

## Milestone 3 Progress Evaluation

Project Ragnarok: Post-Quantum Cybersecurity with Lightning Data

### Group Members

- Joanna Zhang ([zhangj2022@my.fit.edu](mailto:zhangj2022@my.fit.edu))
- Gianni Bubb ([gbubb2022@my.fit.edu](mailto:gbubb2022@my.fit.edu))
- Aidan Nelappana ([anelappana2021@my.fit.edu](mailto:anelappana2021@my.fit.edu))

### Advisor / Client

- Dr. Bhattacharyya, [sbattacharyya@fit.edu](mailto:sbattacharyya@fit.edu)
- Dr. Amitabh Nag

**Institution:** Florida Institute of Technology – Senior Design

### 1. Progress of Current Milestone (Progress Matrix)

Task	Completion %	Aidan	Gianni	Joanna	To Do
1. Complete backend integration	40%	20%	10%	10%	Integrate the data fetchers into the API and the database
2. NIST 800-90B entropy wrappers	98%	98%	0%	0%	Publish to cargo and integrate with server
3. Finalize API & routing	90%	30%	50%	10%	More testing and features as necessary
4. Strengthen security controls	40%	0%	10%	30%	Continue with frontend integration

## **2. Discussion of Accomplished Tasks**

### **Task 1: Complete backend integration**

The backend integration consists of combining all the features and dev branches that have been made and ensuring their seamless interaction with legacy and future features. The Python API works with the new API request system in the webserver; we need to fix the authentication on the website. Currently it does not match the development on the backend, and we need to start integrating the data fetchers into the server while ensuring data safety and security. We must also decide which legacy features made in Project Thor should be phased out and replaced with the new feature workflow.

### **Task 2: NIST 800-90B entropy wrappers**

The NIST 800-90B entropy Rust wrappers have been made and tested. We need to upload the wrappers to cargo for public use before we integrate with the Rust server.

### **Task 3: Implement API & routing**

The backend API endpoints have been fully implemented and will be continued to be tested to ensure it meets requirements. Currently it supports server health, key generation, file encryption, entropy analysis, and file upload requests. The corresponding design and requirements will be updated to match. The Python wrapper API has been implemented and can properly communicate with the backend.

### **Task 4: Strengthen security controls**

We will be building the frontend authentication layer for an API by creating the user interface and logic that allow users to securely log in, manage their session, and access protected resources. A login form will be designed where users can enter their username and password, and those credentials will be sent to the backend API for verification. The response will be handled by storing an authentication token or session when login is successful. User state will be managed across the application, and routes will be protected so only authenticated users can access certain pages. A logout feature will also be implemented. Overall, a bridge will be established between the frontend and backend authentication system to ensure secure and controlled access to the application.

## **3. Team Member Contributions**

**Aidan Nelappana:** NIST 800-90B Entropy testing suit library wrappers for Rust, developed data fetchers for the NASA historical lighting data, NOAA solar wind data, geomagnetic data, and earthquake data. Assistance with problems running Thor-Rust server on Gianni's computer.

**Gianni Bubb:** Finished development of Python API wrapper and will continue to test and add features as needed. Python API correctly communicates with Rust Server and can perform requests. Currently it can perform a POST request to request a key in JSON format. I updated our repository with a branch for this feature and created the documentation.

**Joanna Zhang:** I started on building the frontend authentication layer for an API by creating the user interface and initial logic for user login and session handling. I built a login form where users can enter their username and password, and I set up the structure to send those credentials to the backend API for authentication. I also began implementing token handling logic to support storing and using an authentication token once login is successful. In addition, I prepared the foundation for attaching the token to protected API requests and setting up protected routes for restricted pages. Overall, I established the core frontend structure needed to support secure authentication and further backend integration.

**Plan for Next Milestone (Task Matrix)**

<b>Task</b>	<b>Aidan</b>	<b>Gianni</b>	<b>Joanna</b>
1. Test Rust Server on Raspberry Pi	Will assist with any debugging	Lead testing on Raspberry Pi and ensure features work as expected.	Will assist with any debugging
2. Data fetchers	Build and rollout fully integrated high entropy data fetchers and test min entropy	Support with API integration	Support with Server integration and security
3. Entropy analysis and maximization	Will gather $10^6$ data points per data set and run NIST 800-90B entropy testing suite	Support with API integration	Support with server integration
4. Gather high entropy data sets with over $10^6$ samples	Construct and deploy the data fetchers	Assist wherever needed	Assist wherever needed

**Discussion of Planned Tasks**

**Task 1:**

We aim to run the webserver on the Raspberry Pi and tackle any bugs that might arise in getting the product to be production ready.

**Task 2:**

We aim to fully integrate the data fetchers of different proposed sources of high entropy data. This data will be prepared for efficient storage and security. We need  $10^6$  samples of each proposed high entropy data source.

**Task 3:**

We aim to finalize and implement the entropy analysis and maximization tools for the server; this includes a fast implementation of Toeplitz Hashing and SHA-3.

**Task 4:**

We aim to gather enough data in order to ensure statistical accuracy in our minimum entropy results for base line testing.

**Meetings**

**Client Meeting Date(s)**

April 17, 2026

**Client Feedback**

**Faculty Advisor Meeting Date(s)**

April 17, 2026

**Faculty Advisor Feedback**

**Task 1:**

**Task 2:**

**Task 3:**

**Task 4:**

**Faculty Advisor Signature:** \_\_\_\_\_ **Date:** \_\_\_\_\_

**Evaluation by Faculty Advisor**

- Faculty Advisor: detach and return this page to Dr. Chan (HC 209) or email the scores to [pkc@cs.fit.edu](mailto:pkc@cs.fit.edu)
- Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Joanna Zhang	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Gianni Bubb	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Aidan Nelappana	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Faculty Advisor Signature: \_\_\_\_\_ Date: \_\_\_\_\_