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:** Dr. Snedah Sudhakaran and Dr. Abdullah Aydeger
4. **Client:** Prospective Manufacturers/Engineers
5. **Date(s) of Meeting(s) with the Client for developing this Plan:**
   Friday, Jan 16 @ 12:00 and Friday, Jan 23rd @ 3:00 p.m.
6. **Goal and motivation:**
   The goal of this project is to create a cybersecurity testbed for examining IoT and medical devices. This will allow us to easily probe for security vulnerabilities, for the benefit of researchers, and manufacturers.
   Currently, the IoT lab lacks a dedicated platform for pulling binaries, examining firmware, and exploring other device exploits; moreover, we intend to satisfy this need by creating this testbed.
   This project will challenge our technical abilities, provide real world benefits, and further our knowledge of cybersecurity, integrating tools, automation, and other related skills.

7. **Approach (key features of the system):**

- Users can connect a variety of IoT and medical devices that use standard interfaces to the test bed to enable device access and manipulation
- Users can extract firmware directly from connected IoT and medical devices, obtaining binary images, for detailed offline examination, and vulnerability assessment.
- Users can analyze firmware entropy to evaluate randomness and compression characteristics, while attempting pre-existing decryption techniques, to potentially access encrypted and/or obfuscated sections of the firmware.
- Users can apply pre-existing automated exploitation scripts to quickly test known vulnerabilities, thereby streamlining the process of confirming known security weaknesses in device firmware, or uncovering new vulnerabilities.
- Users can develop, integrate, and execute custom exploitation scripts of their own design, providing full flexibility to target specific vulnerabilities or implement novel attack vectors.
- Users can perform all connection, extraction, analysis, and exploitation tasks via an intuitive web-based graphical interface, significantly reducing complexity and improving efficiency for both novice and experienced users.

8. **Novel features/functionalities: Discuss which features/functionalities, if any, are novel and why.**

   The notion of a tightly integrated testbed, which automates as much as possible, while maintaining user-friendliness via a GUI, is a novel approach to cybersecurity analysis. This approach would be beneficial because it would lessen the background knowledge requirements for manufacturers or other researchers, while unifying the reverse engineering toolset significantly. Moreover, this testbed's benefit would be magnified by the hardware difficulties of medical devices, which are notoriously more difficult to analyze. By lowering the barrier to entry for examining IoT/Medical devices, we can ultimately achieve a more secure future for IoT and Medical devices, by providing ease of use to users.

9. **Algorithms and tools: potentially useful algorithms and software tools**

   Github: Version control and code collaboration tool
   Binwalk CLI: Useful for examining files, extracting sub-files, and entropy analysis
   React: Frontend
   TailwindCSS: Styling
   JavaScript/TypeScript: API
   Proxmox: Virtualization platform
   Docker: Containerization platform
   Kubernetes: Containerization platform
   Foremost CLI: Useful for examining files, extracting sub-files, and entropy analysis
   Ghidra: Proprietary tool for binary/firmware analysis
   IDA64: Proprietary tool for binary/firmware analysis
   Binary Ninja: Proprietary tool for binary/firmware analysis
   Python: Ideal for scripting purposes and simple integrations with many tools via pre-existing libraries/APIs
   Oracle Cloud Infrastructure (OCI)
   Amazon Web Services (AWS): Provides web server hosting, database backend, and other miscellaneous compute needs listed below:

   - Finch containerization platform
   - EC2 Compute instance
   - Amazon RDS DB, MongoDB, or DynamoDB
   - S3 Object Storage
   - Lambda isolated code execution
   - CodeCommit version control
   - Cloud9 code editor/IDE
   - VPC networking system

10. Technical Challenges: Discuss three main CSE-related challenges (for example, "we plan to use/do javascript for web programming, but we don't know much about javascript").

One major challenge in this project is our limited experience with decrypting and deobfuscating firmware extracted from IoT and medical devices. Many devices use proprietary encryption, compression techniques, or custom formats that complicate analysis, requiring additional research and experimentation to identify effective decryption and extraction methods.

Another challenge is developing the cloud-based infrastructure that supports the testbed, as we have limited experience designing and deploying secure and scalable cloud systems. Integrating web services, storage, and compute resources while managing access control and cost constraints will require careful architectural decisions.

The final challenge will be ensuring that exploitation and analysis tools can be executed safely presents a technical challenge, as the system must properly isolate scripts and processes to prevent unintended effects on the host system, connected devices, or cloud infrastructure.

11. Milestone 1 (Feb 23): itemized tasks:

- Compare and select technical tools for web server hosting, firmware/RE/entropy analysis; moreover, finalize comparisons Binwalk vs. Foremost, Ghidra vs. Binary Ninja vs. IDA, cloud provider AWS vs. OCI.
- Build and run small “hello world” demos for selected tools; moreover, basic firmware extraction with Binwalk, simple entropy plot, basic React page, AWS EC2 instance spin-up.
- Identify and propose initial solutions for the three main technical challenges (firmware decryption, cloud architecture, and script execution).
- Compare and select collaboration tools (version control, documents/presentations, communication, and shared calendar).
- Create Requirements Document (functional and non-functional requirements).
- Create High-level Design Document (architecture diagram, component breakdown, and data flow).
- Create Test Plan (test cases for each major feature and summarized acceptance criteria).

12. Milestone 2 (Mar 30): itemized tasks:

- Set up cloud infrastructure (VPC, EC2 instance, RDS/MongoDB/DynamoDB, S3 storage, and basic security groups/IAM roles).
- Implement device connection interface (support standard interfaces such as serial, JTAG, USB, network, and safe hardware passthrough or emulation).
- Implement firmware extraction module (direct binary dump from connected devices and storage in S3).

- Integrate entropy analysis and basic extraction tools (Binwalk + Foremost CLI wrappers in Python).
- Implement a safe execution environment for pre-existing automated exploitation scripts (i.e. sandboxing with Docker/containers or Lambda).
- Build backend API endpoints for extraction, entropy analysis, and script execution.
- Perform security audit of backend (basic access control, input validation).
- Demo: Connect a sample IoT device, extract firmware, run entropy analysis, run one automated script

13. Milestone 3 (Apr 20): itemized tasks:

- Build full web-based GUI using React + TailwindCSS (device connection page, firmware upload/extraction view, entropy visualization, script selection/execution interface, andresults dashboard).
- Integrate custom script upload and execution (user-provided Python scripts with strict sandboxing via Finch/Kubernetes/Docker).
- Implement decryption/deobfuscation workflow (UI for trying common keys/techniques and integration with Ghidra/Binary Ninja where possible).
- Add user authentication and role-based access if time allows; otherwise basic API keys.
- Perform end-to-end testing according to the Test Plan with manual and automated tests.
- Conduct usability testing, UX iterations, and user testing
- Prepare final demonstration video/walkthrough and project report.
- Demo full workflow on at least two different sample devices (one IoT, device and one medical device if possible) including custom script execution.

14. Task matrix for Milestone 1 (teams with more than one person)

Team Member	**Main Responsibility Category**	**Sub-Task 1**	**Sub-Task 2**	**Sub-Task 3**
Justin	RE/Cyber Tooling	Sourcing reverse engineering scripts	Compare Firmware, RE, and Entropy analysis tools	ToolScript integration with GUI/Web Server
Nathan	Web/Database Backend	Research/compare cloud providers	Set up basic cloud infrastructure and databases	Set up basic frontend
Ipule	Web Front End/ GUI Integration	Design & Build Client Side Interface	Integrate Frontend with Backend	Perform tests and iterations to improve UX