1. **Healthcare and IoT Device Vulnerability Testbed**
2. **Team Members**

- Justin Bower - jbower2024@my.fit.edu
- Nathan Maloney - nmaloney2022@my.fit.edu
- Ipule Pipi - ipipi2022@my.fit.edu

3. **Faculty advisor(s):** Dr. Snedah Sudhakaran and Dr. Abdullah Aydeger
4. **Client(s):** Prospective Manufacturers/Engineers
5. **Progress of Current Milestone (Milestone 1 - Feb 23)**

Task	Completion %	Justin	Nathan	Ipule	To-Do
Tool Investigation	100%	33%	33%	33%	Nothing
Tooling Demos	66%	33%	33%	0%	Create demo webpage
Tooling Integration	35%	10%	25%	0%	Integrate remaining tools from investigation
Custom Scripting	50%	50%	0%	0%	Integrate existing scripts and create more
Demo/Testing Environment Integration	50%	25%	25%	0%	Integrate container and webpage to AWS

6. **Discussion of each accomplished task (and obstacles) for Milestone 1:**

• Task 1: Tool Investigation

Initial tools were investigated and researched to find the most optimal usage for the given time constraintsMost tools related to reverse engineering/analysis of devices were chosen based on licensing requirements and availability. The feature set was an aspect but was not reviewed as heavily. Cloud resources were investigated mainly by feature set and experience. All options reviewed offered similar payment options to keep costs low so more weight was given to options with larger sets of features. For web and backend the review process mainly focused on experience with the tools. Due to time constraints tools where the features were already understood were chosen.

• Task 2: Tooling Demos

We created test scripts/demos for major tools being used in this project. For RE/analysis tools this involved creating demo containers and scripts. The containers were created to showcase how the tools would be used in the final implementation while scripts were used to showcase functionality earlier. For cloud infrastructure the demo involved creating a test virtual private cloud using infrastructure as code. This allows for a consistent deployment throughout the testing phase.

• Task 3: Tooling Integration

The main Docker environment has been created for initial tooling set up and ultimate integrations with AWS. Further container work will be required to allow all tools to work together as one cohesive test suite. However, the environment is ready to be integrated so that said tooling work can be completed, along with further GUI work. This GUI work will be terminal/tool windows that pop up, isolated within the container, with kernel-level access given to tasks that require it (as docker containers share the kernel with the host).

• Task 4: Custom Scripting

Wrote preliminary firmware extraction and entropy analysis scripts in python, with helper functions, which are generalized for various interfaces. The obstacle in this milestone was that we have yet to investigate the common interfaces across devices, hence why the scripts were written as general implementations, with room for further implementations after researching devices' interfaces.

• Task 5: Demo/Testing Environment Integration

Did heavy research into the process of deploying containers to the cloud. This involved not only reading documentation and best practices but also interviewing an expert on the matter to gain a deeper understanding of the platform. This has allowed us to begin using containerized services faster than would have been possible before allowing us to skip making temporary dedicated compute based solutions. Tests have also begun involving attempting to deploy a container to the cloud using ECS.

7. Discussion of contribution of each team member to Milestone 1:

Justin Bower:

- Wrote initial docker container code, along with README/comments for use and integration of said code within AWS, then committed to respective branch within the main project GitHub repository
- Wrote sample python scripts for firmware extraction and entropy analysis
- Spearheaded email communication with faculty advisors
- Explained/illustrated design for preliminary GUI interface design to the rest of the team, for eventual implementation into frontend and backend of the test suite.

Nathan Maloney:

- Setup dev container to allow for a consistent development environment across team members and systems along with initial repository setup including file structure, README, and .gitignore
- Created testing environment on AWS to allow integration to begin immediately through the use of infrastructure as code to allow for faster deployment
- Stayed late after initial meetings to ensure greater team understanding, lab access, and material availability
- Researched AWS to provide team with deeper understanding of available resources and tools

Ipule Pipi:

- Began work on setting up basic front end and back end testing
- Stayed late after initial meetings to ensure greater team understanding, lab access, and material availability
- Worked with IT department to coordinate obtaining wired ethernet connections for the

IoT lab

8. Plan for the next Milestone (Milestone 2 - Mar 30)

Task	Justin	Nathan	Ipule
Tooling Demos	Nothing	Nothing	Create webpage demo using investigated tools
Integration Testing	Work to integrate container into AWS	Ensure cloud resources are properly running	Work to integrate frontend and backend with AWS
Device Examination	Examine chipsets, documentation, and plan device mounting	Nothing	Nothing
Tooling implementation	Migrate containers to AWS ECS to allow for deployment and access	Deploy and test database structures such as DynamoDB and S3	Add in support on front and backend for users to access currently running containers/services
Scripting	Examine interfaces to look for most common RE vectors	Investigate integrations of across multiple device interfaces	Create backend microservices
Infrastructure development	Nothing	Create scripts to automate infrastructure creation and security	Nothing

9. Discussion of each planned task for Milestone 2:

• Task 1: Tooling Demos

Tooling demos will consist of finalizing the frontend and backend demos this will mainly involve creating a test webpage that can be locally hosted running on the tools that will be used for the final deployment. This will also be uploaded to AWS for part of the integration testing.

• Task 2: Integration Testing

Integration testing will mainly involve ensuring that tool demos are able to function properly when combined with each other. This will involve allocating cloud compute to allow for basic webpage functionality along with containerized services.

• Task 3: Device examination

Initial analysis of the target Healthcare/IoT devices will enable an expansion of the test suite that is tailored to the available interfaces and reverse engineering vectors. The most common interfaces will become vectors by which further reverse engineering and vulnerability research can be executed by users, using the software test suite, and scripts geared toward those ends.

• Task 4: Tooling implementation

Integration of tools that will be used will involve heavy setup of ECS. This is due to the very different access layer when a container is placed within ECS compared to running normally through docker. To prevent the user from having access to databases that they shouldn't this will involve creating security layers composed of other containerized compute services. The data generated by this step will be passed onto the databases to allow for storage and multi user work.

• Task 5: Scripting

Further scripting will require the deeper examination of the physical devices, referencing documentation, and established interfaces. The goal of additional scripts will be to further the test suite's capabilities, and complement the physical side of the test suite, through automations and custom tooling.

• Task 6: Infrastructure development

Develop more infrastructure as code to deploy more services such as databases and EC2 instances. This infrastructure will need security roles and routing to allow for secure access without allowing the user to potentially impact the data stored within databases or open up new computers. This will be done through a series of layers of dedicated servers and serverless compute instances.

10. Date(s) of meeting(s) with Client during the current milestone:

2/18/26 @ 2:00 p.m.

11. Client feedback on the current milestone

• See Faculty Advisor Feedback below

12. Date(s) of meeting(s) with Faculty Advisor during the current milestone:

2/18/26 @ 2:00 p.m.

13. Faculty Advisor feedback on each task for the current Milestone

• Task 1:

• Task 2:

• Task 3:

• Task 4:

• Task 5:

Faculty Advisor Signature: _____ Date: 2/23/26

Faculty Advisor Signature: _____ Date: 2/23/26