Network Feature Extraction From High Fidelity Cyber Simulation Environment #025-345

Project Proposal

Prepared for

DoD Aspire/Brian Cavanagh

DoD

By

Christopher Castro, Ryan Collette, William Lagos, Sam Soltanian

Under the supervision of

Irfan Ahmed

10/11/2024

Executive Summary

This document outlines an OS device classification system that utilizes machine learning. Rather than focusing on real-time breach detection, this system passively collects and analyzes web traffic, classifying operating systems within the network. By employing AI-driven classification through network feature extraction, we aim to provide enhanced visibility into device behavior, addressing gaps in both research and practical implementation of consumer-level hardware for OS identification. With wireless enabled devices becoming increasingly prevalent across consumer and critical environments, understanding network composition and behavior is crucial for anticipating and managing potential vulnerabilities. This project focuses on creating a scalable, easily replicable software solution capable of classifying devices on a network with minimal setup, operating as a passive extension to the router to streamline device monitoring.

Note: The Executive Summary should be updated between major reports as more knowledge is acquired and understanding of the project expands. For example, when submitting Preliminary Design Report in December 2024, make sure you update this page to reflect the progress on the project since the submission of Project Proposal in early October 2024.

Table of Contents

Section A. Problem Statement

OsirisML is a machine learning model designed to identify the operating system origin of a network request. As network complexity increases and organizations around the world fail to update their systems, we are left with a wide range of possible candidates for where a request originated from. The ability to identify these operating systems using passive network monitoring can prove to be a useful ability in modern warfare. Acquisition of these details provides us with a massive advantage in the enumeration phase while remaining undetected.

By leveraging passive data collection and applying ML classification models, we can extract significant features from TCP IP packets that enable a comprehensive classification system for identifying devices based on network behavior. This non-intrusive model allows us to monitor without alerting potential malicious actors, maintaining a low profile while gathering essential data. This implementation additionally allows for a massive range of expansion in terms of deployment, usage of data acquired from this enumeration phase, and efficiency improvements to current implementations.

Current research in operating system classification implementations remains underexplored, especially in high-fidelity environments where such devices are prevalent. Implementing a low-cost, deployable model that runs on accessible consumer grade hardware allows for easy reproductivity. This model could be seamlessly introduced into a military environment with minimal setup, allowing for a broad scope of traffic capture across various sectors.

Practical challenges include ensuring compatibility with a range of routers and maintaining system resilience against potential countermeasures that may attempt to interfere with passive monitoring. To support the model's complexity, the device may require augmentation with external processing resources or edge computing, adding an external processing layer that introduces new security considerations.

This approach paves the way for advanced network device mapping and monitoring, enriching our ability to understand and manage the dynamic composition of multi-platform environments across conflict and civilian zones alike.

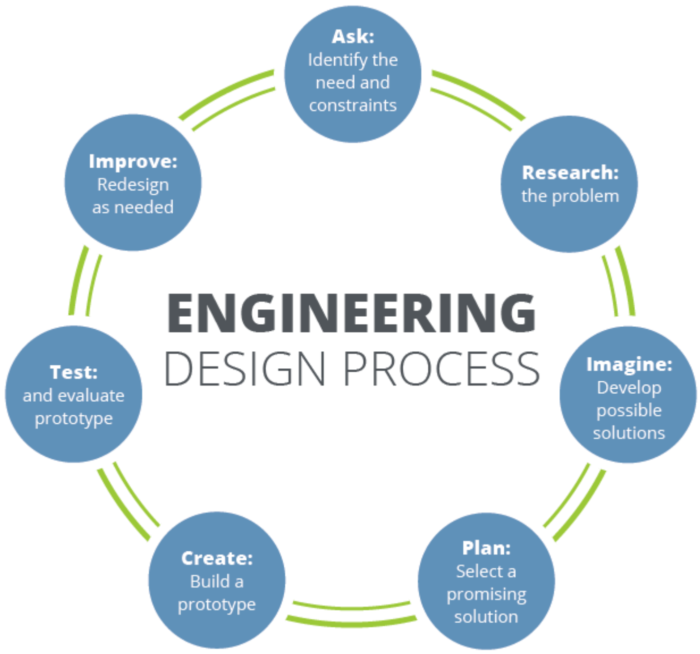


Figure 1. The iterative nature of the engineering design process [2].

Section B. Engineering Design Requirements

B.1 Project Goals (i.e. Client Needs)

- **Enhance Existing Model Functionality:**
  - Improve the current OS fingerprinting model to accurately identify not just the operating system type but also specific builds and versions from passive network traffic.
- **Increase Model Accuracy and Robustness:**
  - Utilize underexplored machine learning techniques to enhance the model's performance.
- **Streamline Deployment and Usability:**
  - Simplify the setup, usage, and deployment of the model.

B.2 Design Objectives

- **Improve Model Accuracy:**
  - Implement k-fold cross-validation to enhance model robustness.
  - Perform extensive hyperparameter tuning using advanced optimization tools like Optuna or Hyperopt.
- **Develop a Command-Line Interface (CLI):**
  - Create a user-friendly CLI to simplify interactions with the model, covering functionalities from data preprocessing to prediction.
- **Containerize the Application:**
  - Package the model and its dependencies into a Docker container to streamline setup and ensure consistent deployment across different environments.
- **Expand OS Detection Capabilities:**
  - Extend the model's functionality to identify specific OS builds and versions.

B.3 Design Specifications and Constraints

- The model will retain at least the same level of accuracy in classifying the original test set. (Functional constraint).
- The model will use K-fold cross validation to improve accuracy (Functional constraint)
- The model will be containerized to allow for ease of use on other platforms (Functional constraint)
- A user will be able to interact with the model using a command line tool (Functional constraint)

B.4 Codes and Standards

- **IEEE P7001**: Transparency in AI—document the decision-making process of the AI.

Section C. Scope of Work

The project scope defines the boundaries of the project encompassing the key objectives, timeline, milestones and deliverables. It clearly defines the responsibility of the team and the process by which the proposed work will be verified and approved. A clear scope helps to facilitate understanding of the project, reduce ambiguities and risk, and manage expectations. In addition to stating the responsibilities of the team, it should also explicitly state those tasks which fall *outside* of the team's responsibilities. *Explicit bounds* on the project timeline, available funds, and promised deliverables should be clearly stated. These boundaries help to avoid *scope creep*, or changes to the scope of the project without any control. This section also defines the project approach, the development methodology used in developing the solution, such as waterfall or agile (shall be chosen in concert with the faculty advisor and/or project sponsor). Good communication with the project sponsor and faculty advisor is the most effective way to stay within scope and make sure all objectives and deliverables are met on time and on budget.
Project Timeline:

- **November - December 2024:**
  - **Finalize team roles, project objectives, and deliverables.**
  - **Conduct initial research and begin drafting the preliminary design report.**
  - **Receive the new dataset from the sponsor and perform initial data preprocessing.**
- **January - February 2025:**
  - **Model development with k-fold cross-validation.**
  - **Extensive hyperparameter tuning using advanced optimization techniques.**
- **March 2025:**
  - **CLI application development and integration.**
  - **Containerization of the application and system integration.**
- **April 2025:**
  - **Documentation and user guides development.**
  - **Capstone EXPO preparation, including final reports and presentations.**

We will be following Agile methodology for development, so we can ensure each component is thoroughly tested and improved. This allows for flexibility and continuous feedback from the project sponsor and faculty advisor.

C.1 Deliverables

The project deliverables are those things that the project team is responsible for providing to the project sponsor. They are the things that are to be produced or provided as a result of the engineering design process. Some deliverables might include a specific number of alternative designs, required analyses to prove the design meets specifications, detailed machine drawings, functional diagrams or schematics, required computer code, flow charts, user manuals, desktop models, and functioning prototypes. A design "proof of concept" is not specific and should be more clearly defined. Academic deliverables include the team contract, project proposal, preliminary design report, fall poster and presentation, final design report, and Capstone EXPO poster and presentation. Provide a bulleted list of all agreed upon project deliverables.

In order to mitigate risks associated with the completion and delivery of the project deliverables, provide an outline of the most potentially disruptive, foreseeable obstacles. Some important issues to discuss with the design team, sponsor, and faculty advisor include the following:

- What deliverables require access to campus? Which/how many students regularly access campus and are physically available to complete tasks?
- What work can be done remotely? What resources might be needed in order to ensure that remote work can be completed effectively (e.g. software licenses, shared drives/folders, etc.)?
- What deliverables require ordering from third-party vendors? Will any components potentially required extended lead times? What can the team do in order to mitigate potential supply chain disruptions?

The project will produce the following deliverables:

1. **Enhanced Machine Learning Model:**
    - **A trained model capable of detecting specific OS minor versions and builds using the new dataset provided by the sponsor.**
    - **Implementation of k-fold cross-validation to improve model robustness.**
    - **Extensive hyperparameter tuning, including exploration of boost types, max_depth, gamma, learning rate, eta, and others.**
2. **Unified Command-Line Interface (CLI) Application:**
    - **A cohesive CLI that integrates all scripts into a user-friendly application.**
    - **Improved usability with clear documentation and help messages.**
    - **Error handling, logging, and configuration management features.**
3. **Containerized Application:**
    - **A Docker container of the OsirisML application for easy deployment across various operating systems, not limited to Debian Linux.**
    - **Dockerfile and instructions for building and running the containerized application.**

Potential Risks and Mitigation:

- **Data Quality and Availability:**
    - ***Risk:* The new dataset may have limitations or require additional preprocessing.**
    - ***Mitigation:* Close collaboration with the sponsor to understand the dataset and adjust preprocessing steps accordingly.**
- **Computational Resource Constraints:**

    - ***Risk:*** **Hyperparameter tuning and k-fold cross-validation may be computationally intensive.**
    - ***Mitigation:*** **Utilize university computational resources or cloud services.**
- **Time Management:**
    - ***Risk:*** **Potential delays due to unforeseen technical challenges.**
    - ***Mitigation:*** **Regular progress reviews and agile adjustments to the project plan.**

C.2 Milestones

Milestones are major project phases or tasks that need to be completed in order to ensure the project deliverables. They may include, among other things, completion of calculations, the development of a computational model, completion of an analysis, set-up of an experiment, completion of data acquisition, purchasing of hardware, assembly of a prototype, completion of testing procedures, development of required code, completion of wiring, post processing, etc.

A good rule of thumb is to break the project down into tasks of no larger than 2-3 weeks in length. These can be individual or group tasks. Breaking down the project into tasks/milestones gives the team and the advisor/sponsor a realistic understanding of what can be done in the allotted time. In an agile development approach, later tasks are expected to be adjusted (or changed) as the team works with the earlier developed tasks.

The amount of time it will take to accomplish each milestone and the approximate date that each milestone will be completed should be considered. Do not underestimate the time that it takes to write and prepare major reports and presentation materials. All deliverables and milestones should be included in the project timeline found in Appendix 1. Provide a summary table of all project milestones including required times and completion dates here.

Note: While the project scope, deliverable, and milestones are not intended to change throughout the project, this section should be revisited between major reports to ensure that it still accurately reflects the expectations and requirements of the project team, client, and faculty advisor. Any changes to the project scope, deliverable, and milestones should be thoroughly discussed and mutually agreed upon by all parties. Any changes to this section should be documented and justified in detail.

Project Milestones Timeline:

Finalize Team Roles, Project Objectives, and Deliverables

- **Tasks:**
    - Finalize team roles, project objectives, scope, and deliverables.
    - Submit the project proposal to VCU.
    - Prepare and submit the fall design poster.
- **Expected Completion: November 15, 2024**

---

2. Resource Procurement and Research

- **Tasks:**
  - **Resource Procurement:**
    - Contact VCU IT services to procure a high-performance virtual machine (VM) with at least 128 GB of RAM, accessible by all team members.
      - If VCU IT is unable to provide, research other alternatives such as AWS EC2 or GCP.
  - **Research:**
    - Conduct in-depth research on:
      - Advanced machine learning techniques for OS version detection.
      - Feature extraction methods relevant to OS fingerprinting and distinguishing specific OS versions.
      - K-Fold Cross-Validation and its impact on improving model accuracy and robustness.
      - Hyperparameter optimization techniques, including tools like GridSearchCV, RandomizedSearchCV, and Optuna.
      - Boosting algorithms and their variants to understand how different boost types affect model performance.
    - Begin drafting the **Preliminary Design Report**.
- **Expected Completion: November 29, 2024** (2 weeks)

---

3. Getting Model to Run Locally

- **Tasks:**
  - Set up development environments on procured machines.
  - Adjust existing scripts to accommodate the new dataset structure.
  - Ensure the initial model runs successfully on local machines using a subset of the data.
  - Begin initial data preprocessing and feature extraction.
  - Continue working on the **Preliminary Design Report** and complete it.
- **Expected Completion: December 13, 2024** (2 weeks)

---

4. Data Preprocessing and Feature Extraction

- **Tasks:**
  - Perform comprehensive data preprocessing and feature extraction, including:
    - Adjusting preprocessing scripts to fully accommodate the new dataset structure.

        - Extracting features relevant to specific OS version detection.
        - Cleaning and normalizing the data to prepare it for model training.
    - Validate the processed data and ensure it's ready for model development.
- **Expected Completion: December 27, 2024** (2 weeks)

---

5. Model Development with K-Fold Cross-Validation

- **Tasks:**
    - Implement **k-fold cross-validation** in the training pipeline to enhance model robustness.
    - Train initial machine learning models using the preprocessed dataset.
    - Analyze model performance metrics to identify areas for improvement.
    - Document initial findings and update the design report accordingly.
- **Expected Completion: January 31, 2025** (5 weeks)

---

6. Extensive Hyperparameter Tuning

- **Tasks:**
    - Perform comprehensive hyperparameter tuning using advanced optimization techniques.
    - Explore and optimize parameters such as:
        - **Boost Types** (e.g., gbtree, gblinear, dart).
        - **Max Depth**.
        - **Gamma**.
        - **Learning Rate (Eta)**.
        - **Min Child Weight**.
        - **Subsample** and **Colsample_bytree**.
    - Utilize tools like **Optuna** or **Hyperopt** for efficient hyperparameter search.
    - Record the optimal hyperparameters and their impact on model performance.
    - Update documentation with tuning results and insights.
- **Expected Completion: February 28, 2025** (4 weeks)

---

7. CLI Application Development

- **Tasks:**
    - Refactor existing scripts into a modular and maintainable codebase.
    - Develop a unified **Command-Line Interface (CLI)** using Python libraries such as **argparse** or **click**.
    - Implement user-friendly features including:

        - Clear help messages and usage instructions.
        - Input validation and informative error messages.
        - Logging mechanisms for tracking application processes.
    - Ensure the CLI covers all functionalities, from data preprocessing to model training and testing.
    - Begin integrating the CLI with the developed models.
- **Expected Completion: March 21, 2025** (3 weeks)

---

8. Containerization of the Application and System Integration

- **Tasks:**
    - Create a **Dockerfile** to containerize the OsirisML application.
    - Ensure all dependencies and environment configurations are included.
    - Integrate all components into a single, containerized application.
    - Perform extensive testing, including:
        - **Unit Tests** for individual components.
        - **Integration Tests** to ensure modules work together seamlessly.
        - **System Tests** to validate the entire application's functionality.
    - Test the containerized application on different operating systems to verify compatibility.
    - Optimize the Docker image for minimal size and efficient performance.
    - Provide documentation on how to build and run the Docker container.
- **Expected Completion: April 11, 2025** (3 weeks)

---

9. Documentation and User Guides

- **Tasks:**
    - Prepare comprehensive technical documentation, including:
        - Installation and setup instructions.
        - Detailed usage guides for all application features.
        - Developer documentation for future maintenance and enhancements.
    - Develop user manuals with examples and best practices.
    - Update the design report with finalized designs, methodologies, and findings.
- **Expected Completion: April 18, 2025** (1 week)

---

10. Capstone EXPO Preparation

- **Tasks:**
    - Prepare for the Capstone EXPO by:

        - Creating the poster presentation.
        - Writing the final project report.
        - Developing presentation materials and rehearsing the presentation.
    - Seek feedback from the project sponsor and faculty advisor to refine deliverables.
- **Expected Completion: April 25, 2025** (1 week)

C.3 Resources

Resources needed for project completion should be listed at the proposal stage. These resources can either be purchased within the Project Budget, or provided by the project sponsor. Some examples are: hardware such as HPCs or servers, software such as IDEs, data analysis platforms or version control systems. Access to cloud computing services may also be necessary to scale certain procedures. Additionally, databases containing operational data for testing, as well as libraries or APIs relevant to predictive analytics and machine learning may be required.

Hardware:

- **Computational Resources:**
    - **High-Performance Computing (HPC) Clusters:**
        - Access to university-provided HPC clusters or servers for computationally intensive tasks such as extensive hyperparameter tuning and k-fold cross-validation on large datasets.
        - Necessary during the **Extensive Hyperparameter Tuning** phase and when training models with k-fold cross-validation to handle increased computational load.

Software:

- **Programming Languages and Development Tools:**
    - **Python 3.x:**
        - The primary programming language for all development tasks.
        - Utilized extensively across all project phases.
    - **Integrated Development Environments (IDEs):**
        - **PyCharm**, or **Visual Studio Code** for code development and debugging.
    - **Version Control Systems:**
        - **Git** for version control.
        - **GitHub** for code hosting, collaboration, and issue tracking.

Machine Learning and Data Processing Libraries:

- **XGBoost:**
    - For model training, evaluation, and implementing advanced boosting algorithms.

  - Central to the **Model Development** and **Hyperparameter Tuning** phases.
- **Scikit-learn:**
  - For implementing k-fold cross-validation, initial model development, and basic machine learning tasks.
  - Used during the **Model Development with K-Fold Cross-Validation** phase.
- **Optuna** or **Hyperopt:**
  - Advanced hyperparameter optimization frameworks to efficiently search for optimal model parameters.
  - Employed during the **Extensive Hyperparameter Tuning** phase.
- **Pandas and NumPy:**
  - For data manipulation, preprocessing, and analysis.
  - Essential during the **Data Preprocessing** and throughout model development.
- **Scapy** or **Tshark** (if needed):
  - For advanced packet analysis and feature extraction from PCAP files.
  - May be required during **Dataset Acquisition and Data Preprocessing** if additional feature extraction is necessary.

Command-Line Interface (CLI) Development Libraries:

- **argparse**, **click**, or **typer:**
  - For building a user-friendly CLI application.
  - Integral to the **CLI Application Development** phase.

Containerization Tools:

- **Docker:**
  - For containerizing the application to ensure consistent deployment across different environments and operating systems.
  - Central to the **Containerization of the Application and System Integration** phase.

Testing Frameworks:

- **PyTest**:
  - For writing and running unit tests.
  - Employed during the **Containerization and System Integration** phase.

Data Resources:

- **Labeled Dataset Provided by the Sponsor:**
  - A new dataset containing network traffic data with specific OS version labels for training and testing the machine learning models.

    - Central to the **Dataset Acquisition and Data Preprocessing** and subsequent modeling phases.

Compute Services:

- **Cloud Computing Platforms (if required):**
    - **Amazon Web Services (AWS) EC2**, **Google Cloud Platform (GCP)**, or **Microsoft Azure:**
        - For scalable computing resources, especially during extensive hyperparameter tuning and model training.
        - Backup option if university HPC resources are insufficient or unavailable.

Section D. Concept Generation

A number of methods can be used to help generate design concepts from simple reflection and brainstorming, to working the problem backwards, using reverse thinking techniques, and looking to nature for inspiration (i.e. biomimicry). Existing solutions, or components of existing solutions, can be substituted, combined, adapted, modified, put to other uses, eliminated, or rearranged to meet new design objectives and specifications. A minimum of 3 overall design concepts is required for this section although more are welcome. Provide a brief description of how each design concept addresses the design problem. Discuss the potential pros and cons, including and potential risks of failure, of each of these concepts.

It is likely that each design concept may consist of several components. In this case, one or more of these components may offer a sub-problem that can be further explored, modified, or otherwise improved upon. These sub-problems may lead to the addition of several additional design concepts and may require the inclusion of a design concept chart or matrix to organize all ideas and potential solutions.

Provide any initial design sketches, drawings, 3D renderings, or conceptual models such as dataflow diagrams, process flows, etc. developed during the concept ideation phase. All hand drawings should be drawn to scale using basic engineering drafting tools (i.e. ruler, protractor, and compass). Geometric stencils can also be used to help produce quality hand drawings. Drawings should be presented in a profession manner, preferably done on engineering graph paper and using a high-quality scan. All sketches should be labeled to identify major components and different drawing views or projections if applicable. Basic dimensions should be provided to give a general sense of scale. Label each sketch or drawing with the name of the team member responsible for the sketch, the date it was drawn, and the drawing scale.

Section E. Concept Evaluation and Selection

Using a systematic decision-making process, evaluate each of the design concepts and choose the one that is most likely to succeed in meeting the design objectives and constraints. A Decision Matrix, or Pugh Matrix, helps to analyze alternatives, eliminate biases, and make rational decisions through thought and structure. First, work to develop a set of selection criteria for which to evaluate the previously generated design concepts. Selection criteria often include concepts of performance, cost, safety, reliability, risk, etc. Note that the selection criteria developed here will likely be more general than the project design objectives. As with the design objectives, conversations with the client help define appropriate selection criteria.

In many cases, the client may value the selection criteria differently, preferring that more emphasis be placed on some than others. In this case, weighting factors may be used to place more or less importance on the various criteria in the decision making process. Again, conversations with the client can be used to define criteria weighting factors. Often times, these conversations must be analyzed and interpreted by the team to determine which criteria are more important to the client and by how much. Feel free to discuss the assigned weighting factors with the client to see if they seem accurate.

Next, define an associated metric to represent each criteria. Metrics should be specific and quantifiable, providing numerical values that quantify the often vague concepts of the selection criteria. Metrics can be obtained, generated, or estimated through a number of methods including simple background research, preliminary design calculations, or basic analyses. Note that these metrics do not need to specifically align with the design specifications although there may be some commonality between the two. Provide a brief discussion of the rationale for selecting each of the assigned metrics.

Using the defined metrics, evaluated each design concept against all selection criteria by filling out a Decision Matrix. Design concepts can be compared by using simple rank scoring, raw scoring, or weighted scoring techniques and design concept with which to move forward can be selected. This type of process provides a meaningful, unbiased means for choosing a preliminary design concept prior to moving forward with more comprehensive, detailed analyses as provided in the design methodology section below. The results of this process should be discussed with the project client prior to moving forward with the selected design. Table 1 provides an example of a simple decision matrix.

Table 1. Example of a Decision Matrix.

	Design Concept A	Design Concept B	Design Concept C	Design Concept D
Criteria 1				
Criteria 2				
Criteria 3				
Criteria 4				
Criteria 5				
Total Score				

Note: Weights can be assigned to each criterion if desired.

Section F. Design Methodology

Provide a detailed explanation of the methods that will be used to help evaluate, improve, and evolve the design through the iterative engineering design process. Consider that ultimately, the final design must be verified and validated to ensure that it meets all of the previously developed and listed design objectives and specifications. Verification ensures that the design meets all specification, while validation confirms that the design functions as intended such to meet the client's needs. While it is common for initial design concepts to first be evaluated using simplified design criteria and metrics, the chosen design should be advanced, and later verified, using engineering calculations, computational models, experimental data, and/or testing procedures.

Use this section to describe any underlying physical principles and mathematical equations that govern the design. Provide details of any computer-aided modeling techniques used to evaluate the design including the software used, prescribed boundary conditions, and assumptions. Include a detailed description of any experimental testing methods including required testing equipment, test set-up layout, data acquisition and instrumentation, and testing procedures. If one or more prototypes is to be produced and tested, provide a detailed description of how each will be evaluated.

Note: The contents of this section are expected to vary from project to project. Subsections may be appropriate for providing details of analytical, computational, experimental, and/or testing methods. Some potential subsections that may be included in this section are provided. While critical design equations may be provided here, lengthy mathematical derivations may be included in an appendix. Validation procedures are critical and all projects should address such topic.

F.1 Computational Methods (e.g. FEA or CFD Modeling, example sub-section)

F.2 Experimental Methods (example subsection)

F.3 Architecture/High-level Design (example subsection)

F.5 Validation Procedure

Describe how the design team will validate that the final design meets the client's needs. This section should include a plan to meet with the client towards the end of the project to discuss final design details and demonstrate a prototype, experimental test, and/or simulation results. Provide a relative time frame for this validation to occur (e.g. "mid-March" or "early-April"). Include a brief discussion on how client feedback will be captured, such as a formal survey, interview, or observation notes of the client using the prototype. It may also include plans to solicit feedback from other stakeholders and/or potential users.

Section G. Results and Design Details

Use this section to highlight the major results of the design methodology described above including important analytical, computational, experimental, modeling, assembly, and testing results. This section should be one of the most substantial sections of the report showcasing all of the hard work and effort that went into the completion of the final design and delivery of the project deliverables. Show how the identified problem was solved.

Highlight the prominent features of the final design through analysis results, modeling, drawings, renderings, circuit schematics, instrumentation diagrams, flow and piping diagrams, etc. to show that the design functions as intended and meets all design objectives and constraints. Overview designs such as dataflow diagrams, process flow, swim lane diagrams, as well as presentation-layer designs (e.g. storyboards for front-ends) should be included here. Detailed designs such as database designs, software designs, procedure flowcharts, or pseudocode should be included here. Support computational and experimental results with key plots and figures. All supporting figures should be clearly labeled and annotated to highlight the most important points of the figure (i.e. explicitly point out what the reader should focus on or understand about the image).

Note that while all results should be used to help inform design decisions, not all results may be necessary to include in the main body of the report. Extraneous supporting results (e.g. graphs, data, design renderings, drawings, etc.) that are not necessary for presenting the fundamental findings can be placed in one or more appendices. Detailed documentation of each program module can be provided as appendix.

G.1 Modeling Results (example subsection)

G.2 Experimental Results (example subsection)

G.3 Prototyping and Testing Results (example subsection)

G.4. Final Design Details/Specifications (example subsection)

Note that while the design constraints and specifications may have provided minimum or maximum values, or ranges or values, that the design needed to meet, the final design specifications should be listed here showing that the required design values were met. A list of final design details can also be included demonstrate fulfillment of the design objectives.

Note: Preliminary results should be included in the Preliminary Design Report to show the progress made of the selected design concept to-date. This section should be updated for the Final Design Report to include documentation of all of the work that was completed on the project throughout the entirety of the academic year.

Section H. Societal Impacts of Design

In addition to technical design considerations, contemporary engineers must consider the broader impacts that their design choices have on the world around them. These impacts include the consideration of public health, safety, and welfare as well as the potential societal, political/regulatory, economic, environmental, global, and ethical impacts of the design. As appropriate for the project design, discuss how each of these considerations influenced design choices in separate subsections. How will the design change the way people interact with each other? What are the political implications of the design? Does the technology have the potential to impact or shift markets? Does the design have any positive or negative effects on the environment? Don't forget to consider unintended consequences such as process or manufacturing byproducts. What impacts might the design have on global markets and trade? Are there any ethical questions related to the design?

While it is hard to forecast the various impacts of a technology, it is important to consider these potential impacts throughout the engineering design process. When considered during the early stages of the design phase, consideration of these impacts can help determine design objectives, constraints, and specifications and help drive design choices that may mitigate any potential negative impacts or unintended consequences.

Note: A minimum of 4 of these design considerations, including the consideration of public health, safety, and welfare, are required for the Preliminary Design Report while a section for all considerations must be included in the final design report.

H.1 Public Health, Safety, and Welfare

Provide a list of all design safety features and provide a brief description of each. Discuss the potential effects the design may have on public health, safety, and welfare. References to the codes and standards previous provided and the organizations that produced them may be summarized or referenced here.

H.2 Societal Impacts

H.3 Political/Regulatory Impacts

H.4. Economic Impacts

H.5 Environmental Impacts

H.6 Global Impacts

H.7. Ethical Considerations

Section I. Cost Analysis

Provide a simple cost analysis of the project that includes a list of all expenditures related to the project. If an experimental test set-up or prototype was developed, provide a Bill of Materials that includes part numbers, vendor names, unit costs, quantity, total costs, delivery times, dates received, etc. Do not forget to include all manufacturing costs incurred throughout the completion of the project. If the design is expected to become a commercial product, provide a production cost estimate including fixed capital, raw materials, manufacturing (including tooling and/or casting), and labor costs to produce and package the device. Note that this type of detailed cost analysis may be listed as a project deliverable.

Note: The Preliminary Design Report should include all costs incurred to date. It is expected that this section will be expanded and updated between the preliminary and final design reports.

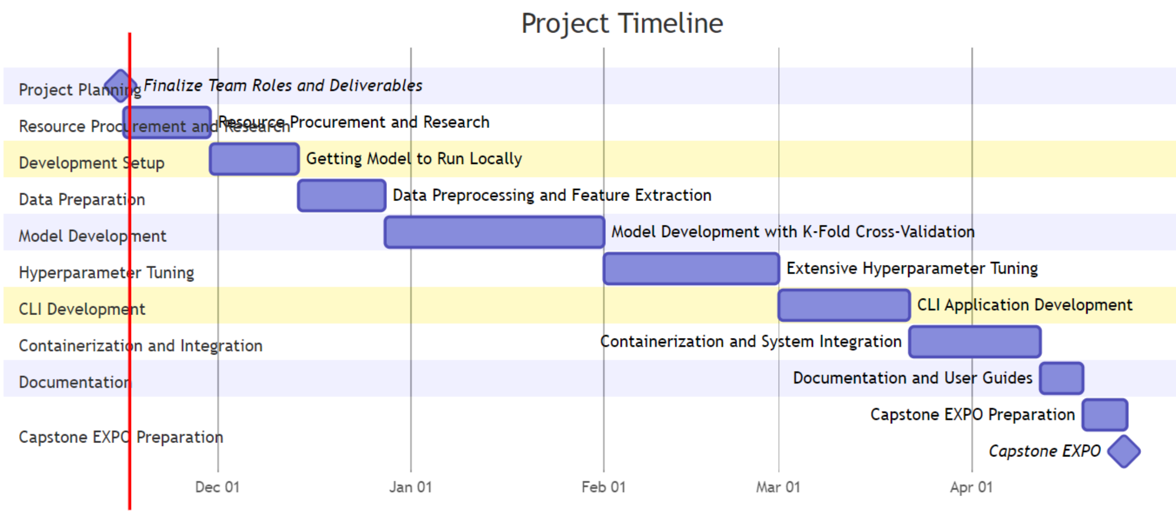
Section J. Conclusions and Recommendations

Use this section to summarize the story of how the design team arrived at the final design. Focus on the evolution of the design through the use of the engineering design process including lessons learned, obstacles overcome, and triumphs of the final design. Revisit the primary project goals and objectives. Provide a brief summary of the final design details and features paramount to the function of the design in meeting these goals and objectives.

A discussion may be included to discuss how the design could be further advanced or improved in the future. If applicable, summarize any questions or curiosities that the final results/design of this effort bring to mind or leave unanswered. If this project might continue on as a future (continuation) senior design project, detail the major milestones that have been completed to date and include any suggested testing plans, relevant machine drawings, electrical schematics, developed computer code, etc. All relevant information should be included in this section such that future researchers could pick up the project and advance the work in as seamless a manner as possible. Documents such as drawings, schematics, and codes could be referenced here and included in one or more appendix. If digital files are critical for future work, they should be saved on a thumb drive, external hard drive, cloud, etc. and left in the hands of the project advisor and/or client.

Appendix 1: Project Timeline

Provide a Gantt chart of similarly composed visual timeline showing the start and end dates of all completed tasks and how they are grouped together, overlapped, and linked together. Include all senior design requirements including design reports and Expo materials (i.e. Abstract, Poster, and Presentation). All major milestones should be included in the timeline.



Appendix 2: Team Contract (i.e. Team Organization)

Copy and paste the content from the completed Team Contract here starting with Step 1 of the Team Contract and including all content following the 'Contents' list.

Step 1: Get to Know One Another. Gather Basic Information.

Task: This initial time together is important to form a strong team dynamic and get to know each other more as people outside of class time. Consider ways to develop positive working relationships with others, while remaining open and personal. Learn each other's strengths and discuss good/bad team experiences. This is also a good opportunity to start to better understand each other's communication and working styles.

Team Member Name	***Strengths each member bring to the group***	***Other Info***	***Contact Info***
William Lagos	*Technical skills, flexibility, problem solving, teamwork*		*lagoswj@vcu.edu*
Sam Soltanian	*Always trying to learn, problem-solving flexible*	*I enjoy learning from other people and in turn offering new knowledge*	soltanians@vcu.edu
Christopher Castro	Friendly, curious, bias for action, hard worker	Unfamiliar with cybersecurity topics / PCAP data, but eager to fill in knowledge gaps.	castrocm5@vcu.edu
Ryan Collette	*Always want to learn, willing to adapt to complete goals, enjoy working with a team.*	*Lack of in depth cyber-security concepts, but surface level knowledge and very interested in learning.*	colletterc@vcu.edu

Other Stakeholders	***Notes***	***Contact Info***
Project Sponsor	*Meeting Friday's @ 12:30PM @ Teams*	*Jae Sung Kim jaesung.kim6.civ@army.mil*
Staff Mentor	*Initial Meeting Wed, Sep. 4 @ Zoom*	*Irfan Ahmed iahmed3@vcu.edu*

Step 2: Team Culture. Clarify the Group's Purpose and Culture Goals.

Task: Discuss how each team member wants to be treated to encourage them to make valuable contributions to the group and how each team member would like to feel recognized for their efforts. Discuss how the team will foster an environment where each team member feels they are accountable for their actions and the way they contribute to the project. These are your Culture Goals (left column). How do the students demonstrate these culture goals? These are your Actions (middle column). Finally, how do students deviate from the team's culture goals? What are ways that other team members can notice when that culture goal is no longer being honored in team dynamics? These are your Warning Signs (right column).

Resources: More information and an example Team Culture can be found in the Biodesign Student Guide "Intentional Teamwork" page (webpage | PDF)

Culture Goals	***Actions***	***Warning Signs***
1: Punctuality	- *Students will arrive to meetings on time* - *Students need to communicate any needs to cancel or reschedule meetings*	- *Student misses with no explanation first meeting, warning is granted* - *Student misses meetings with no explanation afterwards – issue is brought up with faculty advisor*
2: Informing the group of any delays in completing assignments	- *Stay up to date with each other's project responsibilities* - *Set reasonable deadlines and note when an extension is needed*	- *Student shows up for weekly meeting with no considerable work done*
3: Be open to critical feedback and receptive	*- Be receptive to ideas that potentially differ from your own.* *- Be open to alternative design options that you may be unfamiliar with.* *- Respond to feedback in a mature and normal manner.*	*- Students are unreceptive/immature about feedback.*

Step 3: Time Commitments, Meeting Structure, and Communication

Task: Discuss the anticipated time commitments for the group project. Consider the following questions (don't answer these questions in the box below):

- What are reasonable time commitments for everyone to invest in this project?
- What other activities and commitments do group members have in their lives?
- How will we communicate with each other?
- When will we meet as a team? Where will we meet? How Often?
- Who will run the meetings? Will there be an assigned team leader or scribe? Does that position rotate or will the same person take on that role for the duration of the project?

Required: How often you will meet with your faculty advisor advisor, where you will meet, and how the meetings will be conducted. Who arranges these meetings?
See examples below.

Meeting Participants	***Frequency Dates and Times / Locations***	***Meeting Goals Responsible Party***
Students Only	*As Needed, in-person or on Discord based on requirements per week. Scheduled based on discussion every Thursday.*	*Actively work on project (Group will document these meetings by taking photos of whiteboards, physical prototypes, etc, then post on Discord and update Capstone Report)*
Students Only	*Every Thursday during the seminar time block*	*Update group on weekly challenges and accomplishments, scheduling as needed meetings for other time slots during the week.*
Students + Faculty advisor	*Every Friday at 12:30 PM on Zoom*	*Update faculty advisor and get answers to our questions, updates following agile methodology (what we've accomplished, what we're blocked on, what we're doing next)*
Project Sponsor	*Every Friday at 12:30 PM on Zoom*	*Update project sponsor and make sure we are on the right track*

Step 4: Determine Individual Roles and Responsibilities

Task: As part of the Capstone Team experience, each member will take on a leadership role, ***in addition to*** contributing to the overall weekly action items for the project. Some common leadership roles for Capstone projects are listed below. Other roles may be assigned with approval of your faculty advisor as deemed fit for the project. For the entirety of the project, you should communicate progress to your advisor specifically with regard to your role.

- **Before meeting with your team**, take some time to ask yourself: what is my "natural" role in this group (strengths)? How can I use this experience to help me grow and develop more?
- **As a group,** discuss the various tasks needed for the project and role preferences. Then assign roles in the table on the next page. Try to create a team dynamic that is fair and equitable, while promoting the strengths of each member.

Communication Leaders

Suggested: Assign a team member to be the primary contact for the client/sponsor. This person will schedule meetings, send updates, and ensure deliverables are met.

Suggested: Assign a team member to be the primary contact for faculty advisor. This person will schedule meetings, send updates, and ensure deliverables are met.

Common Leadership Roles for Capstone

1. **Project Manager:** Manages all tasks; develops overall schedule for project; writes agendas and runs meetings; reviews and monitors individual action items; creates an environment where team members are respected, take risks and feel safe expressing their ideas.
   Required: On Edusourced, under the Team tab, make sure that this student is assigned the Project Manager role. This is required so that Capstone program staff can easily identify a single contact person, especially for items like Purchasing and Receiving project supplies.
2. **Logistics Manager:** coordinates all internal and external interactions; lead in establishing contact within and outside of organization, following up on communication of commitments, obtaining information for the team; documents meeting minutes; manages facility and resource usage.
3. **Financial Manager:** researches/benchmarks technical purchases and acquisitions; conducts pricing analysis and budget justifications on proposed purchases; carries out team purchase requests; monitors team budget.
4. **Systems Engineer:** analyzes Client initial design specification and leads establishment of product specifications; monitors, coordinates and manages integration of sub-systems in the prototype; develops and recommends system architecture and manages product interfaces.
5. **Test Engineer:** oversees experimental design, test plan, procedures and data analysis; acquires data acquisition equipment and any necessary software; establishes test protocols and schedules; oversees statistical analysis of results; leads presentation of experimental finding and resulting recommendations.

Team Member	***Role(s)***	***Responsibilities***
William Lagos	Systems Engineer	Establish project specifications, monitor, coordinate and manages integration of subsystems in the prototype, and develop/recommend systems for use in product
Christopher Castro	Project Manager	Write agendas and run meetings, review and monitor individual action items. Come up with major stories and set up agile board, organize action items.
Sam Soltanian	Logistics Manager	Communicate with Faculty Advisor & Advisor and communicate between team and faculty/sponsor. Make sure everyone is on the same page.
Ryan Collette	Test Engineer	Collaborate and plan experimental design, plan testing frameworks, scheduling testing, statistical analysis of training results or confidence weights.

Step 5: Agree to the above team contract

Team Member: Sam S *Signature:* **Sam Soltanian**

Team Member: Ryan C *Signature: Ryan Collette*

Team Member: Chris C *Signature: Christopher Castro*

Team Member: William L *Signature: William Lagos*

Appendix 3: [Insert Appendix Title]

Note that additional appendices may be added as needed. Appendices are used for supplementary material considered or used in the design process but not necessary for understanding the fundamental design or results. Lengthy mathematical derivations, ancillary results (e.g. data sets, plots), and detailed mechanical drawings are examples of items that might be placed in an appendix. Multiple appendices may be used to delineate topics and can be labeled using letters or numbers. Each appendix should start on a new page. Reference each appendix and the information it contains in the main text of the report where appropriate.

Note: Delete this page if no additional appendices are included.

References

Provide a numbered list of all references in order of appearance using APA citation format. The reference page should begin on a new page as shown here.

[1] VCU Writing Center. (2021, September 8). *APA Citation: A guide to formatting in APA style.* Retrieved September 2, 2024. https://writing.vcu.edu/student-resources/apa-citations/

[2] Teach Engineering. *Engineering Design Process*. TeachEngineering.org. Retreived September 2, 2024. https://www.teachengineering.org/populartopics/designprocess