Introduction

For this project, our goal was to build a tool that combines a real-time messaging platform with digital forensics logging. We wanted something straight forward and easy to use, but still capable of capturing valuable forensic data behind the scenes. We wanted it to have encrypted messaging standards and another website that does something similar is called WhatsApp, hence the name WhereApp.

The main digital forensics problem this project solves is evidence collection. Many digital forensic investigations consist of analyzing chat messages and network activity as they are critical pieces of information. Our tool captures usernames, timestamps, ip addresses, and mac addresses. Our tool also stores the message content in encrypted form. This allows someone to use the tool for investigative purposes to preserve communication data without exposing any sensitive information.

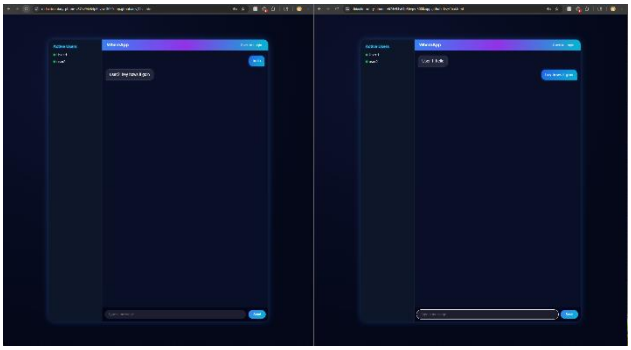
Technical Implementation

We built WhereApp using a mix of HTML, CSS, Shell, JavaScript, Node.js, Express, and Socket.io. The login page and chat interface were both coded in HTML/CSS with a JavaScript client that handles real-time communication. The backend is where all of the forensics comes into the mix. We used Node.Js with Express to serve the pages, and Socket.io to take care of live communication between users. Every time a user connects, sends a message, or disconnects the script logs, Username, IP address, Mac address, the encrypted version of the message that was sent, and the IV. Message encryption was implemented in AES-256, so the server never stores anything in plaintext. The messages were stored inside of the queue/inbox directory, in the form of a text file. Even though the logs are usable for analysis, the actual message itself stays protected unless a key is provided. According to our research, this is exactly how a real organization would go about handling user privacy while also complying with law enforcement when it comes to forensic analysis. The communication logging script was created to log MAC addresses which allows investigators to tie a message to actual hardware instead of just a username. We had to download Socket.io and Express. Socket.io was used to allow users to

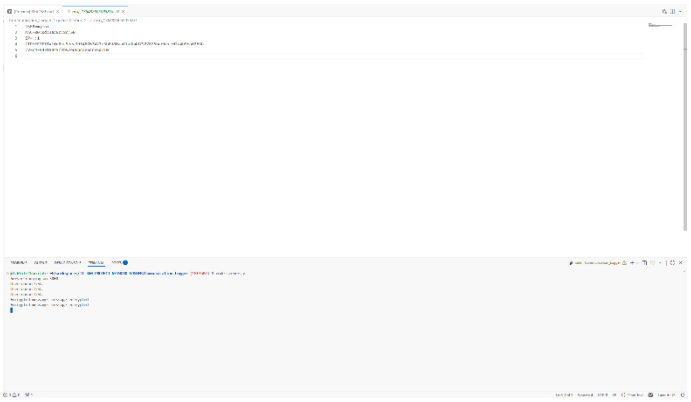
communicate in real time, and express was used to create a web server that multiple people can access.

Results

When we were testing the application, we first needed to ensure that multiple people could be on the site and were able to exchange messages with each other in real time.



The image on the right shows two users communicating with each other. The next thing we needed to ensure that it worked was the backend logging part. As you can see, the backend shows that the communication aspect with documented, and most importantly, encrypted. After the files are encrypted, they are stored inside of the queue/inbox directory in the form of a .txt file as stated above. With this, it also shows the Username, MAC address, IP address, the message itself, and the IV. This shows successful communication between two parties being stored for future viewing.



What We learned/Conclusion

The main thing that we learned from this project was how much more difficult things become when you add a complicated tool like a forensics logger to the backend of a website. Making the actual messaging site was pretty straightforward, but the backend was a whole different story. We had to think about proper logging structure, encryption, and how to collect MAC addresses. Another lesson we learned was that keeping the repository organized from the beginning is a priority that we did not realize existed. Adding everything to the correct spots after the fact was a pain as committing changes in GitHub is much more difficult than we anticipated. There was always something wrong with the syncing. Overall, WhereApp was a fun project in its own way because it was the first time we had ever made a chat room type of website. Another very useful thing we found along the way was that the chatroom was perfect for copying and pasting into vms as it was instant compared to the other method of using outlook.