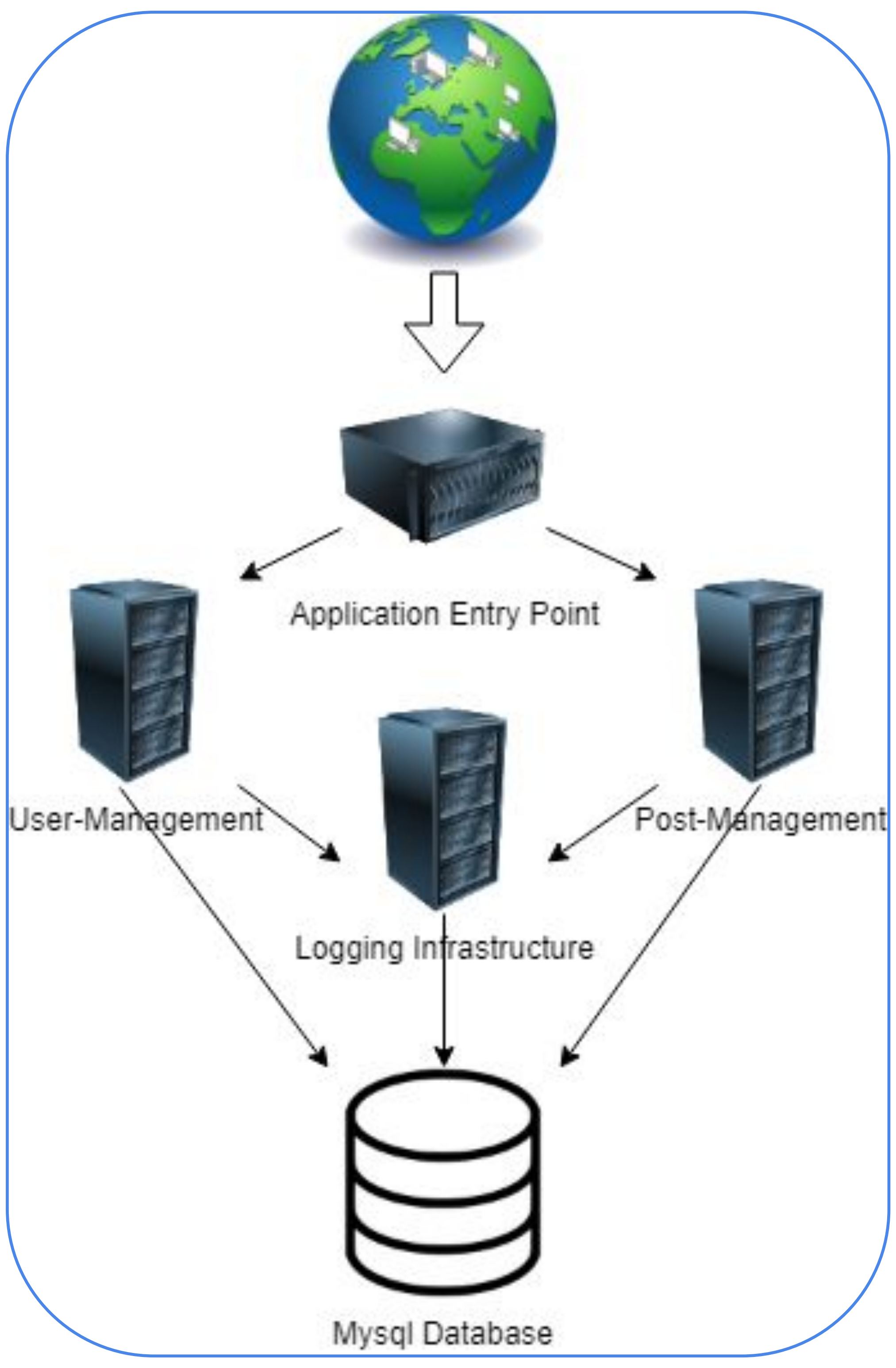
- Email confirmation doesn't support JSON
- MySQL doesn't support JSON blobs

Relevant standards:

- Unified development environment
- Git feature branch workflow
- Separation of development and production environment
- Shared dependency management
- Agile-based development cycle

Testing

Postman, Mocked data, Logging System testing, and Regression testing.



Design Approach

Unlike other Senior Design projects, we began in the first semester developing our prototype website with the guidance of our adviser and client Lotfi. This resulted in less concept sketches, as our website is meant as a proof of concept itself.

Changes in approach:

Over the course of our project, we determined that we wanted to shift to a blog style website that is similar to social media. Users can comment, upvote helpful threat models, and post incident examples related to these threats. The benefit of this is the ability to interact directly with cybersecurity experts. Our architecture developed into a microservice-based architecture to easily allow a multitude of hosting options.

Modules and features:

Post Management
- Blog posting and viewing
- Post discovery and search
- Tag system
- Incident example posting and linking to threat posts

User Management
- Commenting system
- Voting system
- User sign-in and registration

Logging Infrastructure
- Customizable entry points for internal services
- Unified collection point for logs

Security considerations:

Our team considered leveraging a Google SSO solution for user-management and Oauth2 user token management to minimize a data breach of our users protected data. Our prototype implementation utilizes prepared statements for database interactions, and also whitelists characters.

Technical Details

React
React-bootstrap
Draft.js
Express.JS
Node.JS
LogLevel
MySQL
Linux VM